SECURWARE 2014

The Eighth International Conference on Emerging Security Information, Systems and Technologies

ISBN: 978-1-61208-376-6

November 16 - 20, 2014

Lisbon, Portugal

SECURWARE 2014 Editors

Rainer Falk, Siemens AG - München, Germany

Carlos Becker Westphall, Federal University of Santa Catarina, Brazil

SECURWARE 2014

Foreword

The Eighth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2014), held between November 16-20, 2014 in Lisbon, Portugal, continued a series of events covering related topics on theory and practice on security, cryptography, secure protocols, trust, privacy, confidentiality, vulnerability, intrusion detection and other areas related to low enforcement, security data mining, malware models, etc.

Security, defined for ensuring protected communication among terminals and user applications across public and private networks, is the core for guaranteeing confidentiality, privacy, and data protection. Security affects business and individuals, raises the business risk, and requires a corporate and individual culture. In the open business space offered by Internet, it is a need to improve defenses against hackers, disgruntled employees, and commercial rivals. There is a required balance between the effort and resources spent on security versus security achievements. Some vulnerability can be addressed using the rule of 80:20, meaning 80% of the vulnerabilities can be addressed for 20% of the costs. Other technical aspects are related to the communication speed versus complex and time consuming cryptography/security mechanisms and protocols.

Digital Ecosystem is defined as an open decentralized information infrastructure where different networked agents, such as enterprises (especially SMEs), intermediate actors, public bodies and end users, cooperate and compete enabling the creation of new complex structures. In digital ecosystems, the actors, their products and services can be seen as different organisms and species that are able to evolve and adapt dynamically to changing market conditions.

Digital Ecosystems lie at the intersection between different disciplines and fields: industry, business, social sciences, biology, and cutting edge ICT and its application driven research. They are supported by several underlying technologies such as semantic web and ontology-based knowledge sharing, self-organizing intelligent agents, peer-to-peer overlay networks, web services-based information platforms, and recommender systems.

We take here the opportunity to warmly thank all the members of the SECURWARE 2014 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to SECURWARE 2014. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the SECURWARE 2014 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that SECURWARE 2014 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in emerging security information, systems and technologies.

We are convinced that the participants found the event useful and communications very open. We hope Lisbon provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

SECURWARE 2014 Chairs:

SECURWARE Advisory Chairs
Juha Rőning, University of Oulu, Finland
Catherine Meadows, Naval Research Laboratory - Washington DC, USA
Petre Dini, Concordia University, Canada / China Space Agency Center - Beijing, China
Reijo Savola, VTT Technical Research Centre of Finland, Finland
Masaru Takesue, Hosei University, Japan
Mariusz Jakubowski, Microsoft Research, USA
Emmanoil Serelis, University of Piraeus, Greece
William Dougherty, Secern Consulting - Charlotte, USA

SECURWARE 2014 Industry Liaison Chair
Rainer Falk, Siemens AG - München, Germany

SECURWARE 2014 Research/Industry Chair
Mariusz Jakubowski, Microsoft Research, USA

SECURWARE 2014

Committee

SECURWARE Advisory Chairs

Juha Röning, University of Oulu, Finland
Catherine Meadows, Naval Research Laboratory - Washington DC, USA
Petre Dini, Concordia University, Canada / China Space Agency Center - Beijing, China
Reijo Savola, VTT Technical Research Centre of Finland, Finland
Masaru Takesue, Hosei University, Japan
Mariusz Jakubowski, Microsoft Research, USA
Emmanoil Serelis, University of Piraeus, Greece
William Dougherty, Secern Consulting - Charlotte, USA

SECURWARE 2014 Industry Liaison Chair

Rainer Falk, Siemens AG - München, Germany

SECURWARE 2014 Research/Industry Chair

Mariusz Jakubowski, Microsoft Research, USA

SECURWARE 2014 Technical Program Committee

Habtamu Abie, Norwegian Computing Center - Oslo, Norway
Afrand Agah, West Chester University of Pennsylvania, USA
Maurizio Aiello, National Research Council of Italy - IEIIT, Italy
Jose M. Alcaraz Calero, University of the West of Scotland, United Kingdom
Firkhan Ali Bin Hamid Ali, Universiti Tun Hussein Onn Malaysia, Malaysia
Hamada Alshaer, Khalifa University of Science, Technology & Research (KUSTAR), UAE
Claudio Agostino Ardagna, Università degli Studi di Milano, Italy
David Argles, Haven Consulting, UK
George Athanasiou, KTH Royal Institute of Technology, Sweden
Ilija Basicevic, University of Novi Sad, Serbia
Lejla Batina, Radboud University Nijmegen, The Netherlands
Georg T. Becker, University of Massachusetts Amherst, USA
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Francisco Jose Bellido Outeiriño, University of Cordoba, Spain
Malek Ben Salem, Accenture Technology Labs, USA
Jorge Bernal Bernabé, University of Murcia, Spain
Catalin V. Birjoveanu, "Al.I.Cuza" University of Iasi, Romania
Lorenzo Blasi, Hewlett-Packard, Italy
Carlo Blundo, Università di Salern, Italy
Wolfgang Boehmer, Technische Universitaet Darmstadt, Germany
Ravishankar Borgaonkar, Technical University Berlin and Deutsche Telekom Laboratories, Germany
Jérémy Briffaut, ENSI - Bourges, France

Julien Bringer, SAFRAN Morpho, France
Christian Callegari, University of Pisa, Italy
Juan Vicente Capella Hernández, Universidad Politécnica de Valencia, Spain
Hervé Chabanne, Morpho & Télécom ParisTech, France
Hyunseok Chang, Bell Labs/Alcatel-Lucent, USA
Fei Chen, VMware, Inc., USA
Lisha Chen-Wilson, University of Southampton, UK
Feng Cheng, Hasso-Plattner-Institute at University of Potsdam, Germany
Jin-Hee Cho, US Army Research Laboratory Adelphi, USA
Te-Shun Chou, East Carolina University - Greenville, USA
Cheng-Kang Chu, Institute for Infocomm, Singapore
Mario Ciampi, National Research Council of Italy - Institute for High Performance Computing and Networking (ICAR-CNR), Italy
Stelvio Cimato, Università degli studi di Milano - Crema, Italy
David Chadwick, University of Kent, UK
Frédéric Cuppens, Télécom Bretagne, France
Pierre de Leusse, HSBC, Poland
Sagarmay Deb, Central Queensland University, Australia
Mourad Debbabi, Concordia University, Canada
Tassos Dimitriou, Computer Technology Institute, Greece / Kuwait University, Kuwait
Changyu Dong, University of Strathclyde, U.K.
Zheng Dong, Indiana University Bloomington, USA
Safwan El Assad, University of Nantes, France
El-Sayed El-Alfy, King Fahd University of Petroleum and Minerals - Dhahran, KSA
Wael Mohamed El-Medany, University Of Bahrain, Bahrain
Navid Emamdoost, University of Minnesota, USA
Keita Emura, National Institute of Information and Communications Technology (NICT), Japan
David Eyers, University of Otago, New Zealand
Rainer Falk, Siemens AG - München, Germany
Eduardo B. Fernandez, Florida Atlantic University - Boca Raton, USA
Luca Ferretti, University of Modena and Reggio Emilia, Italy
Ulrich Flegel, HFT Stuttgart University of Applied Sciences, Germany
Anders Fongen, Norwegian Defence Research Establishment, Norway
Robert Forster, Edgemount Solutions, USA
Keith Frikken, Miami University, USA
Somchart Fugkeaw, Thai Digital ID Co., Ltd. - Bangkok, Thailand
Amparo Fuster-Sabater, Information Security Institute (CSIC), Spain
Clemente Galdi, Universit`a di Napoli “Federico II”, Italy
Amjad Gawanmeh, Khalifa University of Science, Technology & Research - Sharjah, UAE
Bogdan Ghita, Plymouth University, UK
Danilo Gligoroski, Norwegian University of Science and Technology, Norway
Luis Gomes, Universidade Nova de Lisboa, Portugal
Hidehito Gomi, Yahoo! JAPAN Research, Japan
Pankaj Goyal, MicroMega, Inc., USA
Stefanos Gritzalis, University of the Aegean, Greece
Vic Grout, Glyndŵr University - Wrexham, UK
Yao Guo, Pekin University, China
Bidyut Gupta, Southern Illinois University Carbondale, USA

Gang Tan, Lehigh University, USA
Li Tan, Washington State University, USA
Toshiaki Tanaka, KDDI R & D Laboratories Inc., Japan
Carlos Miguel Tavares Calafate, Universidad Politécnica de Valencia, Spain
Enrico Thomae, operational services GmbH & Co. KG, Germany
Tony Thomas, Indian Institute of Information Technology and Management - Kerala, India
Panagiotis Trimintzios, European Network and Information Security Agency (ENISA), Greece
Raylin Tso, National Chengchi University, Taiwan
Ion Tutanescu, University of Pitesti, Romania
Shambhu Upadhyaya , State University of New York at Buffalo, USA
Miroslav Velev, Aries Design Automation, USA
José Francisco Vicent Francés, University of Alicante, Spain
Calin Vladeanu, "Politehnica" University of Bucharest, Romania
Tomasz Walkowiak, Wrocław University of Technology, Poland
Alex Hai Wang, The Pennsylvania State University, USA
Shiyuan Wang, Google Inc., USA
Wendy Hui Wang, Stevens Institute of Technology - Hoboken, USA
Wenhua Wang, Marin Software Company, USA
Steffen Wendzel, Fraunhofer FKIE, Bonn, Germany
Matthias Wieland, Universitaet Stuttgart, Germany
Wojciech Wodo, Wroclaw University of Technology, Poland
Tzong-Chen Wu, National Taiwan University of Science & Technology, Taiwan
Yongdong Wu, Institute for Infocomm Research, Singapore
Yang Xiang, Deakin University - Melbourne Burwood Campus, Australia
Sung-Ming Yen, National Central University, Taiwan
Xie Yi, Sun Yat-Sen University - Guangzhou, P. R. China
Xun Yi, Victoria University - Melbourne, Australia
Hiroshi Yoshiura, The University of Electro-Communications, Japan
Heung Youl Youm, KIISC, Korea
Amr Youssef, Concordia University - Montreal, Canada
Jun Zhang, Deakin University, Geelong Waurn Ponds Campus, Australia
Wenbing Zhao, Cleveland State University, USA
Yao Zhao, Beijing Jiaotong University, P. R. China
Xinliang Zheng, Frostburg State University, USA
Albert Zomaya, The University of Sydney, Australia

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Embedded Web Device Security

Michael Riegler
IT Solutions
RMTec
Baumgartenberg, Austria
michael.riegler@rmtec.at

Johannes Sametinger
Dept. of Information Systems – Software Engineering
Johannes Kepler University
Linz, Austria
johannes.sametinger@jku.at

Abstract—**Due to the increasing networking of devices and services to the Internet of Things, security requirements are rising. Systems that were previously operated in isolation can be attacked over the Internet today. Industrial control systems often form the core of critical infrastructures. Their vulnerabilities and too lax security management can have fatal consequences. With the help of vulnerability databases and search engines, hackers can get instructions and targets to exploit. Routers, printers, cameras and other devices can be the gateway to the home or corporate network. Cyber criminals can enter sensitive areas through inadequately protected remote access. In a case study of a central water supply control system, we present typical security problems. We show that security vulnerabilities are wide-spread in current embedded web devices and demonstrate that appropriate countermeasures can reduce the attack surface significantly.**

Keywords-web; embedded devices; web security; industrial control systems.

I. INTRODUCTION

Embedded devices increasingly include connectivity as a standard feature, putting them at risk to malicious attack if not secured properly. Some device vendors are offering solutions to protect their embedded devices, including anti-malware technology, access control, data encryption, and real-time threat analysis, as well as maintenance, support, and update/patch services [8]. However, there is insufficient awareness of both device manufacturers and their users about the risks that stem from lacking protection. Take medical devices as an example. Today, even heart pacemakers communicate wirelessly. Parameter settings can be sent to the devices, and usage data including alerts are automatically sent to manufacturers and clinics. If not properly secured, these devices are a threat to patients' privacy as well as to their well-being and even life [12]. Users of devices like personal computers, smartphones as well as operators of web servers are typically aware to some extent about the security risks. Security risks of devices that we use more unobtrusively often go unnoticed. Such devices include printers, routers, and cameras [14]. Their use is widespread, they are connected to the Internet, they often provide web interfaces, and they are often unprotected. On the road to a secure Internet of Things (IoT) [3], we will have to do our homework and provide security as needed to all devices.

In this paper, we will describe security issues of what we call embedded web devices, i.e., embedded devices with web access. We will outline the importance of the security of such devices, demonstrate how neglected it is, and also present a case study of a water supply facility. The paper is structured as follows. Section II introduces embedded web devices. In Section III, we outline security aspects. Risks to industrial control systems are described in Section IV. Section V explains how vulnerable web devices can be found on the Internet. A case study is given in Section VI. Related work and a conclusion follow in Sections VII and VIII, respectively.

II. EMBEDDED WEB DEVICES

The Internet has started as a DARPA project, interconnecting computers through which everyone could quickly access data and programs from any site. Today's Internet provides access to a plethora of information from not just computers – back then only mainframes were available – but from devices like smartphones, and television sets. Additionally, access is not limited to other computers but is increasingly available to other devices like printers, routers or webcams. Web devices are any devices that are connected to the Internet via the Hypertext Transfer Protocol (HTTP) protocol, including web servers, personal computers and smartphones. Embedded web devices are with a different focus than just providing and retrieving information on the web. We have already mentioned printers, routers, and webcams. Additional examples include network devices, sensors in smart homes, smart meters in smart grids, smart TVs, etc.

A. Embedded Web Servers

Web servers are running on a combination of hardware and software. They deliver web content to be accessed through the Internet. Embedded web servers are components of systems that, like web servers, communicate via the HTTP protocol. Typically, they provide a thin client interface for traditional applications. Lightweight web servers work with small resources and are often used in embedded systems. Examples include Barracuda [41], Mongoose [26], Appweb [2], and RomPager [33]. Embedded web servers are used for special purpose and are not as extensive as major web servers like Apache [1] and Microsoft's IIS [25].

Embedded web servers are important for the IoT. Today computers and the Internet are almost completely dependent on humans for information. Radio-frequency Identification (RFID) and sensor technology enable computers to observe

 ISBN: 978-1-61208-376-6

and identify the world without the limitations of data entered by humans [3]. IoT refers to uniquely identifiable objects, e.g., sensors having an Internet Protocol (IP) address, and their virtual representations in the Internet.

B. ICS – Industrial Control Systems

The term Industrial Control System (ICS) encompasses several types of control systems that are used in industrial production. Examples are Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and Programmable Logic Controllers (PLC). ICSs are instrumental in the production of goods and in the provision of essential services, e.g., to monitor and control processes like gas and electricity distribution or water treatment. The largest group of ICS is SCADA. For example, all nine Austrian Danube power plants are controlled centrally from Vienna. The stations are connected with optical fibers to the central control room and cannot easily be accessed over the Internet [42].

C. CPS – Cyber Physical Systems

Embedded systems contain computer systems with dedicated functions within larger mechanical or electrical systems, often having real-time computing constraints [13]. In Cyber Physical Systems (CPS) computational elements collaborate to control physical entities. CPSs have a tight integration of cyber objects and physical objects. They can be systems at various scales, e.g., large smart bridges with fluctuation detection and responding functions, autonomous cars, and tiny implanted medical devices [15].

III. SECURITY

Security is about protecting information and information systems, including embedded devices, from unauthorized access and use. The core goals are to retain confidentiality, integrity and availability of information. Often used terms include IT security, network security, computer security, web security, mobile security, and software security. They describe different, but sometimes overlapping aspects of reaching the above mentioned core goals. For example, software security is "the idea of engineering software so that it continues to function correctly under malicious attack" [23], while network security involves the isolation and protection of assets via firewalls, demilitarized zones and intrusion detection systems.

A. Threats, Vulnerabilities, Risks

Threats refer to sources and means of particular types of attacks. Vulnerabilities are security bugs or flaws in a system that allow successful attacks. A security risk is the likelihood of being targeted by a threat. The most important potential threats to be addressed can be determined with a risk assessment. We can assess risks by enumerating the most critical and most likely threats, by evaluating their levels of risk as a function of the probability of a threat and the associated cost if the threat becomes true. Secure devices have to continue to function correctly even if under malicious attack. Vulnerabilities are mistakes in devices, typically in software, which can be directly used to gain access to the device. They pose a threat to the device itself, to the information it contains, to other devices it communicates with, and to the environment that it manipulates.

TABLE II. SECURITY GOALS IN IT AND ICS SYSTEMS

Priority	IT system	ICS system
1	Confidentiality	Availability
2	Integrity	Integrity
3	Availability	Confidentiality

B. ICS Security

ICSs are often found in critical infrastructures, thus have a high need for security. A typical information network will prioritize its security objectives as CIA, i.e., first confidentiality and integrity, and then availability. Industrial control systems often have a high need for availability, reversing the security objectives for most control entities [8]. Table 1 shows these different security goals.

There are other crucial differences that have an influence on the security of these systems. Table 2 summarizes some of these differences. For example, ICSs are usually real-time systems, whereas in IT systems delays are acceptable most of the time. Also, an ICS has to run continuously and cannot simply be rebooted. More details are given by Stouffer et al. [36].

According to a security survey, 70% of SCADA system operators consider the risks to their systems to be high to severe, and 33% suspect they may have had incidents [21]. While IT security is not entirely different from ICS security, there are several differences; see [22]:

- ICS security failures often have physical consequences, thus having more severe and immediate impact.
- ICS security failures can easily and wrongly be interpreted as traditional maintenance failures, making them more difficult to diagnose and remedy.

Security management of an ICS is often much more difficult than of a regular IT system. They more often rely on

TABLE I. DIFFERENCES BETWEEN TYPICAL IT AND ICS SYSTEMS

Category	IT system	ICS system
Performance	Delay acceptable	Real-time
Availability	Reboot acceptable	24 x 7 x 365
Risk	Data confidentiality	Human safety
Interaction	Less critical	Critical
System	Standard OS	Proprietary OS
Resources	Sufficient	Limited
Lifetime	3-5 years	5-20 years
Support	Diversified	Single vendor
Location	Local, easy accessible	Remote, isolated
Virus protection	Standard	Complex

 ISBN: 978-1-61208-376-6

old systems that cannot be patched or upgraded any more. They often do not have a suitable testing environment. They also often require frequent remote access with commands that must not be blocked for safety or production issues [22].

IV. RISKS

Today, many systems are connected to the Internet, even though they were not originally intended for that purpose. Additionally, it has been shown that even systems without any connection to the outside world can be at risk [35]. This can be done by implanting tiny transceivers on hardware parts like USB plugs or small circuit boards in a device.

A. Software Bugs

Prominent software security bugs include buffer overflows, SQL injections and cross-site scripting. They can be exploited in any connected device, embedded or not. There are many examples, where these bugs have occurred and caused damage. While security bugs are problems at the implementation level, security flaws are located at the architecture or design level. A list of the most widespread and critical errors that can lead to serious vulnerabilities in software is presented in [39]. These errors are often easy to find, easy to exploit and often dangerous. In many cases, they allow attackers to steal and modify data, as well as to run arbitrary code on the attacked machine. The Open Web Application Security Project (OWASP) is a not-for-profit organization focused on improving the security of software. They regularly publish a list with the top 10 security bugs with the goal to raise security awareness [40].

B. Back Doors

Reverse engineering of firmware often reveals back doors to devices. Developers often have implemented hard coded user names and passwords for debugging and maintenance purposes. For example, two supposedly hidden user accounts are in the Sitecom WLM-3500 router. The manufacturer has released a new version of the firmware where these accounts were disabled, but thousands with the old version are still accessible via the Internet [30]. Other network devices like webcams or printers often suffer from similar problems. These devices are usually used for years without getting too much attention from their owners. But they can provide an ideal entry for malicious attackers. For example, thousands of webcams have been shown to be accessible via back doors at the Black Hat 2013 conference [14].

C. Configurations

Insecure default configurations make it easier for cyber criminals to enter systems. Default passwords are publicly known and are published on websites like [7][41].

Default running services like Universal Plug and Play (UPnP) are often unused and increase the risk of an attack. Google Hacking and Shodan simplify the search for such insecure systems; see Section 5. With the mobile exploitation framework Routerpwn [34] it is possible to get access to routers, switches and access points from over 30 manufacturers. The 150+ provided exploits could be executed over the browser locally and remote over the Internet. Predefined access keys especially for the wireless network can be discovered when the calculation methods are known.

D. Other Risks

Automatic update functions, as we know them from desktop operating systems, are not the general rule for ICSs. Firmware manufacturers usually take their time with the provision of updates. The fact that such updates are often provided for free apparently does not motivate to more frequent updates. Sometimes, updates are not even provided at all. But even if updates are available, private customers may not know about them or may simply not have any interest in installing them. Professional users may refrain from updates when they cannot be performed without shutting down the system. Updating a system can lead to crashes or may need reconfigurations of parts or even the entire system. System crashes can disrupt or paralyze business operations. In addition to technical risks, legal risks may also be involved.

Attacks can be a problem for managers. Insufficient security management can lead to personal responsibility. The company may use their staff or service providers to reach the security goals. However, the final responsibility remains with the company. Unsecured or poorly secured home networks have already led to court proceedings. In case of damage, device misconfiguration and insufficient secured smart home solutions can result in problems with the insurance.

E. Countermeasures

Measures for enhanced security are manifold and have to be taken at various levels. Measures at the technical level include software security, encrypted communication, authentication, authorization as well as firewalls and demilitarized zones. At the organizational level, example countermeasures include password management, patch and update management, backup, and security awareness. Requirements for establishing, implementing, maintaining and continuously improving an Information Security Management System (ISMS) are given in the International Organization for Standardization (ISO) 27000 standard. The ISO 27000 series of standards have been reserved for information security matters [38]. ISO 27000 certification demonstrates an organization's ability to provide information assurance services utilizing best practice processes and methodologies.

V. SEARCHING FOR VULNERABLE WEB DEVICES

Search engines are not only used for information search, they are also used to discover embedded web devices and data that were not meant for the public. Cyber criminals use this for victim search and attack preparation, because many of these devices provide sensitive information. The Google and Shodan search engines provide plenty of information that is useful for attackers.

A. Google Hacking

Johnny Long, a pioneer in the field of Google Hacking, has used Google to find security holes from websites and everything else on the web. Advanced search operators can be used to find passwords, internal price lists, confidential

 ISBN: 978-1-61208-376-6

documents, as well as vulnerable systems. Hacker groups like *LulzSec* and *Anonymous* have used special search queries to break into foreign systems [5]. For example, the following search term can be used to find authentication data for Virtual Private Networks (VPN).

```
! Host=*.* intext:enc_UserPassword=* ext:pcf
```

The shown example works in a similar way with other search engines like Bing or Yahoo. These special searches are called Google Dorks and are collected in the Google Hacking Database; see [9]. The database has grown to include several thousand entries. The National Security Agency (NSA) has also been using special search operators for their Internet research. A previously confidential guide with over 640 pages has been published by the NSA [28].

B. Shodan Hacking

Shodan is a search engine that is specialized to find online devices like webcams, routers and printers, as well as industrial control systems from power plants, which are connected to the Internet. Even nuclear power plants can be found with Shodan [11]. The following search term provides industrial control system from Siemens in the United States.

```
"Simatic+S7" country:US
```

The Stuxnet worm had targeted these systems in Iran. Searching for webcams with Shodan is quite popular. For example, a search for “netcam” results in thousands of hits to TRENDnet webcams with a backdoor to access the video stream.

C. Exploits and Vulnerabilities

Besides the search for vulnerable web devices, it is also possible to search for their exploits and vulnerabilities. Many websites like the Exploit Database [10], Metasploit [24], Packet Storm [29], and others provide information, tools and sample code to exploit web devices. For example, the Exploit Database provides 20,000+ exploits for a variety of programs and systems. New entries are added almost daily. The global web application vulnerability search engine PunkSPIDER [32] scans websites for vulnerabilities and provides this highly sensitive information for the public. Script kiddies can use this information to attack computer systems and networks or to hack websites.

VI. CASE STUDY: WATER SUPPLY FACILITY

Water supply facilities are used to take countermeasures to the variability and intensity of rainfall and to guarantee water supply for a specific geographic region like a town or city. To ensure water supply, water from wells or rainfall is pumped into big containers that make sure that water is available during dry seasons. Maintenance of a water supply facility includes checking of water levels, proper working of water pumps, checking for existing leaks. Recently, computerization has helped to automate these processes. The following case study deals with facilities that ensure water supply for several thousand people in a small community. We have developed a monitoring and control system that allows administrators to access all water supply facilities, to access remote cameras at these sites, and to turn on/off pumps. Additionally, various sensors have been installed at the remote sites. For example, if a door is opened or water is raising or falling above or below a predefined level, then an alarm will be raised by sending an email or text message to the administrators. Administrators may then check the status of facilities remotely by turning on video cameras, and make corrections by turning on/off pumps, etc.

We will depict the general architecture of the facility, and show threats due to computerization as well as countermeasures that were actually taken.

A. Architecture

Several water supply facilities are connected over the Internet to a control server. The control server provides a web interface that allows all systems to be monitored and controlled centrally. Figure 1 shows the general architecture of the system. Each water supply facility includes a control system that controls several pumps in the facility. Additionally, webcams are used to allow users to inspect the facility and also to read several gauges analogously. Features of the system include turning on/off pumps, defining on/off times for pumps, turning on/off lights, and choosing among various predefined operation modes.

B. Implementation Aspects

At each water supply location there is a PLC called Loxone mini server [20], called control system in Figure 1. Every mini server has an embedded web server with an interface to control connected pumps through HTTP requests, thus, allowing pumps to operate centrally from the control server. The mini servers monitor their area and send operational states to the control server. They also transmit statistical data for further evaluations. In addition to the mini servers, webcams, i.e., INSTAR IP cameras [16] have been installed at some locations. Because these cameras can be controlled through their web interface, they are also integrated in the

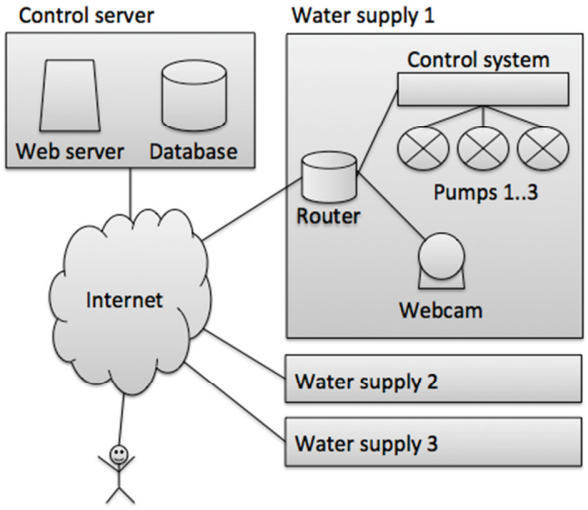


Figure 1. Architectural overview

 ISBN: 978-1-61208-376-6

central web interface of the control server. Thus, the mini server and the IP cameras can receive commands over the Internet. The routers at the water supply facilities use port forwarding. As the routers have no fixed IP addresses assigned, communication is based on domain names. Dynamic Domain Name Service (DDNS) is used to react to changed IP addresses.

On the server side there is a user interface for the central control and a web interface to receive data from the facilities. Both the user interface and the web interface have been programmed with PHP: Hypertext Preprocessor (PHP) [31] and use a MySQL database [27]. Data received from the facilities is checked for plausibility and then stored in the database. jQuery mobile [18] is used to create a user-friendly interface for various devices. Charts are generated with the JavaScript library d3.js [6].

C. Threats

Both the Loxone mini server and the cameras provide sensitive information in their service banner without authentication. Additionally, the firmware version of the mini server is revealed. The cameras provide even more sensitive information like device ID, firmware, etc., with the simple link http://ip-camera/get_status.cgi. Because the mentioned systems use unencrypted protocols like HTTP and File Transfer Protocol (FTP) for communication, the submitted credentials can easily be read through man-in-the-middle attacks. As systems are accessible via the Internet, they can also become victims of denial of service attacks. This can disrupt the water supply. Bugs in software and in firmware of the used components and backdoors can lead to dangerous situations. Outdated software and firmware versions increase the risk. If cyber criminals enter a system, they can manipulate values and settings. This could cause hidden error messages or simulated defects. Moreover, safety mechanisms could be overridden and endanger persons. If the IP cameras are deactivated it is no longer possible to monitor the physical access to the facility. Turning off the pumps can paralyze water supply. Repeatedly switching the pumps on and off within short periods of time can destroy them. When all pumps are at full power, the pressure in the lines can increase to a level so that they could burst.

D. Countermeasures

Because it is not possible with the current firmware of the devices to hide sensitive information and use secure protocols and services, the access through port forwarding on the router is only allowed for stored IP addresses as those from the control server. Thereby, access for Shodan, Google and others is denied. It is interesting to note that a Shodan search results in more 10,000+ Loxone systems. The used webcam can be found 700,000+ times.

The access to the web interface of the central control server is protected with digest authentication and blocks IP addresses after a certain number of incorrect login attempts. Furthermore, HTTP Secure (HTTPS) is used for an encrypted data communication between the control server and the clients. For further analyzes every user activity is logged.

An additional countermeasure is the secure data communication over the Internet with VPNs. VPN requires authentication and provides encryption, so that data is transmitted through a secure channel between a facility and the control server. Thus, insecure protocols like HTTP and FTP can be secured, and man-in-the-middle attacks can be prevented. The most important measure is to increase security awareness of users. Each technical measure is useless if users are careless. Passwords on post-it notes, insecure storage of access devices and evil apps can cause problems. Therefore, it is important to train users.

VII. Related Work

ISE researchers discovered critical security vulnerabilities in numerous routers for small offices and small home offices as well as in wireless access points. The found vulnerabilities allowed remote attackers to take full control of the device's configuration settings. Some even allowed a direct authentication bypass. Attackers were able to intercept and modify network traffic as it entered and left the network [17]. The authors reported that the rich service and feature sets implemented in these routers, e.g., Server Message Block (SMB), Network Basic Input/Output System (NetBIOS), HTTP(S), FTP, UPnP, Telnet, come at a significant cost to security. The incorporation of additional services typically exposes additional attack surfaces that malicious adversaries can use to gain a foothold in a victim's network.

Leverett examined results over two years through the Shodan search engine [19]. He located, identified and categorized more than 7500 such devices, i.e., Heating, Ventilation, and Air Conditioning (HVAC) systems, building management systems, meters, and other industrial control devices or SCADA servers. He concluded that combined with information from exploit databases, remote attacks on selected devices could be carried out or networks could be identified for further reconnaissance and exploitation.

A recent analysis of a widespread compromise of routers for small offices and home offices has been reported in [37]. Attackers were altering the device's Domain Name Service (DNS) configurations in order to redirect DNS requests of their victims to IP addresses and domains controlled by the attackers.

VIII. Conclusion

Embedded devices increasingly get connected to the Internet. If not properly secured, they are at risk to malicious attack. Malicious users find tempting targets across various markets, including consumer electronics, automobiles, medical equipment, and even military hardware. We have taken a closer look at devices that are accessible through the Internet today, but that are often not secured properly. We have also shown in a small case study that these devices, if unsecured, can pose a threat not just to the privacy of individuals and organizations, but also to the proper functioning of critical infrastructure. Appropriate countermeasures can considerably increase an attacker's effort needed to compromise a system with reasonable expenses on the defender's side.

 ISBN: 978-1-61208-376-6

REFERENCES

[1] Apache HTTP Server Project, http://httpd.apache.org. [retrieved: September, 2014]

[2] App-web, http://appwebserver.org/. [retrieved: September, 2014]

[3] K. Ashton, "That 'Internet of Things' Thing", FRiD Journal, Jun 22, 2009. http://www.rfidjournal.com/articles/view?4986 [retrieved: September, 2014]

[4] Barracuda Web Server, https://realtimelogic.com/products/barracuda-web-server/. [retrieved: September, 2014]

[5] F. Brown and R. Ragan, "Pulp Google Hacking: The Next Generation Search Engine Hacking Arsenal", Black Hat 2011. https://media.blackhat.com/bh-us-11/Brown/BH_US_11_BrownRagan_Pulp_Google.pdf [retrieved: September, 2014]

[6] D3.js - Data-Driven Documents, http://d3js.org. [retrieved: September, 2014]

[7] DefaultPassword, http://default-password.info. [retrieved: September, 2014]

[8] DHS, "Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-in-Depth", Department of Homeland Security, Control Systems Security Program, National Cyber Security Division, October 2009. http://ics-cert.us-cert.gov/sites/default/files/recommended_practices/Defense_in_Depth_Oct09.pdf [retrieved: September, 2014]

[9] Exploit Database, "Google Hacking-Database", http://exploit-db.com/google-dorks. [retrieved: September, 2014]

[10] Exploit Database, http://www.exploit-db.com. [retrieved: September, 2014]

[11] D. Goldman, "Shodan: The scariest search engine on the Internet", CNNMoney. April 2013. http://money.cnn.com/2013/04/08/technology/security/shodan/index.html. [retrieved: September, 2014]

[12] D. Halperin, T. S. Heydt-Benjamin, K. Fu, T. Kohno, and W. H. Maisel, "Security and privacy for implantable medical devices", IEEE Pervasive Computing, Special Issue on Implantable Electronics, January 2008.

[13] S. Heath, "Embedded systems design", EDN series for design engineers, (2 ed.). Newnes, 2nd edition, ISBN 978-0-7506-5546-0, 2003.

[14] C. Heffner, "Exploiting Surveillance Cameras - Like a Hollywood Hacker", Black Hat, July 2013. https://media.blackhat.com/us-13/US-13-Heffner-Exploiting-Network-Surveillance-Cameras-Like-A-Hollywood-Hacker-Slides.pdf [retrieved: September, 2014]

[15] F. Hu, Cyber-Physical Systems: Integrated Computing and Engineering Design, CRC Press, ISBN 978-1466577008, 2013.

[16] INSTAR IP cameras, http://instar.com. [retrieved: September, 2014]

[17] ISE – Independent Security Evaluators, "Exploiting SOHO Router Services", Technical Report, July 2013. http://securityevaluators.com/content/case-studies/routers/soho_techreport.pdf [retrieved: September, 2014]

[18] jQuery mobile, http://jquerymobile.com. [retrieved: September, 2014]

[19] E. P. Leverett, "Quantitatively Assessing and Visualising Industrial System Attack Surfaces", University of Cambridge, PhD Thesis, 2011. http://www.cl.cam.ac.uk/~fms27/papers/2011-Leverett-industrial.pdf [retrieved: September, 2014]

[20] Loxone Home Automation, http://www.loxone.com. [retrieved: September, 2014]

[21] M. E. Luallen, "SANS SCADA and Process Control Security Survey", A SANS Whitepaper, February 2013. http://www.sans.org/reading-room/analysts-program/sans-survey-scada-2013. [retrieved: September, 2014]

[22] T. Macaulay and B. L. Singer, "Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS", Auerbach Publications, ISBN 978-1439801963, 2011.

[23] G. McGraw, "Software Security", IEEE Security & Privacy, vol. 2, no. 2, pp. 80-83, March-April 2004.

[24] Metasploit, http://www.metasploit.com. [retrieved: September, 2014]

[25] Microsoft Internet Information Services, http://www.iis.net. [retrieved: September, 2014]

[26] Mongoose – easy to use web server, http://code.google.com/p/mongoose. [retrieved: September, 2014]

[27] MySQL, http://www.mysql.com. [retrieved: September, 2014]

[28] NSA, "Untangling the Web-A Guide To Internet Research", National Security Agency, 2007, released in 2013. http://www.nsa.gov/public_info/files/Untangling_the_Web.pdf. [retrieved: September, 2014]

[29] Packet Storm, http://packetstormsecurity.com. [retrieved: September, 2014]

[30] R. Paleari, "Sitecom WLM-3500 back-door accounts". Emaze Networks S.p.A., 2013, http://blog.emaze.net/2013/04/ sitecom-wlm-3500-backdoor-accounts.html. [retrieved: September, 2014]

[31] PHP: Hypertext Preprocessor, http://php.net. [retrieved: September, 2014]

[32] PunkSPIDER, http://punkspider.hyperiongray.com. [retrieved: September, 2014]

[33] RomPager Embedded Web Server, http://allegrosoft.com/ embedded-web-server. [retrieved: September, 2014]

[34] Routerpwn, http://routerpwn.com. [retrieved: September, 2014]

[35] D. E. Sanger and T. Shanker, "N.S.A. Devises Radio Path-way Into Computers", The New York Times, Jan. 14, 2014. http://www.nytimes.com/2014/01/15/us/nsa-effort-pries-open-computers-not-connected-to-internet.html. [retrieved: September, 2014]

[36] K. Stouffer, J. Falco, and K. Scarfone, "Guide to Industrial Control Systems (ICS) Security", NIST Special Publication 800-82, Revision 1, May 2013. http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf. [retrieved: September, 2014]

[37] Team Cymru, "SOHO Pharming: The Growing Exploitation of Small Office Routers Creating Serious Risk", Whitepaper, February 2014. https://www.team-cymru.com/ReadingRoom/Whitepapers/2013/TeamCymruSOHOPharming.pdf. [retrieved: September, 2014]

[38] The ISO 27000 Directory, http://www.27000.org. [retrieved: September, 2014]

[39] The MITRE Corporation, "2011 CWE/SANS Top 25 Most Dangerous Software Errors", 2011. http://cwe.mitre.org/top25/. [retrieved: September, 2014]

[40] The Open Web Application Security Project, "OWASP Top Ten - 2013 – The Ten Most Critical Web Application Security Risks", 2013. https://owasp.org/index.php/Top_10#OWASP_Top_10_for 2013. [retrieved: September, 2014]

[41] Unknown Password, http://unknownpassword.com. [retrieved: September, 2014]

Verbund AG, "Freudenau Power Plant to Become the Danube's Central Nervous System", Press Release, July 2011. http://verbund.com/cc/en/news-media/news/2011/07/07/freudenau. [retrieved: September, 2014]

 ISBN: 978-1-61208-376-6

Design Issues in the Construction of a Cryptographically Secure Instant Message Service for Android Smartphones

Alexandre Melo Braga, Daniela Castilho Schwab
Centro de Pesquisa e Desenvolvimento em Telecomunicações (Fundação CPqD)
Campinas, São Paulo, Brazil
{ambraga,dschwab}@cpqd.com.br

***Abstract*—This paper describes design and implementation issues concerning the construction of a cryptographically secure instant message service for Android devices along with its underlying cryptographic library. The paper starts by discussing security requirements for instant message applications, and proceeds to the architecture of cryptographic components and selection of cryptographic services. Concerning this last point, two sets of services were implemented: one based only on standardized algorithms and other based solely on non-standard cryptography.**

Keywords-Cryptography; Security; Android; Instant Message.

I. INTRODUCTION

Currently, the proliferation of smartphones and tablets and the advent of cloud computing are changing the way software is being developed and distributed. Contemporary to this context change, the use in software systems of security functions based on cryptographic techniques is increasing as well.

The scale of cryptography-based security in use today has increased not only in terms of volume of encrypted data, but also relating to the amount of applications with cryptographic services incorporated within their functionalities. In addition to the traditional use cases historically associated to cryptography (e.g., encryption/decryption and signing/verification), there are several new usages, such as privacy preserving controls, bringing diversity to the otherwise known universe of threats to cryptographic software.

This paper discusses the construction of a mobile application for secure instant messaging on Android devices and a cryptographic library intended to support it. The paper focuses on design decisions as well as on implementation issues. This work contributes to the state of the practice by discussing the technical aspects and challenges of cryptographic implementations on modern mobile devices. The contributions of this paper are the following:

- The design of cryptographically secure instant message service;
- The elicitation of strong security requirements for cryptographic key negotiation over instant messages;
- The selection of a minimum set of standard cryptographic services capable of fulfill the requirements;
- The selection of non-standard cryptography in order to replace the whole standard algorithm suite.

The remaining parts of the text are organized as follows. Section II presents related work. Section III details requirements and design decisions. Section IV describes implementation aspects. Section V outlines improvements under development. Section VI concludes this text.

II. RELATED WORK

Nowadays, secure phone communication does not mean only voice encryption, but encompasses a plethora of security services built over the ordinary smartphone capabilities. To name just a few of them, these are SMS encryption, Instant Message (IM) encryption, voice and video chat encryption, secure conferencing, secure file transfer, secure data storage, secure application containment, and remote security management on the device, including management of cryptographic keys. All these security applications have been treated by an integrated framework [3] as part of a research project [4].

This section focuses on security issues of IM protocols and applications, as well as cryptography issues on Android devices.

A. Security issues in IM protocols and applications

The work of Xuefu and Ming [7] shows the use of eXtensible Messaging and Presence Protocol (XMPP) for IM on web and smartphones. Massandy and Munir [12] have done experiments on security aspects of communication, but there are unsolved issues, such as strong authentication, secure storage, and implementation of good cryptography, as shown by Schrittwieser et al.[39].

It seems that the most popular protocol for secure IM in use today is the Off-the-Record (OTR) Messaging [32], as it is used by several secure IM apps. OTR Messaging handshake is based upon the SIGMA key exchange protocol [15], a variant of Authenticated Diffie-Hellman (ADH) [45], just like Station-to-Station (STS) [6][46], discussed in further detail at Section IV.

A good example of security issues found in current IM software is a recently discovered vulnerability in WhatsApp [36]. The vulnerability resulting from misuse of the Rivest Cipher 4 (RC4) stream cipher in a secure communication

 ISBN: 978-1-61208-376-6

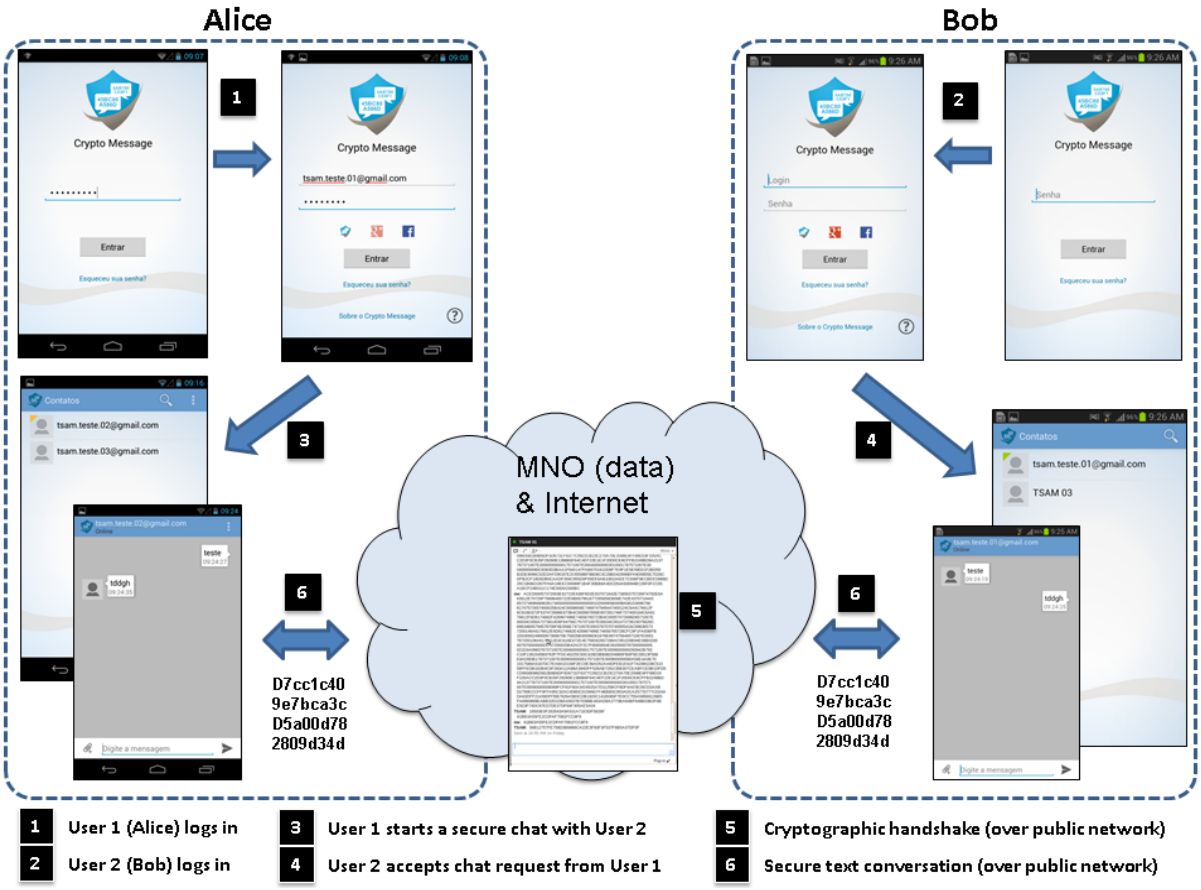


Figure 1. Basic flow of the secure exchange of instant messages.

protocol allowed the decryption, by a malicious third party able to observe conversations, of encrypted messages exchanged between two WhatsApp users. The issues related to this vulnerability are twofold. First, the incorrect use of RC4 stream cipher in place of a block cipher. Second, the reuse of cryptographic keys in both communication directions. The reuse of keys in a stream cipher and the existence of fixed parts, such as headers, at the communication protocol enabled the partial discovery of cryptographic keys.

B. Cryptography issues on Android devices

A recent study [2] showed that despite the observed diversity of cryptographic libraries in academic literature, this does not mean those implementations are publicly available or ready for integration with third party software. In spite of many claims on generality, almost all of them were constructed with a narrow scope in mind and prioritizes academic interest for non-standard cryptography. Furthermore, portability to modern mobile platforms, such as Android, is a commonly neglected concern on cryptographic libraries, as that evaluation has shown [2].

Moreover, there are several misuse commonly found on cryptographic software in use today. According to a recent study [24], the most common misuse of cryptography in mobile devices is the use of deterministic encryption, where a symmetric cipher in Electronic Code Book (ECB) mode appears mainly in two circumstances: Advanced Encryption Standard (AES) in ECB mode of operation (AES/ECB for short) and Triple Data Encryption Standard in ECB mode (TDES/ECB). There are cases of cryptographic libraries in that ECB mode is the default option, automatically selected when the operation mode is not explicitly specified by the programmer. A possibly worse variation of this misuse is the Rivest-Shamir-Adleman (RSA) cryptosystem in Cipher-Block Chaining (CBC) mode with Public-Key Cryptography Standards Five (PKCS#5) padding (without randomization), which is also available in modern cryptographic libraries, despite of been identified more than 10 year ago [34].

Another frequent misuse is hardcoded Initialization Vectors (IVs), even with fixed or constant values [34]. A related misuse is the use by the ordinary programmer of hardcoded seeds for PRNGs [24].

A common misunderstanding concerning the correct use of IVs arises when (for whatever reason) programmers need to change operation modes of block ciphers. For instance, the Java Cryptographic API [20] allows operation modes to be easily changed, but without considering IV requirements.

 ISBN: 978-1-61208-376-6

According to a NIST standard [30], CBC and Cipher feedback (CFB) modes require unpredictable IVs. However, Output feedback (OFB) mode does not need unpredictable IVs, but it must be unique to each execution of the encryption operation. Considering these restrictions, IVs must be both unique and unpredictable, in order to work interchangeably with almost all common operation modes of block ciphers. The Counter (CTR) mode requires unique IVs and this constraint is inherited by authenticated encryption with Galois/Counter mode (GCM) [31].

The two remarkable differences between the prototype described in this text and the related work are the following. First, the prototype uses STS protocol and its variants to accomplish authenticated key agreement. This has the benefit of facilitating protocol extension to use alternative cryptographic primitives. Second, authenticated encryption is the preferred encryption mechanism to protect messages, so the burden of IV management is minimized.

III. REQUIREMENTS FOR SECURE IM APPLICATIONS

This section describes the primary usage scenario of a mobile application for secure IM, as well as the selection of cryptographic services required by that application. This scenario illustrates the requirements elicitation that guided the design of the library.

A. Primary usage scenario for IM applications

The prototype for cryptographically secure, end-to-end communication operates on a device-to-device basis, exchanging encrypted IM via standard transport protocols. In the following text, the prototype is called CryptoIM.

CryptoIM implements the basic architecture used by all IM applications, using the standard protocol XMPP [35] at the transport layer. The application then adds a security layer to XMPP, which is composed of a cryptographic protocol for session key agreement and cryptographic transaction to transport encrypted messages. Therefore, CryptoIM is able to transport encrypted information through public services offered by providers such as Google (for Gtalk or HangOut) and Whatsapp.

The usage scenario that inspired the implementation of CryptoIM was to secure end-to-end communication, as described before. The two sides of communication (Alice and Bob) want to use their mobile device to exchange confidential and authentic messages. In CryptoIM, when a user selects a contact she wants to talk to, the protocol for secure conversation is initiated behind the scenes. The following action flow can be observed in Figure 1:

1) User 1 enters the application;
2) User 2 enters the application;
3) User 1 opens a conversation with User 2;
4) User 2 accepts the conversation;
5) Security negotiation occurs;
6) Secure conversation proceeds as expected.

This basic flow represents the simplest behavior needed for secure conversation. A secure conversation can be canceled by either party by sending a cancellation message.

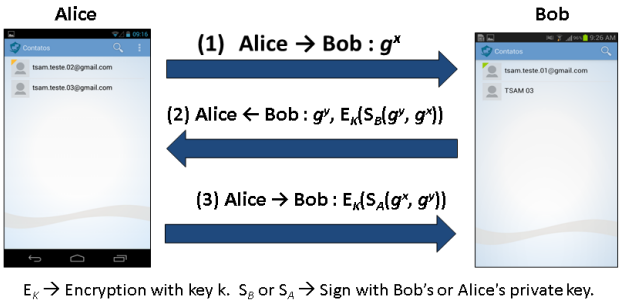


Figure 2. Station to Station (STS) protocol.

The security negotiation phase is indeed a protocol for key agreement, as illustrated by Figure 2.

B. Selection of cryptographic services

To accomplish the above mentioned scenario, Alice and Bob choose to use cryptographically secure communication with the following general requirements:

- An authentication mechanism of individual messages;
- An encryption algorithm and modes of operation;
- A key agreement protocol;
- A mechanism to protect cryptographic keys at rest.

In addition to a unique key for each conversation, that ensures security in the exchange of messages, a unique IV is generated for each exchanged message. To ensure that the protocol was followed in a transparent manner without user interference, automated messages were sent behind the scenes, so that the user does not see the exchange of messages for key negotiation. This prevents user from trying to interfere in the key agreement process.

To avoid known security issues in instant messaging applications [36][39], the key agreement protocol must provide the security properties described below [47]:

a) *Mutual authentication of entities*. For this property to be sustained in the protocol, signed messages must include the identities of both participants;
b) *Mutually authenticated key agreement*. The shared secret is a result of the underlying Key Agreement (KA) protocol. The freshness or novelty of the secret is the result of choosing random values for each conversation. The authenticity of secret sharing is guaranteed by digital signatures;
c) *Mutual confirmation of secret possession*. The decryption using a derived secret key confirms the possession of secret and evidences that the entity with knowledge of the secret is the same one signing the agreement messages. After a run of the protocol, the two participants observe each other performing encryption with shared secret key;
d) *Perfect Forward Secrecy (PFS)*. If a private key is compromised at some point in time, the security of session keys previously established is not affected. It is important for the maintenance of this property that the intermediate values are discarded and safely deleted at the end of a protocol run;

 ISBN: 978-1-61208-376-6

e) *Anonymity*. If the certificates are encrypted and the identities were omitted in the body of messages, a third party observing the communication network can not directly identify the interlocutors.

The cryptographic library supporting CryptoIM was designed to meet each one of these general requirements, resulting in an extensive implementation.

IV. DESCRIPTION OF THE IMPLEMENTATION

As a general goal, the CryptoIM cryptographic library is intended to be used in the protection of cryptographically secure communication via mobile devices. In order to be useful, the cryptographic library had to accomplish a minimum set of functional requirements. Each functional requirement generated a set of non-functional or supplementary requirements, mostly related to correctness of algorithms, compliance to industry standards, security, and performance of the implementation.

In order to facilitate the portability of the cryptographic library for mobile devices, in particular for the Android platform, the implementation was performed according to standard cryptographic Application Programming Interface (API) for Java, the Java Cryptographic Architecture (JCA), its name conventions, and design principles [16][20]-[23].

Once JCA was defined as the architectural framework, the next design decision was to choose the algorithms minimally necessary to implement a scenario of secure communication via mobile devices. The choice of a minimum set was an important design decision in order to provide a fully functional Cryptographic Service Provider (CSP) in a relatively short period of time. This minimalist construction had to provide the follow set of cryptographic functions:

a) A symmetric algorithm to be used as block cipher, along with the corresponding key generation function, and modes of operation and padding;
b) An asymmetric algorithm for digital signatures, along with the key-pair generation function. This requirement brings with it the need for some sort of digital certification of public keys;
c) A one-way secure hash function. This is a support function to be used in MACs, signatures and PRNGs;
d) A Message Authentication Code (MAC), based on a secure hash or on a block cipher;
e) A key agreement mechanism or protocol to be used by communicating parties that have never met before, but need to share an authentic secret key;
f) A simple way to keep keys safe at rest and that does not depend on hardware features;
g) A Pseudo-Random Number Generator (PRNG) to be used by all the key generation functions.

The current version of this implementation is illustrated by Figure 3 and presents the cryptographic algorithms and protocols described in the following paragraphs. The figure shows that frameworks, components, services and applications are all on top of JCA API. CryptoIM's Cryptographic Service Provider (CSP) is in the middle, along with BouncyCastle and Oracle providers. Arithmetic libraries are at the bottom.

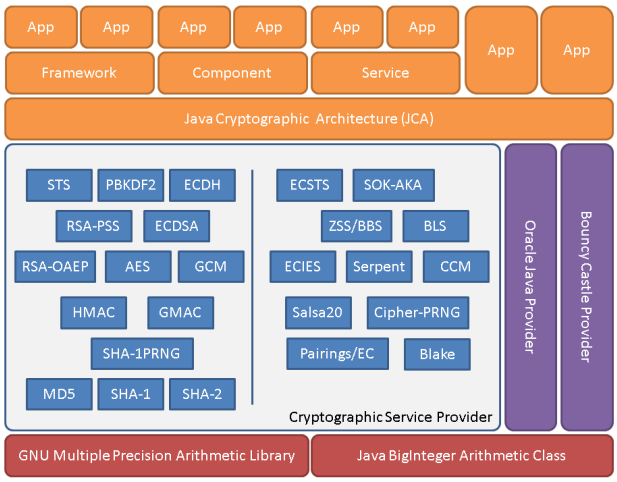


Figure 3. Cryptographic Service Provider Architecture.

Figure 3 shows CryptoIM CSP divided in two distinct cryptographic libraries. The left side shows only standardized algorithms and comprises a conventional cryptographic library. The right side features only non-standard cryptography and is an alternative library. The following subsections describe these two libraries.

A. *Standard Cryptography*

This subsection details the implementation choices for the standard cryptographic library. The motivations behind this implementation were all characteristics of standardized algorithms: interoperability, documentation, and testability. The programming language chosen for implementation of this cryptographic library was Java. The standard cryptography is a pure-Java library according to JCA.

The block cipher is the AES algorithm, which was implemented along with thee of operation: ECB, and CBC [30], as well as the GCM mode for authenticated encryption [31]. PKCS#5 [5] is the simplest padding mechanism and was chosen for compatibility with other CSPs. As GCM mode uses only AES encryption, the optimization of encryption received more attention than decryption. Implementation aspects of AES and other algorithms can be found on the literature [17][28][43]. This AES implementation was inspired by [33].

The asymmetric algorithm is the RSA Probabilistic Signature Scheme (RSA-PSS) built over the RSA signature algorithm. PSS is supposed to be more secure them ordinary RSA [27][43]. Asymmetric encryption is provided by the RSA Optimal Asymmetric Encryption Padding (RSA-OAEP) [27][43].

Two cryptographically secure hashes were implemented, Standard Hash Algorithm 1 (SHA-1) [26] and Message Digest (MD5). It is well known by now that MD5 is considered broken and is not to be used in serious applications, it is present for ease of implementation. In current version, there is no intended use for these two hashes. Their primary use will be as the underling hash function in

 ISBN: 978-1-61208-376-6

MACs, digital signatures and PGNGs. The MAC chosen were the Hash MAC (HMAC) [29] with SHA-1 as the underling hash function, and the Galois MAC (GMAC) [31], which can be directly derived from GCM mode. Standard Hash Algorithm 2 (SHA-2) family of secure hashes supplies the need for direct use of single hashes.

The need for a Key Agreement (KA) was fulfilled by the implementation of Station-to-Station (STS) protocol (Figure 2), which is based on Authenticated Diffie-Hellman (ADH) [45], and provides mutual key authentication and key confirmation [6][46].

Finally, the mechanism for Password-based Encryption (PBE) is based on the Password-Based Key Derivation Function 2 (PBKDF2) [5], and provides a simple and secure way to store keys in encrypted form. In PBE, a key-encryption-key is derived from a password.

B. Non-standard Cryptography

This subsection details the implementation choices for the alternative cryptographic library. The motivation behind the special attention given to the selection of alternative cryptographic algorithms was the recently revealed weaknesses intentionally included by foreign intelligence agencies in international encryption standards [19]. This fact alone raises doubt on the confidence of all standardized algorithms, which are internationally adopted.

In this context, a need arose to treat what has been called "alternative cryptography" in opposition to standardized cryptographic schemes. The final intent was strengthening the implementation of advanced cryptography and fostering their use. The non-standard cryptography is packaged as dynamic library written in C and accessible to Java programs through a Java Native Interface (JNI) connector, which acts as a bridge to a JCA adapter.

By the time of writing, this alternative library was under the final steps of its construction. It provides advanced mathematical concepts, such as bilinear pairings and elliptic curves, which are not fully standardized by foreign organizations and suffer constant improvements. The most advanced cryptographic protocols currently implemented are based on a reference implementation [8] and listed below.

a) Elliptic Curve Diffie–Hellman (ECDH) [11]. The key agreement protocol ECDH is a variation of the Diffie-Hellman (DH) protocol using elliptic curves as the underlying algebraic structure.
b) Elliptic Curve Digital Signature Algorithm (ECDSA) [25]. This is a DSA-based digital signature using elliptic curves. ECSS [11] is a variant of ECDSA.
c) Sakai-Ohgishi-Kasahara (SOK) [37]. This protocol is a key agreement for Identity-Based Encryption (IBE). It is also called SOKAKA (SOK Authenticated Key Agreement).
d) Boneh-Lynn-Shacham (BLS) [9]. A short digital signature scheme in which given a message m, it is computed S = H (m), where S is a point on an elliptic curve and H() is a secure hash function.
e) Zhang-Safavi-Susilo (ZSS) [14]. Similar to the previous case, it is a more efficient short signature, because it utilizes fixed-point multiplication on an elliptic curve rather arbitrary point.
f) Blake [41]. Cryptographic hash function submitted to the worldwide contest for selecting the new SHA-3 standard. It was ranked among the five finalists of this competition.
g) Elliptic Curve Augmented Encryption Scheme (ECIES) [11]. It is an asymmetric encryption algorithm over elliptic curves. This algorithm is non-deterministic and can be used as a substitute for RSA-OAEP, with the benefit of shorter cryptographic keys.
h) Elliptic Curve Station-to-Station (ECSTS) [11]. Variation of STS protocol using elliptic curves and ECDH as a replacement for ADH.
i) Salsa20 [18]. This is a family of 256-bit stream ciphers submitted to the ECRYPT Project (eSTREAM).
j) Serpent [40]. A 128-bit block cipher designed to be a candidate to the contest that chose the AES. Serpent did not win, but it was the second finalist and enjoys good reputation in the cryptographic community.

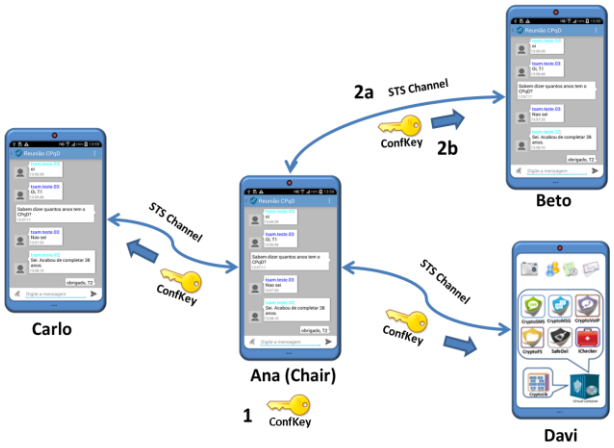


Figure 4. Key agreement for secure conference.

C. Evaluation of standard and non-standard cryptography

A previous work [2] identified lack of alternative cryptography in public libraries, such as non-standard elliptic curves and bilinear pairings. This prototype attempts to fulfill this gap by offering alternatives to possibly compromised standards. Its construction has been discussed in a recent paper [1]. Only key points are recalled here.

Considering security, protection against side-channel attacks was an important issue in the choice of alternative cryptography. Schemes with known issues were avoided, while primitives that were constructed to resist against such attacks were regarded. Also, the library offers alternatives for 256-bit security for both symmetric and asymmetric encryption. For instance, in symmetric encryption, Serpent-256 replaces AES-256. In asymmetric encryption, the same security level is achieved by elliptic curves over 521-bit finite fields, and replaces standard RSA with 15360-bit keys.

Considering performance measurements, experiments [1] have shown that standard cryptography can be competitive to other implementations. Also, in higher security levels, the

 ISBN: 978-1-61208-376-6

performance of non-standard elliptic-curve cryptography is significantly better than standard alternative. In contrast, non-standard pairings-based cryptography has shown relatively low performance. Figure 6 illustrates this behavior for signature operations on a Samsung Galaxy S III (1.4 GHz quad-core Cortex-A9, 1 GB RAM, and 16GB storage). Complete results can be found in [1].

The observed responsiveness shown by the prototype is quite competitive and usage has shown that delay caused by key negotiation is negligible, considering a local wireless network (Wi-Fi) and a household deployment of a XMPP server with few users. However, additional effort needs to be taken in order to optimize the mobile app as well as improve both performance and scalability on server-side application.

V. IMPROVEMENTS UNDER DEVELOPMENT

By the time of writing, two improvements were under construction. The first is a mobile PKI responsible for digital certification, which is fully integrated to the mobile security framework. PKI's Server-side is based upon the EJBCA PKI [13]. Client-side follows recent recommendations for handling certificates on mobile devices [38].

The second is a secure text conference (or group chat) via instant messages. As depicted in Figure 4, the Organizer or Chair of the conference requests the conference creation to the Server, as this is an ordinary XMPP feature. The key agreement for the requested conference proceeds as follows, where $Enc_k(x)$ means encryption of x with key k:

1. Chair (C) creates the key for that conference (c_k);
2. For each guest (g[i]), Chair (C) does:

   a) Opens a STS channel with key k: C ←→ g[i], key k;

   b) Sends c_k on time t to g[i]: C → g[i]: $Enc_k(c_k)$.

The steps above constitute a point-to-point key transport using symmetric encryption, which is provided by the STS protocol. After that, all guests share the same conference key and the conference proceeds as a multicast of all encrypted messages. Figure 5 shows a screenshot for a secure conference, in which users are differentiated by colors. Both the conversation and the interface are in Portuguese.

VI. CONCLUDING REMARKS

This paper discussed design and implementation issues on the construction of a cryptographically secure Instant Message application for Android and the underlying cryptographic library that supports it. This text has shown how cryptographic services can be crafted to adequately fit to a secure IM service in a way that is transparent to the final user, without sacrificing security. A well defined architecture allowed the selection and use of non-standard cryptography.

Future work includes other cryptographically secure services, such as SMS, group chat, and mobile PKI, as well as protections against side-channels and vulnerabilities of insecure programming. Also, performance over 3G networks is being measured and analyzed, for future improvements.

ACKNOWLEDGMENT

The authors acknowledge the financial support given to this work, under the project "Security Technologies for Mobile Environments – TSAM", granted by the Fund for Technological Development of Telecommunications – FUNTTEL – of the Brazilian Ministry of Communications, through Agreement Nr. 01.11. 0028.00 with the Financier of Studies and Projects - FINEP / MCTI.

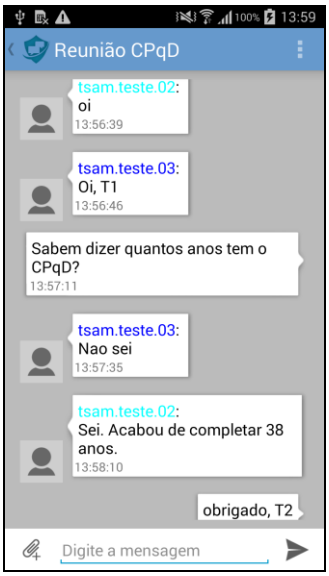


Figure 5. Screenshot of a secure text conference (group chat).

REFERENCES

[1] A. M. Braga and E. M. Morais, "Implementation Issues in the Construction of Standard and Non-Standard Cryptography on Android Devices," The Eighth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2014), in press.

[2] A. Braga and E. Nascimento, Portability evaluation of cryptographic libraries on android smartphones. In Proceedings of the 4th international conference on Cyberspace Safety and Security (CSS'12), Yang Xiang, Javier Lopez, C.-C. Jay Kuo, and Wanlei Zhou (Eds.). Springer-Verlag, Berlin, Heidelberg, 2012, pp. 459-469.

[3] A. M. Braga, "Integrated Technologies for Communication Security on Mobile Devices", The Third International Conference on Mobile Services, Resources, and Users (Mobility) , 2013, pp. 47–51.

[4] A. M. Braga, E. N. Nascimento, and L. R. Palma, "Presenting the Brazilian Project TSAM – Security Technologies for Mobile Environments", Proceeding of the 4th International Conference in Security and Privacy in Mobile Information and Communication Systems (MobiSec 2012). LNICST, vol. 107, 2012, pp. 53-54.

[5] B. Kaliski, "PKCS #5: Password-Based Cryptography Specification", Version 2.0, RFC 2898. Retrieved [July 2014] from tools.ietf.org/html/rfc2898.

[6] B. O'Higgins, W. Diffie, L. Strawczynski, and R do Hoog, "Encryption and ISDN - A Natural Fit", International Switching Symposium (ISS87), 1987.

[7] B. Xuefu and Y. Ming, "Design and Implementation of Web Instant Message System Based on XMPP", Proc. 3rd International Conference on Software Engineering and Service Science (ICSESS), Jun. 2012, pp. 83-88.

[8] D. Aranha and C. Gouvêa, "RELIC Toolkit. Retrieved [July 2014] from code.google.com/p/relic-toolkit.

[9] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing", J. Cryptology, 17(4), Sept. 2004, pp. 297–319.

[10] D. Bornstain, Dalvik VM Internals. Retrieved [July 2014] from sites.google.com/ site/io/dalvik-vm-internals.

[11] D. Hankerson, A. J. Menezes, and S. Vanstone. Guide to Elliptic Curve Cryptography, Springer-Verlag, New York, Inc., Secaucus, NJ, USA, 2003.

[12] D. T. Massandy and I. R. Munir, "Secured Video Streaming Development on Smartphones with Android Platform", Proc. 7th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Oct. 2012, pp. 339-344.

[13] EJBCA PKI CA. Retrieved [July 2014] from http://www.ejbca.org.

[14] F. Zhang, R. Safavi-Naini, and W. Susilo, "An Efficient Signature Scheme from Bilinear Pairings and Its Applications", in F. Bao, R. H. Deng and J. Zhou, ed., 'Public Key Cryptography', 2004, pp. 277-290.

[15] H. Krawczyk, "SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie-Hellman and its use in the IKE protocols." Advances in Cryptology-CRYPTO 2003, Springer Berlin Heidelberg, 2003, pp. 400-425.

[16] How to Implement a Provider in the Java Cryptography Architecture. Retrieved [July 2014] from docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/HowToImplAProvider.html.

[17] J. Bos, D. Osvik, and D. Stefan, "Fast Implementations of AES on Various Platforms", 2009. Retrieved [July 2014] from eprint.iacr.org/2009/501.pdf.

[18] J. D. Bernstein, The Salsa20 family of stream ciphers. Retrieved [July 2014] from cr.yp.to/papers.html#salsafamily.

[19] J. Menn, Experts report potential software "back doors" in U.S. standards. Retrived [July 2014] from http://www.reuters.com/article/2014/07/15/usa-nsa-software-idUSL2N0PP2BM20140715?irpc=932.

[20] Java Cryptography Architecture (JCA) Reference Guide. Retrieved [July 2014] from docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/CryptoSpec.html.

[21] Java Cryptography Architecture Oracle Providers Documentation for Java Platform Standard Edition 7. Retrieved [July 2014] from docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html.

[22] Java Cryptography Architecture Standard Algorithm Name Documentation for Java Platform Standard Edition 7. Retrieved [July 2014] from docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html.

[23] Java Cryptography Extension Unlimited Strength Jurisdiction Policy Files 7 Download. Retrieved [July 2014] from www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html.

[24] M. Egele, D. Brumley, Y. Fratantonio, and C. Kruegel, "An empirical study of cryptographic misuse in android applications," Proceedings of the 2013 ACM SIGSAC conference on Computer and Communications Security (CCS '13), 2013, pp. 73–84.

[25] NIST FIPS PUB 186-2. Digital Signature Standard (DSS). Retrieved [July 2014] from csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf.

[26] NIST FIPS-PUB-180-4. Secure Hash Standard (SHS). March 2012. Retrieved [July 2014] from csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf.

[27] NIST FIPS-PUB-186. Digital Signature Standard (DSS). Retrieved [July 2014] from csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf.

[28] NIST FIPS-PUB-197. Announcing the ADVANCED ENCRYPTION STANDARD (AES). Federal Information Processing Standards Publication 197 November 26, 2001.

[29] NIST FIPS-PUB-198. The Keyed-Hash Message Authentication Code (HMAC). Retrieved [July 2014] from csrc.nist.gov/publications/fips/fips198/fips-198a.pdf.

[30] NIST SP 800-38A. Recommendation for Block Cipher Modes of Operation. 2001. Retrieved [July 2014] from csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf.

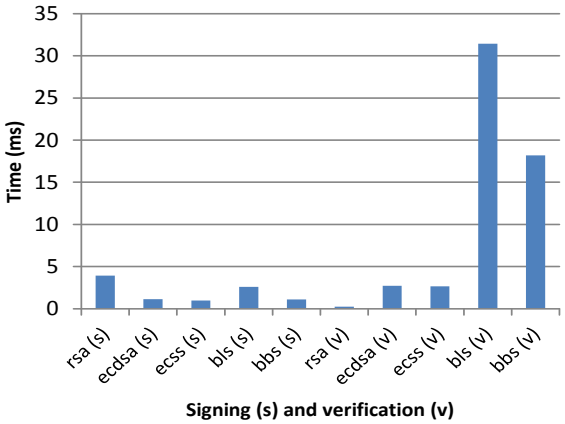


Figure 6. Time measurements for signature algorithms.

[31] NIST SP 800-38D. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. 2007. csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf.

[32] Off-the-Record Messaging webpage. Retrieved [July 2014] from otr.cypherpunks.ca.

[33] P. Barreto, AES Public Domain Implementation in Java. Retrieved [July 2014] from www.larc.usp.br/~pbarreto/JAES.zip.

[34] P. Gutmann, "Lessons Learned in Implementing and Deploying Crypto Software," Usenix Security Symposium, 2002.

[35] P. Saint-Andre, K. Smith, and R. Tronçon, "XMPP: The Definitive Guide - Building Real-Time Applications with Jabber Technologies", O'reilly, 2009.

[36] Piercing Through WhatsApp's Encryption. Retrieved [July 2014] from blog.thijsalkema.de/blog/2013/10/08/piercing-through-whatsapp-s-encryption.

[37] R. Sakai, K. Ohgishi, and M. Kasahara. "Cryptosystems based on pairing". The 2000 Symposium on Cryptography and Information Security (SCIS 2000), Okinawa, Japan, January 2000, pp. 26–28.

[38] S. Fahl, M. Harbach, and H. Perl, "Rethinking SSL development in an appified world," Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13 (2013), 2013, pp. 49–60.

[39] S. Schrittwieser et al., "Guess Who's Texting You? Evaluating the Security of Smartphone Messaging Applications". Proc. 19th Network & Distributed System Security Symposium, Feb. 2012.

[40] SERPENT webpage, "SERPENT A Candidate Block Cipher for the Advanced Encryption Standard". Retrieved [July 2014] from www.cl.cam.ac.uk/~rja14/serpent.html.

[41] SHA-3 proposal BLAKE webpage. Retrieved [July 2014] from https://131002.net/blake.

[42] SpongyCastle webpage, Spongy Castle: Repackage of Bouncy Castle for Android, Bouncy Castle Project (2012), Retrieved [July 2014] from rtyley.github.com/spongycastle/

[43] T. St. Denis. "Cryptography for Developers", Syngress, 2007.

[44] The Legion of the Bouncy Castle webpage. Legion of the Bouncy Castle Java cryptography APIs. Retrieved [July 2014] from www.bouncycastle.org/java.html.

[45] W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Transact. on Inform. Theory, vol. 22, no. 6, Nov. 1976, pp. 644-654.

[46] W. Diffie, P. C. van Oorschot, and M. J. Wiener, "Authentication and Authenticated Key Exchanges", Designs, Codes and Cryptography (Kluwer Academic Publishers) 2 (2), 1992, pp. 107–125.

[47] W. Mao, "Modern cryptography: theory and practice", Prentice Hall PTR, 2004.

Resisting Flooding Attacks on AODV

Mohamed A. Abdelshafy and Peter J. B. King
School of Mathematical & Computer Sciences
Heriot-Watt University, Edinburgh, UK
{ma814, P.J.B.King}@hw.ac.uk

***Abstract*—AODV is a reactive MANET routing protocol that is vulnerable to a dramatic collapse of throughput when malicious intruders flood the network with bogus route requests. We introduce a simple mechanism to resist such attacks that can be incorporated into any reactive routing protocol. It does not require expensive cryptography or authentication mechanisms, but relies on locally applied timers and thresholds to classify nodes as malicious. No modifications to the packet formats are needed, so the overhead is a small amount of calculation at nodes, and no extra communication. Using NS2 simulation, we compare the performance of networks using AODV under flooding attacks with and without our mechanism, showing that it significantly reduces the effect of a flooding attack.**

Keywords–MANET, Routing, AODV, Security, Attack, Flooding

I. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a decentralized infrastructureless network in which nodes cooperate to forward data from a source to a destination. Each node in a MANET acts both as a router and as a host. Several routing protocols have been designed for MANETs [1] to optimize network routing performance. The major issues involved in designing a routing protocol for MANET are nodes mobility, bandwidth constrained and error prone wireless channel, resource constrained nodes, and dynamic changing of the network topology [2].

MANET routing protocols can be classified as proactive or reactive routing protocols. In proactive (table-driven) routing protocols, each node maintains one or more tables containing routing information to every other node in the network. While in reactive (on-demand) routing protocols, routes are created whenever a source requires to send data to a destination node which means that these protocols are initiated by a source on-demand. In this paper, we focus on the AODV protocol [3] which is one of the extensively studied reactive protocols, considered by the IETF for standardization.

AODV [3] is a reactive routing protocol. It uses destination sequence numbers to ensure the freshness of routes and guarantee loop freedom. To find a path to a destination, a node broadcasts a route request (RREQ) packet to its neighbors using a new sequence number. Each node that receives the broadcast sets up a reverse route towards the originator of the RREQ unless it has a fresher one. When the intended destination or an intermediate node that has a fresh route to the destination receives the RREQ, it unicasts a reply by sending a route reply (RREP) packet along the reverse path established at intermediate nodes during the route discovery process. Then the source node starts sending data packets to the destination node through the neighboring node that first responded with an RREP. When an intermediate node along the route moves, its upstream neighbor will notice route breakage due to the movement and propagate a route error (RERR) packet to each of its active upstream neighbors. Routing information is stored only in the source node, the destination node, and the intermediate nodes along the active route which deal with data transmission. This scenario decreases the memory overhead, minimizes the use of network resources, and runs well in high mobility situation.

MANET inherits security threats that are faced in wired as well as wireless networks and also introduces security attacks unique to itself [2] due its characteristics. The limitations associated with battery powered MANET nodes mean that computationally expensive cryptographic techniques such as public key algorithms are undesirable.

MANET routing protocols are designed based on the assumption that all nodes cooperate without maliciously disrupting the operation of the protocol. However, the existence of malicious nodes cannot be disregarded in any system, especially in MANETs because of the wireless nature of the network. A malicious node can attack the network layer in MANET either by not forwarding packets or by changing some parameters of routing messages such as sequence number and IP addresses, sending fake messages several times and sending fake routing information to cause congestion and so disrupt routing operations. Node mobility introduces also the difficulty of distinguishing between stale routes and fake routes. Attacks on MANETs come in a number of classes [4] and a number of defences to these attacks have been proposed and evaluated by simulation [4]–[6]. Attacks against MANET are classified based on modification, impersonation or fabrication of the routing messages. While there are large number of existing attacks, our paper is focused on flooding attack which has a dramatic impact on AODV [2] [4].

In AODV under flooding attack [7], a malicious node floods the network with a large number of RREQs to non-existent destinations in the network. Since the destination does not exist in the network, a RREP packet cannot be generated by any node in the network. When a large number of fake RREQ packets are being injected into the network by malicious nodes, significant proportions of the network capacity are consumed by the RREQ packets, depleting the bandwidth available for data. In addition, routing tables accumulate reverse routes to the source of the fake packets, often leading to table overflow and the inability to record new valid routes. This is a type of denial of service attack.

Security mechanisms are added to existing routing protocols to resist attacks. Cryptographic techniques are used to ensure the authenticity and integrity of routing messages [8]. A major concern is the trade off between security and performance, given the limited resources available at many MANET nodes. Both symmetric and asymmetric cryptography have been used as well as hash chaining. Examples of these

 ISBN: 978-1-61208-376-6

security enhanced protocols are Authenticated Routing for Ad-hoc Networks (ARAN) [9], Secure Link State Routing Protocol (SLSP) [10], and Secure Ad-hoc On-demand Distance Vector routing (SAODV) [11]. In addition to the power and computation cost of using cryptographic techniques, the performance of secured mechanism such as SAODV is worse than AODV [4] in the presence of flooding attack because of the malicious nodes impersonating non-existent nodes which cannot be discovered by other non-malicious nodes. Thus, securing the routing messages cannot guarantee the detection of the flooding malicious nodes.

We introduce a new Anti-Flooding mechanism that can be used for all on-demand routing protocols. Each node in this mechanism is responsible for monitoring the behaviour of its neighbors to detect malicious nodes and exclude them. We integrate our proposed mechanism into AODV and SAODV as examples of on-demand routing protocols. This paper demonstrates a significant improvement in performance when using our mechanism. The results reported here related to AODV, but we have also measured SAODV with this mechanism and the improvement in performance is significantly higher than AODV.

The rest of the paper is organized as follows. Section II presents the related work. In Section III, our proposed mechanism to detect the flooding attack is introduced. In Section IV, the simulation approach and parameters is presented. In Section V, simulation results are given. In Section VI, conclusions are drawn.

II. Related Work

Although significant algorithms have been introduced to secure MANET, most of these algorithms cannot resist a flooding attack. A malicious node initiating a flooding attack generates a large number of RREQs to non-existant nodes. These RREQ flood out through the MANET and because the destination does not exist, are propagated by all nodes. A node has no way of detecting whether the neighbor that sent the RREQ is malicious or not. All suggested solutions to the flooding attack attempt to classify neighbors as normal or malicious nodes and then suppress malicious ones.

Flooding Attack Prevention (FAP) [12] is the first solution to resist against flooding attack. The algorithm defined a neighbor suppression method which prioritizes the node based on the number of RREQ received. A node gets higher priority if it sends fewer RREQ packets. When a malicious node broadcasts large number of RREQ packets, the immediate neighbors of the malicious node observe a high rate of RREQ and then they lower the corresponding priority according to the rate of incoming queries. Forwarding received RREQ depends on the priority value of the sending neighbor. The disadvantage of this algorithm is that it still disseminates flooding packets albeit at a reduced rate.

Threshold prevention [13] is introduced to modify FAP by defining a fixed RREQ threshold. The algorithm assumes that if the number of RREQ packets received from a neighbor exceeds the threshold value, this neighbor is a malicious node and discards all future packets from this malicious node. The algorithm becomes useless if a malicious node knows the threshold value then it can bypass the mechanism. Another disadvantage of this algorithm is that it treats a high mobility normal node as if it is a malicious node.

A distributed approach to resist the flooding attack is introduced in [14]. The algorithm defines two threshold values; RATE_LIMIT and BLACKLIST_LIMIT. A RREQ from a neighbor is processed only if the number of previously received RREQ from this neighbor is less than RATE_LIMIT. On the other hand, if the number of previously received RREQ from this neighbor is greater than BLACKLIST_LIMIT, the RREQ is discarded and this neighbor is blacklisted. If the number of previously received RREQ from this neighbor is greater than RREQ_LIMIT and less than BLACKLIST_LIMIT, the RREQ is queued for processing after a delay expires. A disadvantage of this approach is the ability of the attacker to subvert the algorithm by disseminating thresholds levels and the possibility of permanently suspending a blacklisted neighbor that is not malicious.

The algorithm introduced in [15] tried to find a solution to the flooding attack from the communication point of view. The algorithm defines three threshold values; transmission threshold, blacklist threshold and white listing threshold. A RREQ from a neighbor is processed only if received RREQ rate from this neighbor is less than the transmission threshold; otherwise the node will discards the RREQ. If the received RREQ rate from this neighbor is greater than the blacklist threshold, the RREQ is discarded and this neighbor is blacklisted. This algorithm avoids permanently suspending of a blacklisted neighbor by introducing a white listing threshold. A blacklisted neighbor can be returned to normal status if it behaves correctly for a whitelisting time interval.

The algorithm introduced in [16] extends DSR protocol based on the trust function to mitigate the effects of flooding attack. This algorithm classifies a node neighbors based on a trust value to three categories; friend, acquaintance and stranger. Friend is a trusted node and stranger is a non-trusted node while an acquaintance has the trust value that is greater than a stranger and less than a friend. The algorithm defines a threshold value to each neighbor type. A node decision will be taken based on the neighbor type that sends the RREQ and threshold value of this neighbor type. As a general rule, if a node receives a RREQ from a neighbor, it first checks its relationship class and based on this it checks if this neighbor runs over the relationship class threshold value or not. The node processes the RREQ if this neighbor still running under the relationship class threshold otherwise it discards the RREQ and blacklists this neighbor. The disadvantage of this algorithm is that it cannot support high node mobility. [17] introduces a modification to this algorithm to extend the algorithm for high node mobility. A significant disadvantage of this approach is that it depends on a modification of DSR and cannot be adapted to other MANET protocols.

III. AF-AODV Protocol

AF-AODV is designed to mitigate the effect of the flooding attack on the performance of AODV protocol. The mechanism does not use cryptographic techniques which conserves the power and computation resources. Each node in the network has to monitor the performance of its neighbors to detect if they are trying to flood the network or not. Malicious nodes will be detected reliably within a very few minutes. The only way for a malicious node to subvert the mechanism is to transmit fake RREQ packets at such a low rate that they do not impact the network performance significantly.

 ISBN: 978-1-61208-376-6

The idea is to record for each neighbor the rate at which it transmits RREQs. A node pursuing a flooding attack will be generating a high number of RREQs. If the rate exceeds a threshold, then the neighbor is added to a black list of potential malicious nodes. Once on the black list, RREQs from the black listed node are not forwarded, but they are still recorded. A node can be removed if its rate of RREQ generation reduces below the threshold. If the rate continues high, the offending node is queried - only a non-malicious node will respond. After two queries, the neighbor will be suspended for a period, and if its rate is still high after the period has elapsed it will be declared as malicious. A node implementing the Anti-Flood mechanism behaves as follows:

- Every TRAFFIC_TIME, the number of RREQs received from each neighbor since the last classification update is examined.
- If the number of RREQs received from a neighbor exceeds the threshold RREQ_THRESHOLD, that neighbour has its black_list value set to 1. If multiple neighbours exceed the threshold, the neighbor which has transmitted the largest number of RREQs has its black_list value set to 1. Other neighbors that exceeded the threshold are suspended. RREQs from suspended nodes are ignored and not forwarded. Suspension of neighbors except the one with the largest RREQ count allows the mechanism to avoid double counting of RREQs and concentrate on classification of the worst offender. Choice of the RREQ_THRESHOLD is made by running AODV on a large number of scenarios and observing the largest number of RREQs that can be received in TRAFFIC_TIME.
- RREQ packets are processed normally when received from neighbors with a black_list value of 0. If a RREQ is received from a neighbor with a black_list value of 1, then the node examines how many RREQs have been received in an interval of RREQ_TIME_1. If that is less than RREQ_COUNT_1, the black_list value for that neighbor is reset to 0. If the number exceeds RREQ_COUNT_1, the node tests the authenticity of the neighbor by constructing a fake RREP packet to the RREQ and replying with that RREP. If the neighbor is malicious, this will not result in any data flowing. If it is not malicious, data will flow to the fake RREP originator, which can respond with a RERR so that a new route can be found. If no data flows within RREP_WAIT_TIME, the neighbor's black_list value is set to 2.
- If a RREQ is received from a neighbor with a black_list value of 2, it re-examines the rate of RREQ received from that node. If the number of RREQ received from this neighbor is less than RREQ_COUNT_1 in a duration less than or equals RREQ_TIME_1, it decrements the black_list value to 1. Otherwise the node again sends a fake RREP to the RREQ sender to test its authenticity. If the RREP_WAIT_TIME expires without receiving the data, the node assigns 3 to black_list value of this neighbor and suspends this neighbor for a long period equals to the next TRAFFIC_TIME + EXCLUDE_TIME. This long suspension ensures that if the behaviour of this neighbor has been affected by a malicious node, then that malicious node will have been identified and isolated during this suspension.
- After the long-time suspension has expired, the node restarts the previous process; it counts again the number of received RREQ from this neighbor and if the number is less than the threshold RREQ_THRESHOLD, it decrements the black_list value to 2. Otherwise it will increment the black_list value to 4.
- If a RREQ is received from a neighbor with a black_list value equals 4, it monitors the rate of RREQ received from this neighbor. If the number of RREQ received from this neighbor is less than RREQ_COUNT_1 in a duration less than or equals RREQ_TIME_1, it decrements the black_list value to 3. Otherwise the node sends a fake RREP to the RREQ sender to test its authenticity for the final time. If the RREP_WAIT_TIME expires without receiving the data, the node assigns 5 to black_list value of this neighbor meaning that this neighbor is a malicious node and deletes this neighbor from neighbor list. All received RREQ from a neighbor that has black_list value equals 5 will be dropped without processing as a result of detecting a malicious node.

TABLE I. AF-AODV PARAMETERS

RREQ_THRESHOLD	10
RREQ_COUNT_1	7
RREQ_COUNT_2	3
RREQ_TIME_1	5
RREQ_TIME_2	2
RREP_WAIT_TIME	1 s
TRAFFIC_TIME	10 s
EXCLUDE_TIME	60 s

Table 1 shows the values of parameters that were used in our simulations.

IV. Simulation Approach

NS-2 simulator [18] is used to simulate flooding attack. The simulation is used to analyse the performance of AODV and our new AF-AODV routing protocols under these attacks. The parameters used are shown in Table 2. Node mobility was modelled with the random waypoint method. Our simulation results are obtained from 3 different movement scenarios, 3 different traffic scenarios and 3 different node-type (malicious or non-malicious) scenarios which means that each metric value is the mean of the 27 runs. The node-type scenario is created randomly. In all cases, the 90% confidence interval was small compared with the values being reported. In this paper, we focused on their impact of the flooding attack on the TCP traffic only. We examined our proposed mechanism for different number of nodes (25, 50, 75 and 100) and different node speeds (0, 10, 20 and 30 m/s). Node mobility had no significant effect of performance in the presence of malicious nodes, so we report here only the case of static networks. Similarly, only the case of 100 node networks is reported, corresponding to a high density of nodes. This gives malicious nodes a high number of neighbors. We choose a large simulation time to be sure that all malicious nodes have

 ISBN: 978-1-61208-376-6

TABLE II. SIMULATION PARAMETERS

Simulation Time	600 s
Simulation Area	500 m x 500 m
Number of Nodes	25, 50, 75, 100
Number of Malicious Nodes	0 - 10
Node Speed	0, 10, 20, 30 m/s
Pause Time	10 s
Traffic Type	TCP
Flooding Rate	2 Packets/s

been detected specially for scenarios with a large number of malicious nodes.

Packet Delivery Ratio (PDR): The ratio of packets that are successfully delivered to a destination compared to the number of packets that have been sent out by the sender.

Throughput: The number of data bits delivered to the application layer of destination node in unit time measured in bps.

End-to-End Delay (EED): The average time taken for a packet to be transmitted across the network from source to destination.

Routing Overhead: The size of routing packets measured in Kbytes for route discovery and route maintenance needed to deliver the data packets from sources to destinations.

Normalized Routing Load (NRL): The total number of routing packets transmitted divided by the number of received data packets.

Route Discovery Latency (RDL): The average delay between the sending RREQ from a source and receiving the first corresponding RREP.

Sent Data Packets: The total number of packets sent by all source nodes during the simulation time.

V. SIMULATION RESULTS

The effect of flooding attack on the packet delivery ratio is shown in Figure 1. While the flooding attack has severe impact on the PDR of AODV specially for large number of malicious nodes, AF-AODV has not significantly change for low number of malicious nodes and has negligible decreasing for high number of malicious nodes. AF-AODV enhances PDR over AODV by approximately 5%.

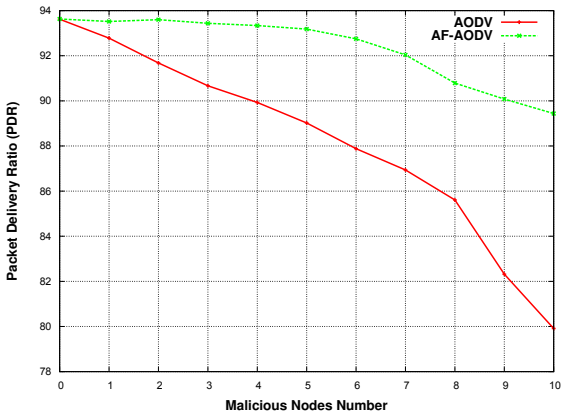


Figure 1. Packet Delivery Ratio

The enhancement of PDR becomes more remarkable if we integrate it to the number of packets that can be sent which is shown in Figure 2. The figure shows that the total number of packets that can be sent is dramatically decreasing as the number of malicious nodes increases to the extent that when the number of malicious nodes becomes 10, it can only send 15% of the packets when there is no malicious nodes. Our proposed mechanism AF-AODV introduces an enhancement of about 35% over AODV. In addition to this advantage, AF-AODV has not significantly change for low number of malicious nodes. By combining Figure 1 and Figure 2, we can notice a large enhancement in number of packets that is received by destination specially for large number of malicious nodes. As an example, if the number of malicious nodes is 10, the number of received packets by destinations in AODV is approximately 5600 packets while it is about 20250 packets in AF-AODV which means that the number of received packets is improved by approximately 360%.

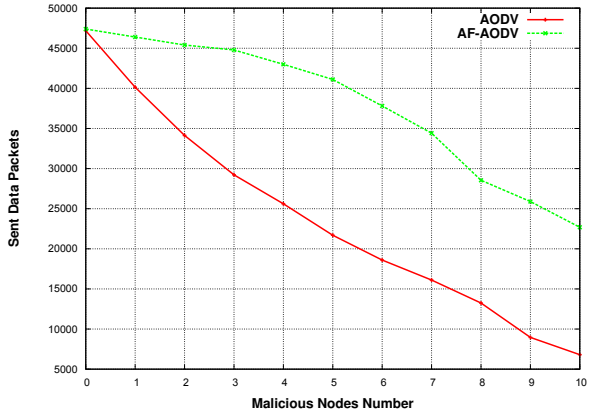


Figure 2. Send Data Packets

Figure 3 shows the effect of flooding attack on the network throughput. Throughput of AF-AODV is better than AODV by approximately 20% for each malicious node. While the throughput of AODV dramatically decreases as the number of malicious nodes increases, AF-AODV slightly decreases for the low number of malicious nodes.

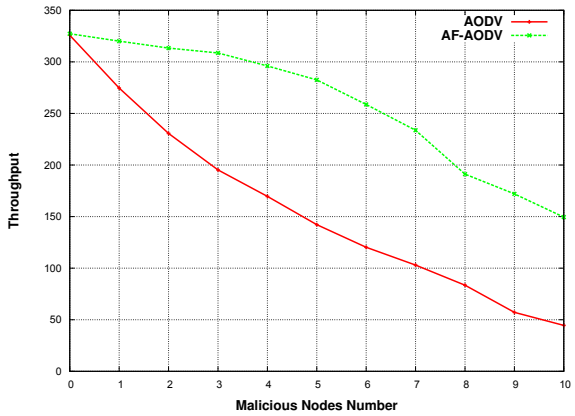


Figure 3. Network Throughput

 ISBN: 978-1-61208-376-6

The effect of flooding attack on the end-end-delay is shown in Figure 4. The result shows that there is no significant change of the delay of AF-AODV while the delay increases as the number of malicious nodes increases.

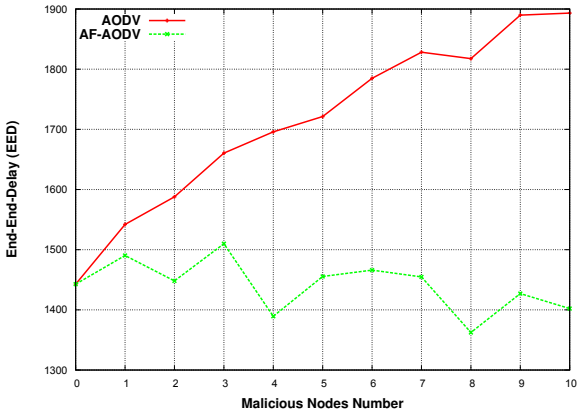


Figure 4. End-to-End Delay

Figure 5 shows the effect of flooding attack on the normalized routing load. The result shows that while the normalized routing load of AODV increases as the number of malicious nodes increases specially for large number of malicious nodes, it has not significant change for AF-AODV.

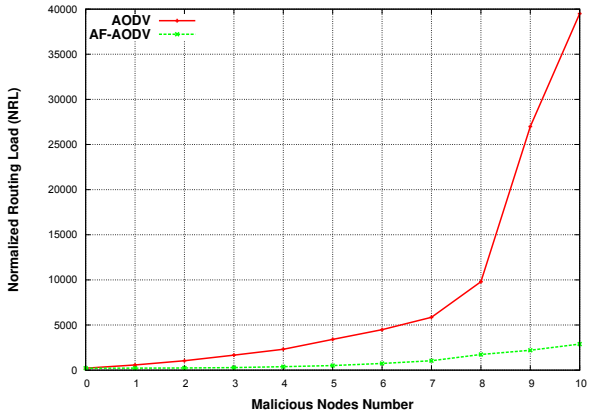


Figure 5. Normalized Routing Load

Figure 6 shows the effect of flooding attack on the routing overhead. The result shows that the routing overhead of AF-AODV has not significantly change for the low number of malicious nodes and slightly increases as the number of malicious nodes increases. On the other hand, it increases dramatically as the number of malicious nodes increases for AODV.

Figure 7 shows the effect of flooding attack on the routing discovery latency. The result shows that the routing discovery latency of AF-AODV is nearly constant regardless the number of malicious nodes. On the other hand, it increases dramatically as the number of malicious nodes increases for AODV.

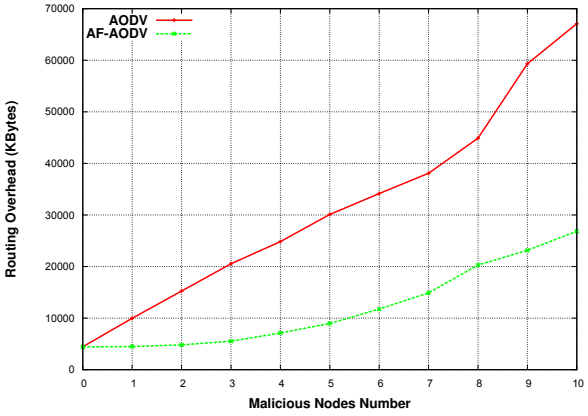


Figure 6. Routing Overhead

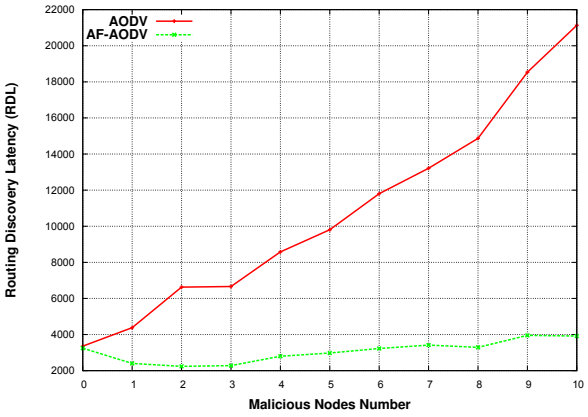


Figure 7. Route Discovery Latency

The number of packets that will be dropped as a result of detecting the presence of malicious nodes is shown in Figure 8. The result shows that while AF-AODV dropped packets increases as the number of malicious nodes increasing, AODV cannot detect the presence of malicious nodes and hence the protocol does not drop packets.

Our simulation shows that regardless the number of nodes and the number of malicious nodes in the network, the malicious node neighbor can detect its presence in a few minutes and the time to detect the last malicious node is increases for sure as the number of malicious nodes increasing. Figure 9 shows the time required by non-malicious nodes to detect the last malicious node in the network.

VI. CONCLUSION

In this paper, we introduced a new anti-flooding mechanism that can be integrated into any reactive routing protocol in MANET. The proposed mechanism did not use cryptographic techniques which conserves the power and computation resources. Furthermore, the mechanism did not require any additional packets and hence does not incur any additional overhead. As an example, we integrated our anti-flooding mechanism with AODV to study the performance of the

 ISBN: 978-1-61208-376-6

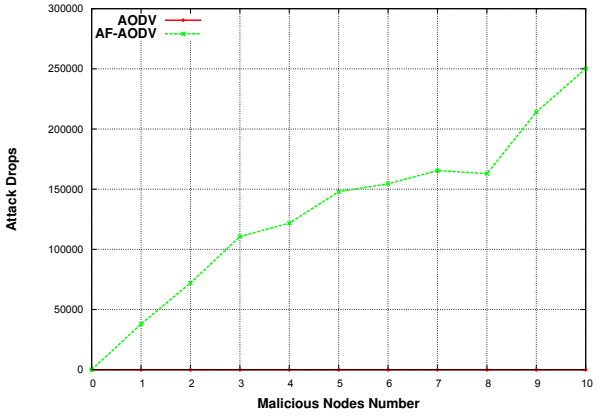


Figure 8. Attack Dropped Packets

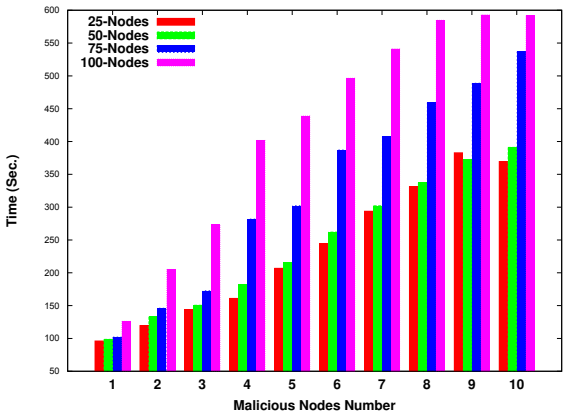


Figure 9. Time Required to Detect the Last Malicious Node by its Neighbors

network under the presence and absence of the mechanism. We validated the performance analysis of our mechanism through NS2 simulations. Simulation results showed that AF-AODV has a remarkable improvement of the network performance in all network metrics than AODV. The proposed mechanism succeeded to detect malicious nodes that try to flood the network within a few minutes regardless the number of malicious nodes and the time they are participating in the network. Future work includes extending this idea to other reactive protocols, and confirming its general applicability.

REFERENCES

[1] A. Boukerche and et al., "Routing protocols in ad hoc networks: a survey," Computer Networks, vol. 55, no. 13, September 2011, pp. 3032–3080.

[2] M. A. Abdelshafy and P. J. King, "Analysis of security attacks on AODV routing," in 8th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, Dec 2013, pp. 290–295.

[3] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1997, pp. 90–100.

[4] M. A. Abdelshafy and P. J. King, "AODV & SAODV under attack:performance comparison," in ADHOC-NOW 2014, LNCS 8487, Benidorm, Spain, Jun 2014, pp. 318–331.

[5] M. Patel and S. Sharma, "Detection of malicious attack in manet a behavioral approach," in IEEE 3rd International on Advance Computing Conference (IACC), 2013, pp. 388–393.

[6] G. Usha and S. Bose, "Impact of gray hole attack on adhoc networks," in International Conference on Information Communication and Embedded Systems (ICICES), 2013, pp. 404–409.

[7] Y. Guo and S. Perreau, "Detect DDoS flooding attacks in mobile ad hoc networks," International Journal of Security and Networks, vol. 5, no. 4, Dec. 2010, pp. 259–269.

[8] P. Joshi, "Security issues in routing protocols in MANETs at network layer," Procedia CS, vol. 3, 2011, pp. 954–960.

[9] K. Sanzgiri and et al., "Authenticated routing for ad hoc networks," IEEE Journal On Selected Areas In Communications, vol. 23, 2005, pp. 598–610.

[10] P. Papadimitratos and Z. J. Haas, "Secure link state routing for mobile ad hoc networks," in Symposium on Applications and the Internet Workshops. IEEE Computer Society, 2003, pp. 379–383.

[11] M. G. Zapata, "Secure ad hoc on-demand distance vector routing," SIGMOBILE Mob. Comput. Commun. Rev., vol. 6, no. 3, jun 2002, pp. 106–107.

[12] P. Yi, Z. Dai, Y.-P. Zhong, and S. Zhang, "Resisting flooding attacks in ad hoc networks," in International Conference on Information Technology: Coding and Computing (ITCC), vol. 2, April 2005, pp. 657–662.

[13] B.-C. Peng and C.-K. Liang, "Prevention techniques for flooding attacks in ad hoc networks," in 3rd Workshop on Grid Technologies and Applications (WoGTA 06), Hsinchu, Taiwan, December 2006, pp. 657–662 Vol. 2.

[14] J.-H. Song, F. Hong, and Y. Zhang, "Effective filtering scheme against rreq flooding attack in mobile ad hoc networks," in Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT). Washington, DC, USA: IEEE Computer Society, 2006, pp. 497–502.

[15] V. Balakrishnan, V. Varadharajan, U. Tupakula, and M. Moe, "Mitigating flooding attacks in mobile ad-hoc networks supporting anonymous communications," in 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless), Aug 2007, pp. 29–34.

[16] R. Venkataraman, M. Pushpalatha, R. Khemka, and T. R. Rao, "Prevention of flooding attacks in mobile ad hoc networks," in Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3). New York, NY, USA: ACM, 2009, pp. 525–529.

[17] U. D. Khartad and R. K. Krishna, "Route request flooding attack using trust based security scheme in MANET," International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN), vol. 1, no. 4, 2012, pp. 27–33.

[18] The Network Simulator NS-2, http://www.isi.edu/nsnam/ns/ [retrieved: September, 2014].

 ISBN: 978-1-61208-376-6

The Policy-Based AS_PATH Verification to Monitor AS Path Hijacking

Je-Kuk Yun, Beomseok Hong
Information Technology
Towson University
Towson, U.S.A.
jyun4, bhong1@students.towson.edu

Yanggon Kim
Information Technology
Towson University
Towson, U.S.A.
ykim@towson.edu

***Abstract*— As the number of IP prefix hijacking incidents has increased, many solutions are proposed to prevent IP prefix hijacking, such as RPKI, BGPmon, Argus, and PHAS. Except RPKI, all of the solutions proposed so far can protect ASes only through the origin validation. However, the origin validation cannot detect specified attacks that alter the AS_PATH attribute, such as AS Insertion attack and Invalid AS_PATH Data Insertion attack. In order to solve these problems, SIDR proposed the RPKI using BGPSEC, but BGPSEC is currently a work in progress. So, we propose Secure AS_PATH BGP (SAPBGP) in which we monitor the AS_PATH attribute in update messages whether each AS in the AS_PATH attribute are connected to each other based on our policy database collected from RIPE NCC repository. Our analysis shows 4.57% of the AS_PATH attribute is invalid and 95.43% of the AS_PATH attribute is valid from the fifteenth of April in 2014 to the eighth of June in 2014. In addition, the performance test verifies that the SAPBGP can process all of the live BGP messages coming from BGPmon in real time.**

Keywords- border gateway protocol; interdomain routing; network security; networks; AS path hijacking.

I. INTRODUCTION

The Border Gateway Protocol (BGP) is the de-facto protocol to enable large IP networks to form a single Internet [1]. The main objective of BGP is to exchange Network Layer Reachability Information (NLRI) among Autonomous Systems (ASes) so that BGP routers can transfer their traffic to the destination.

However, BGP itself does not have mechanisms to verify if a route is valid because BGP speaker completely trusts other BGP speakers. This lack of consideration of BGP vulnerabilities often causes severe failures of Internet service provision [2]. The most well-known threat of the failures is the YouTube hijacking by Pakistan Telecom (AS17557) on the 24th of February in 2008 [3]. In response to the government's order to block YouTube access within their ASes, Pakistan Telecom announced a more specific prefix than YouTube prefix. Then, one of Pakistan Telecom's upstream providers, PCCW Global (AS3491), forwarded the announcement to other neighbors. As a result of this, YouTube traffic from all over the world was misled to Pakistan Telecom (AS17557) for two hours. In order to solve these problems, many studies were conducted, such as Resource Public Key Infrastructure (RPKI) [4], BGPmon [5], Argus [6], and a Prefix Hijack Alert System (PHAS) [7].

While there are many studies to IP prefix hijacking, few studies have been researched about AS path hijacking. There was some misdirected network traffic suspected of the man-in-the-middle (MITM) attack in 2013 observed by Renesys. In February 2013, global traffic was redirected to Belarusian ISP GlobalOneBel before its intended destination and it occurred on an almost daily basis. Major financial institutions, governments, and network service providers were affected by this traffic diversion in several countries including the U.S. From the thirty first of July to the nineteenth of August, Icelandic provider Opin Kerfi announced origination routes for 597 IP networks owned by a large VoIP provider in the U.S through Siminn, which is one of the two ISPs that Opin Kerfi has. However, this announcement was never propagated through Fjarskipti which is the other one of the two ISPs. As a result, network traffic was sent to Siminn in London and redirected back to its intended destination. Several different countries in some Icelandic autonomous systems and belonging to the Siminn were affected. However, Opin Kerfi said that the problem was the result of a bug in software and had been resolved [8].

In order to protect the AS path hijacking, the AS_PATH attribute should not be manipulated. However, the BGP itself cannot check whether the AS_PATH attribute has been changed or not. If a routing hijacker manipulates the AS_PATH attribute in a BGP message that is sent by another router and forwards the manipulated BGP message to other neighbors, the neighbors who receive the manipulated BGP message can be a victim of AS path hijacking. Only Secure Inter-Domain Routing (SIDR) working group proposed the RPKI using BGPSEC to validate the AS_PATH attribute, but BGPSEC is currently a work in progress [9]. In addition, a study propounds that BGP armed with BGPSEC cannot be secured because of BGP's fundamental design [10].

We propose Secure AS_PATH BGP (SAPBGP) in which the SAPBGP constructs its own policy-based database by collecting RIPE NCC repository and checks the AS_PATH attribute in BGP update messages whether or not the ASes listed in the AS_PATH attribute are actually connected. For the validation test with the real BGP messages, the SAPBGP receives a live BGP stream from the BGPmon project [11]. In addition, we conduct the performance test of the SAPBGP to measure the duration of the validation with the live BGP messages.

In this paper, with the fact that BGP is vulnerable to MITM attack, we describe an attack scenario and a solution in Section 3. In Section 4, we introduce and explain the SAPBGP in detail. We discuss the SAPBGP environment and analyze the result of the SAPBGP validation and the

 ISBN: 978-1-61208-376-6

performance test in Section 5. Lastly, we conclude the paper in Section 6.

II. RELATED RESEARCH

A. BGPsec

BGPsec is a mechanism to provide routing path security for BGP route advertisements and a work in progress by SIDR [9]. BGPsec relies on RPKI where the root of trust consists of the Regional Internet Registries (RIRs), including ARIN, LACNIC, APNIC, RIPE, and AFRINIC. Each of the RIRs signs certificates to allocate their resources. RPKI provides Route Origination Authorization (ROA) to ASes that are authorized to advertise a specific prefix [12]. The ROA contains the prefix address, maxlength, and AS number, which certifies the specified AS has permission to announce the prefixes. For routing path validation, each AS receives a pair of keys, which are a private key and a public key, from its RIR. Each AS speaker signs the routing path before forwarding it to their neighbors.

B. BGPmon

BGPmon is a monitoring infrastructure, implemented by Colorado State University that collects BGP messages from various routers that are distributed and offers the BGP messages as the routes for destinations are changed in real-time [5]. Any BGP speaker can be a source that offers real-time update messages if the BGP speaker is connected to BGPmon. Currently, 9 BGP speakers are participated in the BGPmon project as a source router. In addition, BGPmon collects Multi-threaded Routing Toolkit (MRT) format [13] live stream from the RouteViews project through indirect peering. The MRT format defines a way to exchange and export routing information through which researchers can be provided BGP messages from any routers to analyze routing information. Clients can be connected to the BGPmon via telnet and receive the live BGP stream in real time.

C. RIPE NCC

RIPE NCC is one of the Regional Internet Registries (RIRs) in charge of the Europe/Middle-East region. RIPE NCC manages RIPE Data Repository that is a collection of datasets, such as IP address space allocations and assignments, routing policies, reverse delegations, and contacts for scientific Internet research. The organizations or individuals who currently hold Internet resources are responsible for updating information in the database. As a result, RIPE NCC can keep the Data Repository up to date and provide database APIs so that data users can access the RIPE data repository through web browsers or programs.

III. BGP THREATS AND SOLUTION

In this section we introduce a scenario of the AS path hijacking that leads to the MITM attack. In addition, we discuss how the routing policy-based AS_PATH validation is operated in order to prevent the AS path hijacking.

A. Manipulating data in BGP updates

A BGP router inserts its own ASN into the AS_PATH attribute in update messages when the BGP router receives the update message from neighbors. However, the BGP router can insert one or more ASNs into the AS_PATH attribute in update messages other than its own ASN. In addition, a BGP router might pretend as if the BGP router is connected to a certain BGP router by manipulating data contained in BGP updates.

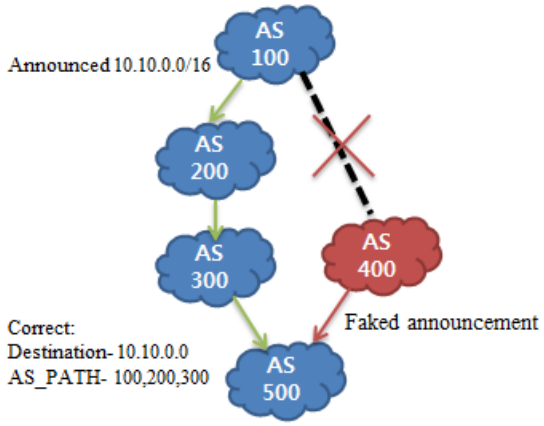


Figure 1. Manipulating a BGP message

Figure 1 demonstrates an example of manipulating data in BGP update messages. Suppose AS 400 has a connection to AS 500 and creates a faked\ BGP announcement to pretend that AS 400 received a BGP message originated by AS 100 and forwarded the update message to AS 500 even though AS 100 and AS 400 actually don't have a BGP connection. In terms of AS 500, the traffic heading for prefix 10.10.0.0/16 will choose AS 400 as the best path because AS 500 selects the shortest path and AS 400 is shorter than AS 300. Even if the AS 500 can conduct origin validation, the AS 500 cannot prevent this attack because prefix and ASN information is correct. As a result, AS 400 will have the traffic heading for prefix 10.10.0.0 and might start another attack using the traffic, such as a MITM attack.

B. Man-in-the-middle (MITM) attack

The man-in-the-middle attack is an active eavesdropping in which the attacker secretly creates connections to the victims and redirects large blocks of internet traffic between the sources and the destinations as if the sources and destinations communicate directly. In such a case, the victims can only notice a little enlarged latency time because the internet packets travel longer hops than normal. In the meantime, the attacker can monitor and manipulate the packets so that the attacker can create future chances to try another attack.

Renesys monitors MITM attacks and its clients were victims of route hijacking caused by MITM attacks for more than 60 days. The victims are governments, Internet Service Providers (ISP), financial institutions, etc.

 ISBN: 978-1-61208-376-6

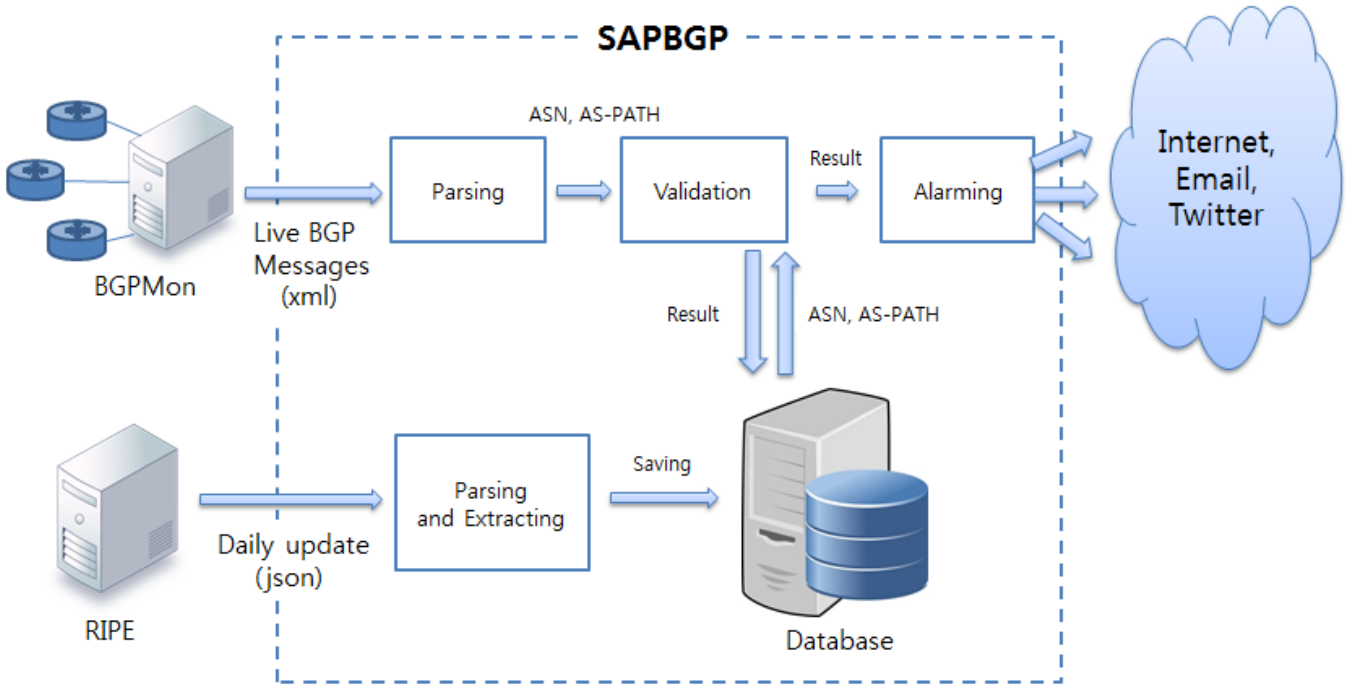


Figure 2. The architecture of the SAPBGP

C. *Routing policy based AS_PATH Validation*

RIPE NCC provides users with RIPE Data Repository that contains BGP peer information. Through this information, we can know if any ASes are connected to other ASes. This peer information has been collected by either Routing Information Service (RIS) or Internet Routing Registry (IRR). RIS has collected and stored Internet routing data from several locations all over the world since 2001.

Using peer information, the SAPBGP monitors live BGP stream from BGPmon. For example, in Figure 1, suppose that AS 400 pretends as if AS 400 is connected to AS 100, and AS 400 creates a BGP message as if the BGP message is coming from AS 100 and forwarding the BGP message. Then, AS 500 cannot check AS 400 and AS 100 are connected to each other even though the AS 500 can conduct the origin validation. However, suppose that either AS 500 or one of AS 500's neighbors is a BGPmon's participant and the SAPBGP can receive the live BGP stream related to AS 500. The AS_PATH attribute in the BGP stream should contain AS_PATH-100, 400, 500. Then, the SAPBGP can find that AS 100 and AS 400 are not connected to each other according to the peer information collected from RIPE NCC repository. As a result of this, an AS 500 administrator will be alerted by the SAPBGP and realize AS 400 might be trying the MITM attack to draw AS 500 traffic heading to AS 100.

IV. SECURE AS_PATH BGP

In this section, we introduce overall how the SASBGP works and Figure 2 describes the architecture of the SAPBGP.

A. *Constructing Database*

We construct our own database by using API provided by RIPE. We have collected, every day, all of the AS imports and exports policies information since the eighteenth of February in 2014. In addition, we have separated tables in the database to keep the daily information as well as the accumulated information by adding new exports and imports to the existing exports and imports.

As of the sixth of June in 2014, there are 77,776 ASes in the world. We sent queries to RIPE one by one. For example, if a query is related to AS 1 then the result includes AS 1's export policies, imports polices, and prefixes in the form of json. The SAPBGP parses the results so that the list of export policies and import policies can be stored to AS 1's record in the table. As a result, a new table is created every day to keep track of the daily policy information. In addition, the accumulated table is updated by adding new policies if AS 1 adds new policies against other ASes. Figure 3 shows the records from AS 10001 to AS 10005 in the policy table.

asn	export	import
10001	4680,2497,2516	
10002	2497,17224,9002,4716,251...	17225,4716,17232,45686,4732,10015
10003	4716,6939,2516,2497	4716,2516
10004	7682,4675,4732,4686,2519	7682,4732
10005		

Figure 3. A screen capture of the policy table

B. *Monitoring Live BGP Stream*

BGPmon provides live BGP stream through telnet to the public. So, whenever the routers that are connected to

 ISBN: 978-1-61208-376-6

BGPmon receives BGP update messages, BGPmon converts BGP update messages to XML format messages and propagates the XML format messages to their clients. Apart from the BGP update message, the XML format message includes timestamp, date time, BGPmon id, BGPmon sequence number, and so on.

Currently, there are 9 participants that are directly connected to BGPmon, such as AS 3043, AS 10876, AS 3257, AS 3303, AS 812, AS 5568, AS 14041, AS 28289, and AS 12145. We measured the number of update messages that BGPmon propagates for 1 hour on the twenty sixth of February in 2014. Table I shows the minimum, maximum, and average number of update messages per 10 seconds.

TABLE I. THE NUMBER OF UPDATE MESSAGES FROM BGPMON

	The number of update messages per 10 seconds
Minimum	38
Maximum	1672
Average	119.43

After parsing the live BGP message, the SAPBGP retrieves the ASN attribute and the AS_PATH attribute to check whether ASes in the AS_PATH attribute are connected to each other. Firstly, we compare the policy table in the database that is collected one day before. If we cannot find the pair, we compare the information from the accumulated table. If we cannot find the pair from the table, we consider the AS_PATH attribute as the suspicious AS_PATH attribute. If we find the suspicious AS_PATH attribute, we notify the AS network administrators of the suspicious AS_PATH attribute.

V. PERFORMACE TEST AND RESULT ANALYSIS

We explain the environment in which the SAPBGP constructs its own database by collecting RIPE repository and check the live BGP stream from BGPmon to check the invalid AS_PATH attribute in the BGP message. In addition, we conduct the performance test and analyze the result of the performance test in this section.

A. Experiment

We have constructed our database by daily collecting BGP policy records from the RIPE repository since the eighteenth of February in 2014. Based on our table, the SAPBGP checked the live BGP stream from BGPmon.

TABLE II. THE COMPARISON OF THE RESULTS

	Original results	***No duplication***
Valid	230575	13490
Invalid	3931	656
Valid by the accumulated records	4508	205

Table II shows the comparison between the original results and the result that does not contain duplications. Because of the difference of variation of BGP update periodic time, some pairs of ASes can be more duplicated than others.

Figure 4 shows the result of the AS_PATH monitoring experiment through the SAPBGP from the eighteenth of February in 2014 to the eighth of June in 2014. We conducted the experiment once a week during that period. The original data collected contains many duplicated results, but the outcome in Figure 4 does not contain the duplications. Our result shows 4.57% of the AS_PATH attribute is invalid and 95.43% of the AS_PATH attribute is valid.

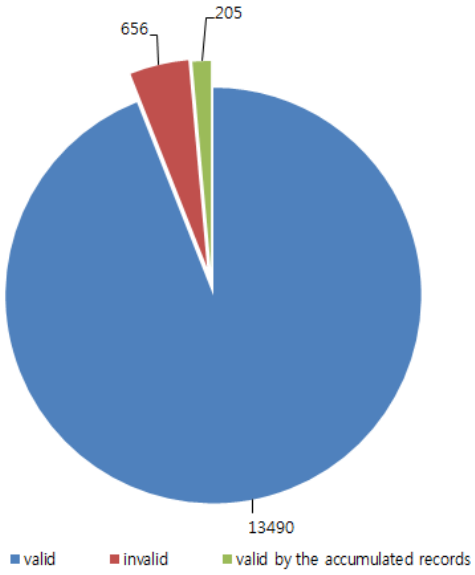


Figure 4. The result of the AS_PATH monitoring experiment

Figure 5 illustrates a portion of the policy table of the invalid ASes that the SAPBGP detected in the experiment. The invalid ASes could signify either the AS holder does not configure policies or the AS_PATH attribute was manipulated by hijackers.

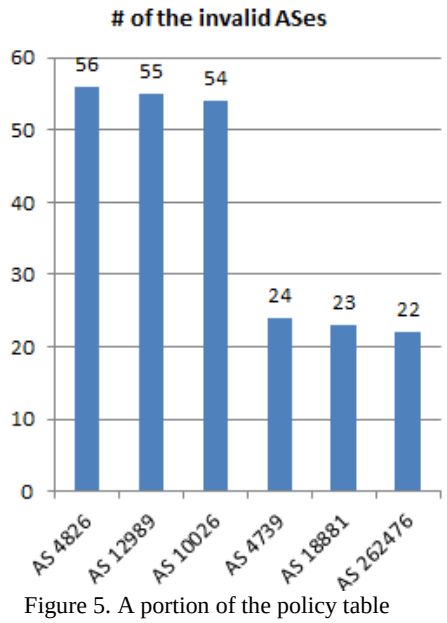


Figure 5. A portion of the policy table

B. *Performance Test*

The SAPBGP runs on a 3.40 GHz i5-3570 machine with 16 GB of memory running Windows 7. MySQL Ver. 14.14 Distrib 5.1.41 is used for the database. The SAPBGP is implemented by JAVA to collect daily updates from RIPE, to receive live BGP stream from BGPmon, and to validate the BGP stream by comparing the AS_PATH attribute to our database. The SAPBGP and database are located in the same machine to reduce the connection latency between them.

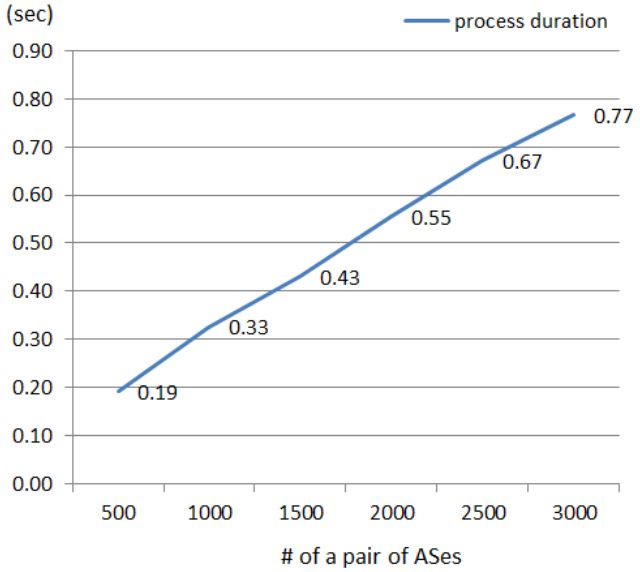


Figure 6. The result of the performance test for the AS_PATH validation

Figure 6 shows the AS_PATH validation time. The validation time includes accessing database, retrieving the specific AS record from a table, and comparing the AS_PATH attribute to the AS's record. It takes 256 microseconds, on average, to validate a pair of ASes. According to Table 1, the maximum number of live BGP messages for 10 seconds is 1672. So, the SAPBGP can process all of the live BGP messages coming from BGPmon in real time.

VI. CONCLUSION

Even though many solutions are proposed to prevent IP prefix hijacking, such as RPKI, BGPmon, Argus, and PHAS, these solutions cannot protect the AS path hijacking except RPKI. SIDR proposed the RPKI using BGPSEC but BGPSEC is currently a work in progress. In order to monitor the AS path hijacking, we propose Secure AS_PATH BGP (SAPBGP) in which we monitor the AS_PATH attribute in update messages whether each AS in the AS_PATH attribute are connected to each other based on our policy database collected from RIPE NCC repository. The result of the AS_PATH validation test shows 4.57% of the AS_PATH attribute is invalid and 95.43% of the AS_PATH attribute is valid from the fifteenth of April in 2014 to the eighth of June in 2014. In addition, the result of performance test verifies that the SAPBGP can process all of the live BGP messages coming from BGPmon in real time. In the result of the AS_PATH monitoring experiment, the ratio of invalid AS_PATH attribute is high because some AS routers still do not configure their policies. For the precise result of the policy based AS_PATH validation, every router needs to configure policies against its peers.

REFERENCES

[1] Y. Rekhter, "A Border Gateway Protocol 4 (BGP-4)," 2006, RFC 4271.

[2] S. Murphy, "BGP Security Vulnerabilities Analysis," 2006, RFC 4272.

[3] Rensys Blog, Pakistan hijacks YouTube [Online]. Available: http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml [Accessed February 2014].

[4] T. Manderson, L. Vegoda, and S. Kent, "Resource Public Key Infrastructure (RPKI) Objects Issued by IANA(Feb. 2012)," 2012, [Online]. Available: http://www.rfc-editor.org/rfc/rfc6491.txt [Accessed January 2014].

[5] BGPmon, Google's services redirected to Romania and Austria [Online]. Available: http://www.bgpmon.net/googles-services-redirected-to-romania-and-austria [Accessed October 2013].

[6] X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu., "Detecting Prefix Hijackings in the Internet with Argus", In Proc. of ACM IMC 2012.

[7] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A prefix hijack alert system," 2006, In Proceedings of the 15th conference on USENIX Security Symposium - Volume 15 (USENIX-SS'06), Vol. 15, pp.153-166.

[8] Renesys Blog, Targeted Internet Traffic Misdirection [Online]. Available: http://www.renesys.com/2013/11/mitm-internet-hijacking [Accessed January 2014].

[9] M. Lepinski, Ed., and BBN, "BGPSEC Protocol Specification," Available: http://tools.ietf.org/html/draft-ietf-sidr-bgpsec-protocol-08.

[10] Q. Li, Y. Hu, and X. Zhang, "Even Rockets Cannot Make Pigs Fly Sustainably: Can BGP be Secured with BGPsec?," 2014.

[11] The BGPmon project, http://bgpmon.netsec.colostate.edu, [Accessed 6th July 2013].

[12] M. Lepinski, S. Kent, and D. Kong, "A Profile for Route Origin Authorizations (ROAs)," [Online]. Available: http://tools.ietf.org/html/rfc6482, [Accessed December 2012].

[13] L. Blunk, "Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format," RFC 6396 , 2011.

 ISBN: 978-1-61208-376-6

A New Property Coding in Text Steganography of Microsoft Word Documents

Ivan Stojanov, Aleksandra Mileva, Igor Stojanović

University of Goce Delčev
Štip, Macedonia
Email: {ivan.stojanov, aleksandra.mileva, igor.stojanovik}@ugd.edu.mk

***Abstract*—Electronic documents, similarly as printed documents, need to be secured by adding some specific features that allow efficient copyright protection, authentication, document tracking or investigation of counterfeiting and forgeries. Microsoft Word is one of the most popular word processors, and several methods exist for embedding data specially in documents produced by it. We present a new type of methods for hiding data in Microsoft Word documents, named as Property coding, which deploys properties of different document objects (e.g., characters, paragraphs, and sentences) for embedding data. We give four different ways of Property coding, which are resistant to save actions, introduce very small overhead on the document size (about 1%), can embed up to 8 bits per character, and of course, are unnoticed by readers. Property coding belongs to format based methods of text steganography.**

Keywords–Data Hiding; Microsoft Word.

I. INTRODUCTION

Steganography is the art of undetectably altering some seemingly innocent carrier to embed or hide secret messages. Modern digital steganography utilizes computers and new information technologies, and one can use an image, text, video, audio, file, protocol header or payload, or similar, as a carrier. Watermarking, on the other hand, is the art of imperceptibly altering some carrier, to embed a message about that carrier. Each steganographic and watermarking system consist of an embedder and a detector, the carrier is called cover work, and the result of embedding is called stego (watermarked) work [1]. Information hiding (or data hiding) is a general term encompassing a more wide range of problems, and it includes steganography and watermarking also.

Text steganography refers to the hiding of information within text (see surveys [2][3]). Text is one of the oldest media used for hiding data, and before the time of digital steganography, letters, books, and telegrams were used to hide secret messages within their texts. Also, text documents are the most present digital media today, which can be found in the form of newspapers, books, web pages, source codes, contracts, advertisements, etc. So, development of text steganography and steganalysis is very important. From one side, data hiding methods in text documents are big threats to cybersecurity and new communication tools for terrorists and other criminals. On the other side, these methods can have legal application in document tracking, copyright protection, authentication, investigation of counterfeiting and forgeries, etc. [4][5][6].

Microsoft Word is one of the most popular document and word processing software, which comes as a part of the Microsoft Office package. It is attractive for average users because of the easiness of text editing and richness of text formatting features.

In this paper, we present four new methods for hiding data in MS-Word documents. We use properties of different document objects, like characters, paragraphs, and sentences, for data hiding. Additionally, these techniques can be adjust for using in the documents produced by other word processors, like Apache OpenOffice, Corel WordPerfect, etc. Section II is devoted to different techniques used in text steganography and Section III gives several existing methods and techniques specially designed for MS-Word documents. Our four new methods are presented in Section IV, and experimental results and discussion are given in Section V.

II. TEXT STEGANOGRAPHY

There are three main categories of text steganography: format based methods, random and statistical generation, and linguistic methods [7].

A. Format based methods

Format based methods generally format and modify existing text to conceal the data. There are several different techniques for hiding data in text documents presented bellow. Some of them like line shift coding or inserting of spacial characters can pass unnoticed by readers, but can be detected by computer; and other like font resizing, can pass undetected by computer, but human can detect it. Hidden information usually can be destroyed for example by character recognition programs.

1) Line Shift Coding: In line shift coding, each even line is shifted by a small predetermined amount (e.g., 1/300 inch and less) either up or down, representing binary one or zero, respectfully [8][9][10]. The odd lines are used as control lines for detection of shifting of the even lines, and their position is static. In this way, the original document is not needed for decoding.

2) Word Shift Coding: Similarly to line shifting coding, in word shifting coding, each even word is shifted by a small predetermined amount (e.g., 1/150 inch and less) left or right, representing binary one or zero, respectfully [9][10]. Again, each odd word serves as a control word, which is used for measuring and comparing distances between words. Since the word spacing in the original document is not uniform, the original document is needed for decoding. Low, Maxemchuk, Brassil, and O'Gorman [8] use combination of line and word shifting, and each even line additionally is divided in three blocks of words and only middle block is shifted left or right. In [11], line is divided in segments of consecutive words, and neighbouring segments share one word. By shifting only middle words of the segment, 1 or 2 bits can be coded per one segment.

3) Feature Coding: In feature coding (or character coding), the feature of some characters in the text are changed [9][10]. For example, change to an individual character's height or its position relative to other characters; extending or shortening

 ISBN: 978-1-61208-376-6

of the horizontal line in the letter t; increasing or decreasing the size of the dot in letters i and j, etc. The last technique can be applied for 14 letters in Arabic alphabet [12]. Another feature coding methods for Arabic alphabet [13][14] use the redundancy in diacritics to hide information.

4) Open method: In this group of techniques, some special characters are inserted in the cover text. For example, spaces can be inserted at the end of each sentence, at the end of each line, between words [15], or at the end of each paragraph [16]. A text processor can change the number of spaces and destroy the hidden message. There are several software tools, which implement some variants of the open method, like SNOW [17], WhiteSteg [18], UniSpaCh [19] which uses Unicode space characters, etc.

Other techniques [20][21], which can be put in this group, use widening, shrinking or unchanging an inter-word space to encode the text format.

5) Luminance Modulation Coding: This coding uses character luminance modulation for hiding data. Borges and Mayer [22] embed data by individually altering the luminance of each character from black to any value in the real-valued discrete alphabet of cardinality S, so that each symbol represents $log_2 S$ bits. One previous method [23], instead of whole character, modulates the luminance of particular pixels from the characters in scanned text document for hiding bits. Similarly in [24], quantization of the color intensity of each character is used, in such a way the HVS cannot make the difference between original and quantized characters, but it is possible for a specialized reader. This technique works well on printed documents, too.

B. Random and Statistical Generation

In methods of random and statistical generation, a new text is generated, which tries to simulate some property of normal text, usually by approximating some arbitrary statistical distribution found in real text [7].

C. Linguistic Methods

Linguistic methods manipulate with lexical, syntactic, or semantic properties of texts for hiding data, while their meanings are preserved as much as possible. Known linguistic methods are syntactic and semantic methods.

With syntactic methods, data can be hidden within the syntactic structure itself. They sometimes include changing the diction and structure of text without significantly altering meaning or tone. Some of them use punctuation, because there are many circumstances where punctuation is ambiguous or when mispunctuation has low impact on the meaning of the text. For example, one can hide one or zero by putting or not, a comma before ”and” [15]. One disadvantage is that inconsistent use of punctuation is noticeable to the readers. In Arabic language, there is one special extension character, which is used with pointed letters, without effect on the content. The authors of [25] suggest to use pointed letters with extension as binary one and pointed letters without extension as binary zero. Wayner [26] proposed Context-Free Grammars (CFGs) to be used as a basis for generation of syntactically correct stego texts. Another method [27] manipulates with sentences by shifting the location of the noun and verb to hide data.

Semantic methods change the words themselves. One method uses the synonym substitution of words for hiding information in the text [15]. Two different synonyms can be used as binary one and zero. Similar is use of paraphrasing of text for hiding messages [28], for example ”can” for binary 0, and ”be able to” for binary 1. Another method [29] changes word spelling, and in order to code zero or one, the US and UK spellings of words are used. One example is the word ”color”, which has different spelling in UK (colour) and US (color). Other semantic methods are given in [5][30]. Semantic methods sometimes can alter the meaning of the sentence.

Different miscellaneous techniques that use typographical errors, using of abbreviations and acronyms, free form formatting, transliterations, use of emoticons for annotating text with feelings, mixed use of languages, and similar ones are given in [31].

III. Existing Methods Specially Designed for MS-Word Documents

Besides the previous more general text steganographic methods that can be applied, there are several methods for data hiding, specially designed for Microsoft Word documents. The most closest technique to ours, is usage of invisible characters, suggested by Khairullah [32]. This technique sets foreground color on invisible characters such as the space, the tab or the carriage return characters, obtaining 24 bits per character.

Another technique, called Similar English Font Types (SEFT) [33], use similar English fonts for hiding data. First, three different similar fonts are chosen (e.g., Century751 BT, CenturyOldStyle, CenturyExpdBT), and then, 26 letters and space character are represented by triple of capital letters, each in one of the chosen fonts.

Liu and Tsai [34] use Change Tracking technique for hiding data in MS-Word documents. First, a cover document is degenerated with different misspellings and other mistakes usual for users, and then, corrections with Change Tracking are added, so it seems like the document is the product of a collaborative writing effort. The secret message is embedded in the choices of degenerations using Huffman coding.

From MS-Office 2007, Microsoft has adopted a new format of its files, and introduced the Office Open XML (OOXML) format. In order to guarantee higher level of privacy and security, it has also presented the feature Document Inspector, which is used for quickly identifying and removing of any sensitive, hidden and personal information. Castiglione et al. present in [35] four new methods for hiding data in MS-Word documents, which resist the Document Inspector analysis. Two of them (with different compression algorithms or revision identifier values) exploit particular features of the OOXML standard, have null overhead, but do not resist to save actions. Other two (with zero dimension image or macro), resist to save actions, but they have an overhead.

IV. Property Coding

We present four new format based methods for hiding data in MS-Word documents, that use some text formattings that are invisible for human eye. They use different choices for some text properties, and because of that, we can name them as Property Codings. Methods presented in [32] and [33] can be also classified as Property codings, because they use font color and font type properties of a given character, respectfully. The novelty of our methods is twofold. First, we introduce other character properties that can be used for hiding data, and second, we show that properties of document objects other than characters (e.g., paragraphs and sentences), can be used for hiding data.

 ISBN: 978-1-61208-376-6

A. Method 1 - Character Scale

When we work in MS-Word, by default text character scale is set to 100%. Increasing the character scale will make your letters larger and scale further apart with more white space between each character. Decreasing the scale will shrink and squish letters closer together. Big differences in character scale are noticeable for human reader. But, if some of the characters are with scale 99% and others with 101%, human eye can not make the differences.

So, in the first method, we use scale of 99% to represent binary one, and scale of 101% to represent binary zero. Scale of 100% can be used for non-encoded characters. In this way, in the cover document, we can hide maximum the same number of bits as the number of characters in the document.

Variants of this method are also possible. For example, instead of using two very close scale values, one can uses four very close scale values **(e.g.,** 97%, 98%, 99% and 101%), and every value will represents two binary digits. In this way, we duplicate the hiding capacity of the same document, and still normal reader won't notice it. Another variant is to change scale on every word, not on every character.

B. Method 2 - Character Underline

One common feature of MS-Word is character underlining. There are 16 different underline styles, with potential of carrying 4 bits, and 2^{24} different underline colors. Because we need underlining to go unnoticed by the user, we use 16 variants of white color, with potential of carrying 4 bits.

In this way, we can hide 8 bits per character. Some characters, as g, j, p, q, and y, have noticeable changes in the look when we use every type of underlining. Because of that, we excluded this group of 5 characters from hiding data.

C. Method 3 - Paragraph Borders

In MS-Word, one can add border to the paragraph, sentence, picture, table, individual page, etc. Border can be left, right, top, bottom, etc. There are 24 different border styles, and only two of them (wdLineStyleEmboss3D and wdLineStyleEngrave3D) are noticeable to human reader. We can use 16 out of the rest 22, with potential of carrying 4 bits.

In this method, we use left and right borders on paragraph for hiding data. Again, we use 16 variants of white color for borders. Each paragraph in the cover document can hide 16 bits, in the following way - 4 bits from left border style, 4 bits from left border color, 4 bits from right border style, and 4 bits from right border color. This is done in our implementation.

We can increase hiding capacity of this method, by using different border width also. There are 13 border styles with 9 different border widths, two border styles with 6 different border widths, three border styles with 5 different border widths, one border style with 8 different border widths, one border style with 2 different border widths, and two border styles with 1 border width, or summary 155 possibilities. Potentially, we have 7 bits per combination border style/width. With experiments, we obtained that RGB colours represented with (R, G, B) components, where $R, G, B > 249$ can not be distinguished from the white color $(255, 255, 255)$. There are 216 different possibilities for colour, which can be used for representing 7 bits. Combining these two techniques, we can hide 28 bits, in the following way - 7 bits from left border style, 7 bits from left border color, 7 bits from right border style, and 7 bits from right border color.

TABLE I. CHARACTERISTICS OF THREE COVER DOCUMENTS

	Document 1	**Document 2**	**Document 3**
Pages	1	11	110
Words	340	2381	30907
Characters	2252	15493	190833
Paragraphs	13	82	802
Lines	42	328	3445
Sentences	21	134	2028
Original size (B)	31122	923090	4589312

TABLE II. COMPARISON OF MAXIMAL NUMBER OF EMBEDDED BITS/CHARACTERS IN OUR METHODS AND METHODS PRESENTED IN [32] AND [33]

	Document 1	Document 2	Document 3
Characters without q, j, p, q, y	2154	14823	182470
Invisible Characters	364	2515	31422
Percent of Invisible Characters	16,2	16,2	16,5
Capital Letters	40	286	4704
Max No. of embedded bits in Method 1	2252	15493	190833
Max No. of embedded bits in Method 2	17232	118584	1459760
Max No. of embedded bits in Method 3	364	2296	22456
Max No. of embedded bits in Method 4	147	938	14196
Max No. of embedded bits in [32]	8736	60360	754128
Max No. of embedded characters in [33]	13	95	1568
Max No. of embedded bits in [33]	104	760	12544

D. Method 4 - Sentence Borders

The final method uses sentence outside border for hiding data. We use only 8 border style out of 16, because other 8 can be noticed by human reader, and only the smallest border width of 0.25pt. Used border styles are wdLineStyleDashDot, wdLineStyleDashDotDot, wdLineStyleDashLargeGap, wdLineStyleDashSmallGap, wdLineStyleDot, wdLineStyleInset, wdLineStyleOutset and wdLineStyleSingle. Each sentence in the cover document can hide 7 bits, with 3 bits from outside border style, and 4 bits from outside border color.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

Each new presented method has implementation in C♯ using the Microsoft.Office.Interop.Word namespace. Our implementation of these four methods, use 8 bits to represent an extended ASCII character for all methods, except for the last, were we use 7 bits to represent an ASCII character. For our experiments, we use three types of MS-Word documents as cover documents - short, medium and large documents, with properties given in Table I.

For each cover document, we hide 10, 50, 100, 500, 1000, and 5000 characters (if it is possible), and we measure the size of the obtained stego document. Normally, the new size is bigger than original size, and it is given in bytes and in percent of increase of original size.

From the results in Tables III, IV and V, one can see that all techniques have small impact of document size, less then

TABLE III. EXPERIMENTAL RESULTS FOR DOCUMENT 1 WITH ORIGINAL SIZE OF 31122B

	10 characters		50 characters		100 characters		500 characters		1000 characters		5000 characters	
	Size	**%**	**Size**	**%**	**Size**	**%**	**Size**	**%**	**Size**	**%**	**Size**	**%**
Method 1	31448	1.01047	32347	1.03936	33390	1.04074	/	/	/	/	/	/
Method 2	31249	1.00408	31530	1.01310	31986	1.02776	34482	1.10796	37517	1.20548	/	/
Method 3	31295	1.00555	/	/	/	/	/	/	/	/	/	/
Method 4	31356	1.00751	/	/	/	/	/	/	/	/	/	/

TABLE IV. EXPERIMENTAL RESULTS FOR DOCUMENT 2 WITH ORIGINAL SIZE OF 923090B

	10 characters		50 characters		100 characters		500 characters		1000 characters		5000 characters	
	Size	**%**	**Size**	**%**	**Size**	**%**	**Size**	**%**	**Size**	**%**	**Size**	**%**
Method 1	923609	1.00056	924750	1.00179	925472	1.00258	934834	1.01272	946697	1.02557	/	/
Method 2	924243	1.00124	924605	1.00164	925180	1.00226	926341	1.00352	928582	1.00624	953474	1.03291
Method 3	923455	1.00039	924398	1.00141	924547	1.00157	/	/	/	/	/	/
Method 4	923587	1.00053	924013	1.00099	925290	1.00238	/	/	/	/	/	/

TABLE V. EXPERIMENTAL RESULTS FOR DOCUMENT 3 WITH ORIGINAL SIZE OF 4589312B

	10 characters		50 characters		100 characters		500 characters		1000 characters		5000 characters	
	Size	**%**	**Size**	**%**	**Size**	**%**	**Size**	**%**	**Size**	**%**	**Size**	**%**
Method 1	4589321	1.00000	4589363	1.00001	4591027	1.00037	4595001	1.00123	4605370	1.00349	4682285	1.02025
Method 2	4589313	1.00000	4589356	1.00000	4589574	1.00005	4592093	1.00060	4595782	1.00140	4608077	1.00408
Method 3	4589512	1.00004	4589567	1.00005	4589597	1.00006	4591231	1.00041	4593443	1.00090	/	/
Method 4	4589376	1.00001	4589396	1.00011	4591778	1.00010	4595859	1.00142	4603958	1.00319	/	/

1.206% for Document 1, less then 1.033% for Document 2, and less then 1.021% for Document 3 for evaluated message lengths. Method 2 has the smallest influence on the size for the short and large documents, and Method 3 has the smallest influence on the size for the medium document.

From the Table II, one can see that Method 2 has the highest embedding capacity, followed by Method 1, and the smallest embedding capacity has Method 4. The number of invisible characters is only a small portion of the number of all characters in every document, and in our three documents is less then 17% (see Table II). So, if we compare our Method 2 with the method introduced by Khairullah [32] (Table II), we can embed more characters by Method 2. One can see that for all three documents, the maximal number of embedded bits by [32] ('number of invisible characters' $\times 24$) is almost a half than the maximal number of embedded bits by Method 2 ('number of characters, without q, p, j, y, and g' $\times 8$). For the method proposed by Bhaya et al. in [33], we have that three consecutive capital letters in the document serve to embed one character, so, the maximal number of embedded bits depends strongly of number of capital letters. If we use 8 bits per character, we have that this method has the smallest embedding capacity compared to other analyzed methods (Table II). Even in the case that all characters are capital letters, we can embed almost three times less characters, than in the case of Method 2. Bhaya et al. in [33] suggested to use only three similar font types, which limits the maximal number of different characters that can be embedded to 27. This can be changed if we use four or five similar font types, resulting in up to 64 and 125 different characters. But finding bigger number of similar fonts is very difficult, and at the end, user may notice the differences in the font used for capital letters. Additional problem can arise if non-Latin language is used and if selected font is not present on the machine. For example, if you use Cyrillic letters, and font is not present, the capital letters will be displayed as Latin,

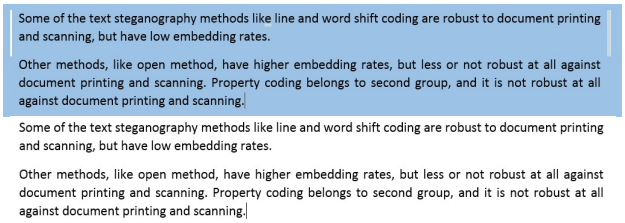


Figure 1. Detection of hiding with Method 2 and 3 by changing page background color

and coding will be visible to human eyes.

A. *Robustness and Steganalysis*

Some of the text steganography methods like line shift coding, word shift coding, and luminance modulation coding are robust to document printing and scanning, but have low embedding rates. Other methods, like open method, have higher embedding rates, but are less or not robust at all against document printing and scanning. Property coding belongs to second group, and it is not robust at all against document printing and scanning. Property Coding is resistant to save actions, compared to two methods presented in [35], and also has smaller overhead compared to other two methods from [35].

Hidden text with Property Coding can be changed or destroyed by text editing. The presence of Methods 2, 3, and 4 can be easily detected if somebody changes intentionally the background color of the document, causing the borders and underlining to became visible (see Figure 1). Method 1 is resistant to this kind of attack.

Property Coding is not entirely suitable for copyright protection applications where robust data-hiding is required, because the attacker can always use Optical Character Recog-

 ISBN: 978-1-61208-376-6

nition (OCR) to completely remove the hidden data.

VI. Conclusion

Four new format based methods specially designated for hiding data in MS-Words documents are given. Because they change the properties of some document objects offered by MS-Word, we called the new type of methods Property Coding. These methods are resistant to saving actions, introduce very small overhead on the document size, and can embed up to 8 bits per character.

References

[1] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, Eds., Digital Watermarking and Steganography. Elsevier Inc., Burlington, MA, 2008, ISBN: 978-0-12-372585-1.

[2] H. Singh, P. K. Singh, and K. Saroha, "A Survey on Text Based Steganography," in Proceedings of the 3^{rd} National Conference INDIACom-2009 2009, New Delhi, India, 2009, pp. 3–9.

[3] M. Agarwal, "Text Steganographic Approaches: A Comparison," International Journal of Network Security & Its Applications, vol. 5(1), 2013, pp. 91–106.

[4] M. J. Atallah et al., "Natural language watermarking: design, analysis, and a proof-ofconcept implementation," in Proceedings of the 4^{th} International Workshop on Information Hiding April 25-27, 2001, Pittsburgh, USA. Springer Berlin Heidelberg, Apr. 2001, pp. 185–200, Moskowitz, I. S., Ed., LNCS: 2137, ISBN: 978-3-540-45496-0.

[5] M. Atallah et al., "Natural Language Watermarking and Tamperproofing," in Proceedings of the 5^{th} International Workshop on Information Hiding October 7-9, 2002, Noordwijkerhout, Netherlands. Springer-Verlag Berlin Heidelberg, Oct. 2003, pp. 196–212, Petitcolas , F. A. P., Ed., LNCS: 2578, ISBN: 3-540-00421-1.

[6] M. Topkara, C. M. Taskiran, and E. J. Delp, "Natural language watermarking," in Proceedings of the SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents VII, 2005, vol. 5681, 2005, doi: 10.1117/12.593790.

[7] K. Bennett, "Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text," 2004, cERIAS Tech Report 2004-13.

[8] S. H. Low, N. F. Maxemchuk, J. T. Brassil, and L. O'Gorman, "Document marking and identification using both line and word shifting," in Proceedings of the 14^{th} Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '95) April 2-6, 1995, Boston, Massachusettes, Apr. 1995, pp. 853–860.

[9] J. T. Brassil, S. Low, N. F. Maxemchuk, and L. O'Gorman, "Electronic Marking and Identification Techniques to Discourage Document Copying," IEEE Journal on Selected Areas in Communications, vol. 13 (8), 1995, pp. 1495–1504.

[10] J. T. Brassil, S. Low, and N. F. Maxemchuk, "Copyright protection for the electronic distribution of text documents," Proceedings of the IEEE, vol. 87 (7), 1999, pp. 1181–1196.

[11] Y. Kim, K. Moon, and I. Oh, "A Text Watermarking Algorithm based on Word Classification and Interword Space Statistics," in Proceedings of the 7^{th} International Conference on Document Analysis and Recognition (ICDAR'03) August 3–6, 2003, Edinburgh, Scotland. IEEE Computer Society Washington, DC, USA, Aug. 2003, pp. 775–779.

[12] M. Shirali-Shahreza and S. Shirali-Shahreza, "A New Approach to Persian/Arabic Text Steganography," in Proceedings of the 5^{th} IEEE/ACIS international Conference on Computer and Information Science and 1st IEEE/ACIS July 2006, Honolulu, USA, Jul. 2006, pp. 310–315.

[13] M. Aabed, S. Awaideh, A.-R. Elshafei, and A. Gutub, "Arabic Diacritics Based Steganography," in Proceedings of the IEEE International Conference on Signal Processing and Communications (ICSPC 2007) November 24–27, 2007, Dubai, UAE, Nov. 2007, pp. 756–759.

[14] A. A. Gutub, L. M. Ghouti, Y. S. Elarian, S. M. Awaideh, and A. K. Alvi, "Utilizing Diacritic Marks for Arabic Text Steganography ," Kuwait Journal of Science & Engineering, vol. 37 (1), 2010, pp. 1–16, ISSN: 1024-8684.

[15] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," IBM Systems Journal, vol. 35, 1996, pp. 313–336.

[16] A. M. Alattar and O. M. Alattar, "Watermarking electronic text documents containing justified paragraphs and irregular line spacing," in Proceedings of the SPIE - Security, Steganography, and Watermarking of Multimedia Contents VI June, 2004, San Jose, California, USA. Society of Photo Optical, Jun. 2004, pp. 685–695.

[17] M. Kwan, "The SNOW Home Page," 2006, URL: http://www.darkside.com.au/snow/ [accessed: 2014-03-03].

[18] L. Y. Por and B. Delina, "Whitesteg: a new scheme in information hiding using text steganography," WSEAS Transaction on Computers, vol. 7, 2008, pp. 735–745.

[19] L. Y. Por, K. Wong, and K. O. Chee, "UniSpaCh: A text-based data hiding method using Unicode space characters," The Journal of Systems and Software, vol. 85, 2012, pp. 1075–1082.

[20] C. Chen, S. Z. Wang, and X. P. Zhang, "Information Hiding in Text Using Typesetting Tools with Stego-Encoding," in Proceedings of the First International Conference on Innovative Computing, Information and Control August 30 - September 1, 2006, Beijing, China, 2006, pp. 459–462.

[21] I.-C. Lin and P.-K. Hsu, "A Data Hiding Scheme on Word Documents using Multiple-base Notation System," in Proceedings of the 6^{th} International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'10) October 15-17, 2010, Darmstadt, Germany, Oct. 2010, pp. 31–33.

[22] P. V. K. Borges and J. Mayer, "Document Watermarking via Character Luminance Modulation," in Proceedings of the IEEE International Conference of Acoustics, Speech and Signal Processing (ICASSP 2006) May 14–16, 2006, Toulouse, France, Jul. 2006, pp. II–317, ISBN: 1-4244-0469-X.

[23] A. K. Bhattacharjya and H. Ancin, "Data embedding in text for a copier system," in Proceedings of the IEEE International Conference on Image Processing (ICIP 99) October 24-28, 1999, Kobe, Japan, Oct. 1999, pp. 245–249.

[24] R. Villán et al., "Text Data-Hiding for Digital and Printed Documents: Theoretical and Practical Considerations," in Proceedings of the SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents VIII, 2006, vol. 6072, 2006, doi: 10.1117/12.641957.

[25] A. Gutub and M. Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions," in Proceedings of the WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE), vol. 21 May, 2007, Vienna, Austria, May 2007, pp. 28–31.

[26] P. Wayner. Elsevier Inc., 2009, 3rd edition, ISBN: 978-0-12-374479-1.

[27] B. Murphy and C. Vogel, "The syntax of concealment: reliable methods for plain text information hiding," in Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents 2007, vol. 6505, 2007, doi: 10.1117/12.713357.

[28] Nakagawa, H. and Matsumoto, T. and Murase, I., "Information Hiding for Text by Paraphrasing," 2002, URL: http://www.r.dl.itc.u-tokyo.ac.jp/ nakagawa/academic-res/finpri02.pdf [accessed: 2014-03-03].

[29] M. Shirali-Shahreza, "Text Steganography by Changing Words Spelling," in Proceedings of the 10^{th} International Conference on Advanced Communication Technology (ICACT 2008) February, 2008, Kitakyushu, Japan, vol. 3, Feb. 2008, pp. 1912–1913.

[30] M. Niimi, S. Minewaki, H. Noda, and E. Kawaguchi, "A Framework for a Simple Sentence Paraphrase Using Concept Hiearchy in SD-Form Semantics Model," in Proceedings of the 13th European-Japanese Conference on Information Modelling and Knowledge Bases (EJC 2003), June 3-6, Kitakyushu, Japan. IOS Press, 2004, pp. 55–66.

[31] M. Topkara, U. Taskiran, and M. J. Atallah, "Information Hiding Through Errors: A Confusing Approach," in Proceedings of the SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents 2007, vol. 6505, 2007, doi: 10.1117/12.706980.

[32] M. Khairullah, "A Novel Text Steganography System Using Font Color of the Invisible Characters in Microsoft Word Documents," in Proceedings of the Second International Conference on Computer and Electrical Engineering (ICCEE '09) December, 2009, Dubai, Dec. 2009, pp. 482–484, ISBN: 978-0-7695-3925-6.

[33] W. Bhaya, A. M. Rahma, and D. Al-Nasrawi, "Text Steganography

based on Font Type in MS-Word Documents," Journal of Computer Science, vol. 9 (7), 2013, pp. 898–904, ISSN: 1549-3636.

[34] T.-Y. Liu and W.-H. Tsai, "A New Steganographic Method for Data Hiding in Microsoft Word Documents by a Change Tracking Technique," IEEE Transactions on Information Forensics and Security, vol. 2 (1), 2007, pp. 24–30.

[35] A. Castiglione, B. D'Alessio, A. De Santis, and F. Palmieri, "New steganographic techniques for the OOXML file format," in Proceedings of the IFIP WG 8.4/8.9 international cross domain conference on Availability, reliability and security for business, enterprise and health information systems August 22-26, 2011, Vienna, Austria. Springer, Aug. 2011, pp. 344–358, Tjoa, A. M., Quirchmayr, G., You, I., Xu, L. Eds., LNCS: 6908, ISBN: 978-3-642-23299-2.

 ISBN: 978-1-61208-376-6

Audio Steganograpgy by Phase Modification

Fatiha Djebbar
UAE University
College of Information Technology
AL Ain, UAE
Email: fdjebbar@uaeu.ac.ae

Beghdad Ayad
University of Wollongong in Dubai
Faculty of Engineering and Information Science
Dubai, UAE
Email: beghdadayad@uowdubai.ac.ae

***Abstract*—In this paper, we propose a robust steganographic system that embeds high-capacity data in phase spectrum. Our approach is based on the assumption that partial alteration of selected frequency bins in the phase spectrum leads to a smooth transition while preserving phase continuity. The frequency bins, in the phase, selected for data hiding are first defined in the magnitude through an election process and then mapped into the phase spectrum to embed data. Perceptual and statistical study results demonstrate that, in comparison with a recently proposed magnitude based audio steganography method, the phase based approach gains a considerable advantage against steganalysis attacks while giving similar or comparable hiding capacity and audio quality.**

Keywords–Information hiding; Phase Coding; Steganalysis.

I. INTRODUCTION

Digital audio steganography has emerged as a prominent source of data hiding across novel telecommunication technologies such as voice-over-IP and audio conferencing. Currently, three main methods are being used: cryptography, watermarking, and steganography. Encryption techniques are based on rendering the content of a message garbled to unauthorized people. In watermarking, data is hidden to convey some information about the cover medium such as ownership and copyright, where the hidden message could be visible or invisible. The primary goal of steganography consists of undetectably modifying a multimedia file to embed these data [1]. While steganography is about concealing the existence of 'hidden message', steganalysis is about detecting its existence [2]. Steganalysis, the counterpart of steganography, is regarded as "attacks" to break steganography algorithms by the mean of different audio processing and statistical analysis approaches.

Steganography in today's computer era is considered a sub-discipline of the data communication security domain. Lately, new directions based on steganographic approaches started to emerge to ensure data secrecy. Modern techniques of steganography exploit the characteristics of digital media by utilizing them as a carrier (cover) to hold hidden information. Covers can be of different types including image [4], audio [5], video [6], text [7], and IP datagram [8].

Several methods of audio data hiding have been proposed, whether in time or frequency domains, including low-bit coding, spread spectrum coding, phase coding, echo data hiding, etc [1]. To hide information within audio signals, [9][10] have designed a steganographic algorithm by manipulating higher LSB layers of the audio signal. Phase alteration and spread spectrum are used in [11][12]; wavelet coding is used in [13][14] and magnitude-based data hiding was proposed by [15]. Most of these methods take information hiding ratio as a major factor in evaluating the robustness of their algorithms. As it is generally expected, higher information-hiding ratio elevates the risk of detecting the presence of hidden data.

In this paper, we present a robust phase coding technique for digital audio steganography. The original contributions of the paper addresses mainly the undetectability issue of hidden data encountered in our previous work, where magnitude was solely considered [15]. The phase spectrum is explored, in particular, to benefit from the inherent advantages of phase data hiding, as it is commonly understood that, when phase coding can be used, it gives better signal to noise ratio [1]. Our work is supported by a thorough comparative study by steganalysis to judge the performance of our steganographic. The comparison is performed against our previously presented algorithm [15] and existing high capacity LSBs-based audio steganographic software: Steghide, S-Tools and Hide4PGP found respectively in [16]–[18]. Perceptual evaluation as well as the steganalysis study show that the resulting stego stream preserves the naturalness of the original signal and resists steganalysis attacks while achieving similar or comparable hiding capacity to that in [15].

The rest of the paper is organized as follows. Phase hiding algorithm is presented in Section II. Section IV describes the steps developed to recover the embedded message at the receiver's end. Section V presents the simulation experiments and subsequent evaluation results. Finally, we conclude our paper with a summary of our work and some future directions in Section VI.

II. MOTIVATION AND BACKGROUND

The particular importance of hiding data in audio files results from the prevailing presence of audio signal as an information vector in our human society. Data hiding in audio files is especially challenging because of the sensitivity of Human Auditory System (HAS). Alterations of an audio signal for data embedding purposes may affect the quality of that signal. However, data hiding in the frequency domain rather than time domain is of nature to provide better results in terms of signal to noise ratio [10]. In addition, Human auditory perception has certain particularities that must be exploited for hiding data efficiently. For example, our ability to resolve tones decreases with the increase of frequency of the tone. Thus, it is more effective for hiding data in the higher frequency regions than in low frequencies [19].

In audio signals sampled at 16 kHz and quantized at 16 bits, frequencies within the range of 50 Hz to 7 kHz are then

 ISBN: 978-1-61208-376-6

eligible to embed data. The cover audio is divided into M equal length frames. For a sampling frequency of 16 kHz, a 4 ms frame for example produces 64 samples. The resolution of each frequency component is equal to $16000/64 = 250Hz$. Thus, the first frequency component that could be used for hiding data will be 250 Hz instead of 50 Hz (the starting frequency of wide-band speech). If we consider the Fourier symmetry feature of the spectrum, the number of normalized frequencies or the number of locations to hide data within each frame will be from $F_{HDmin} = 1$ to $F_{HDmax} = 28$ in [0.25 7] kHz frequency band. In each selected energetic frequency component location, at least a bit from the payload is embedded.

III. PROPOSED HIDING ALGORITHM

In our scheme, the cover-signal is divided into M frames of 4 ms, each contains N samples, $s_c(m,n), \quad 1 \leq m \leq M$ and $1 \leq n \leq N$. The magnitude spectrum $|S_c(m,k)|$ is isolated by transforming each frame to frequency domain using Fast Fourier Transform (FFT), $S_c(m,k) = FFT(s_c(m,n))$. The hiding band is specified by $F_{HDmin} \leq k \leq F_{HDmax}$, where F_{HDmin} and F_{HDmax} are the minimum and the maximum hiding band locations. In our algorithm, we only select high energy frequency components in an attempt to minimize the embedding distortion. A $threshold$ value is set for that purpose where a frequency bin is selected for data hiding only if its energy is higher or equal to the threshold value. Data is embedded along a chosen LSB layer (CLSB) to $\Delta(m,k)_{dB}$. Where CLSB is the LSB layer lower-limit for hiding in a frequency bin. In our experiments, CLSB is chosen to be the 5^{th} LSB layer at minimum. The Δ value models the upper limit for data hiding in a selected bin. Δ value is set to impose a good quality on the stego-audio. The selection process of frequency bins done in the magnitude spectrum as well as the hiding locations are summarized in Figure 2. the details of the embedding process in a selected frequency bin is described in Figure 3. The value of $\Delta(m,k)_{dB}$ is set to $(|S_c(m,k)|)_{dB} - 13dB$. In doing so, we benefit from the fact that noise that is 13 dB below the original signal spectrum for all frequencies is inaudible [20]. Even though the frequency bins qualified for data hiding are selected in the magnitude spectrum, we believe that we will benefit also from mapping it to the phase spectrum for the following reasons:

1) As we partially alter selected frequency bins, only few bits in each selected frequency component are modified, which will give a smooth transition while preserving phase continuity.
2) When phase coding can be used, it gives better signal to noise ratio [20].
3) Opportunities to increase hiding capacity are worth to be explored

To embed in the phase spectrum, we map the exact selected frequency bins from the magnitude spectrum into the phase spectrum $\phi(m,k)$ and data is also embedded along CLSB layer to $\Delta(m,k)_{dB}$. Embedding data in phase spectrum is described as follows:

for $m = 1$ to M **do**
  for $n = 1$ to $N/2$ **do**
    $|\phi_s(m,n)| \leftarrow |\phi_c(m,n)|$
  end for
  for $k = F_{HDmin}$ to F_{HDmax} **do**
    if $10 * log_{10}(|S_c(m,k)|) \geq threshold_{dB}$ **then**
      if $\Delta(m,k)_{dB} \geq CLSB_{dB}$ **then**
        $|\phi_s(m,k)| \leftarrow |\phi_c(m,k)| + \delta(m,k)$
      end if
    end if
  end for
end for

Figure 1: Algorithm used to compute $|\phi_s(m,k)|$

In Figure 1, the value of $\delta(m,k)$ represents the modification in the phase value. A full description of the phase modification induced by embedded bits in a given frequency component is shown in Figure 3. The number of injected bits in a frequency component depends on its energy. In this manner, the embedding in a given frequency bin in $|\phi_s(m,k)| \leftarrow |\phi_c(m,k)| + \delta(m,k)$ is redefined as: $|\phi_c(m,k)| = (a_n 2^n + a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + ...a_2 2^2 + a_1 2^1 + a_0 2^0)$

Where $a_n = \{0,1\}$ and $\delta(m,k) = (d_i 2^i + d_{i-1}2^{i-1} + ... + d_0 2^0)$. The value of stego-phase can be simply calculated using:

$$|\phi_s(m,k)| = (a_n 2^n + a_{n-1}2^{n-2} + d_i 2^i + ...d_1 2^1 + d_0 2^0 + a_1 2^1 + a_0 2^0)$$

Finally, the new phase is multiplied with its magnitude to produce the stego-spectrum such as: $S_s(m,k) = |S_c(m,k)|e^{j\phi_s(m,k)}$. The inverse $iFFT$ transformation is applied on the segment to get the new stego-audio segment $s_s(m,n)$.

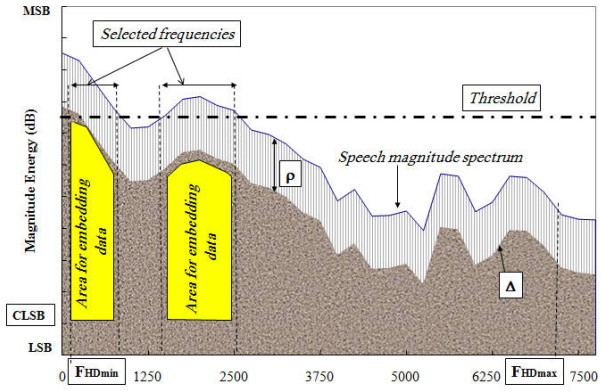


Figure 2: Spectral embedding area located in a frequency frame.

IV. HIDDEN DATA RETRIEVAL

To extract the hidden data from the phase spectrum, two main steps are followed: first, we locate the bins used for data embedding from the magnitude part $|S_s(m,k)|$. To do so, the parameters impacting the location of embedding in each selected bin such as $Threshold$, $\Delta(m,k)$, $CLSB$ are

 ISBN: 978-1-61208-376-6

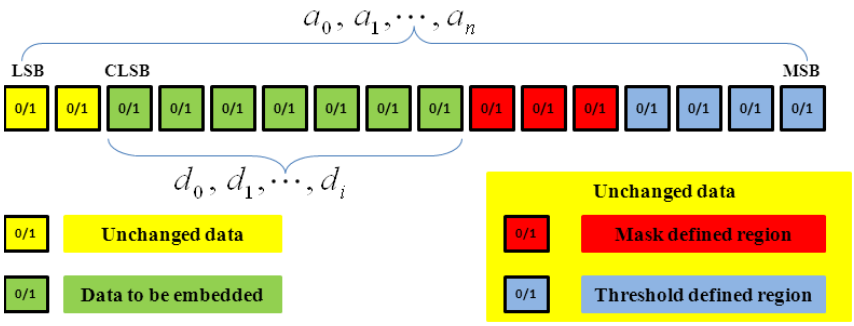


Figure 3: Embedding process in a selected frequency bin.

computed in the same way as done at the sender end. Second, we map the embedding locations found in the magnitude to the phase spectrum. Segments of the secret data are extracted and then reassembled as follows:

for $m = 1$ to M **do**
  for $n = 1$ to $N/2$ **do**
    $|\phi_s(m,n)| \leftarrow |\phi_c(m,n)|$
  end for
  for $k = F_{HDmin}$ to F_{HDmax} **do**
    if $10 * log_{10}(|S_c(m,k)|) \geq threshold_{dB}$ **then**
      if $\Delta(m,k)_{dB} \geq CLSB_{dB}$ **then**
        Extract $\delta(m,k)$ from $|\phi_s(m,k)|$
      end if
    end if
  end for
end for

Figure 4: Algorithm used to extract $\delta(m,k)$

V. Performance Evaluation

To evaluate the performance of the proposed algorithm, we conducted a comparative study between stego- and cover-audio signals. The study is based on (1) perceptual and (2) steganalysis undetectability.

A. Perceptual undetectability

In this section, we assess the quality of the stego-audio, when the hiding capacity is maximized. Tests have been conducted for magnitude [15] and the proposed phase configuration. Perceptual evaluation of speech quality ($PESQ$) measure defined in the ITU-T P862.2 standard combined with segmental SNR ($SegSNR_{dB}$) were utilized for the objective evaluation [21]. The hiding $Rate(Kbps)$ achieved is computed accordingly. Tests are carried out on a set of 100 audio waves, spoken in different languages by male and female speakers. Audio signals are 10s length each and sampled at 16 kHz and data is embedded within [0.25-7] kHz band with maximum hiding ratio of 23 kbps.

The PESQ test produces a value ranging from 4.5 to 1. A PESQ value of 4.5 means that the measured audio signal has no distortion: it is exactly the same as the original. A value of 1 indicates the severest degradation. The effectiveness of our algorithm is evaluated on audio frames sampled at 64. We set the algorithm parameters' value to maximize the hiding capacity while maintaining audio quality quality, i.e, Threshold = -20dB, ρ= 15dB, CLSB=1, F_{HDmin} and F_{HDmax} are set to 1 and 28 for 4ms frame length.

In our simulation, the distortion between stego and cover audio signals is calculated over several frames and by averaging the statistics, the overall measure is obtained. $SegSNR$ value for one modified audio frame of 4 ms is given by the following equation:

$$SegSNR_{dB}=10\log_{10}\left(\frac{\sum_{k=1}^{28}|S_c(m,k)|^2}{\sum_{k=1}^{28}|S_c(m,k)-S_s(m,k)|^2}\right) \quad (1)$$

The summation is performed over the signal per frame basis. To evaluate the results, the following criteria were used. First, the capability of embedding larger quantity of data (Kbps) is sought while naturalness of the stego-audio is retained. Second, the hidden data is fully recovered from the stego audio-signal.

TABLE I: PERFORMANCE EVALUATION AND COMPARISON

Hiding Method	SNR_{dB}	PESQ
[15]	26.86	4.32
Proposed	32.31	4.48

The values of SNR and PESQ registered in Table I are obtained from frames of 4 ms, hiding ration 23 Kpbs and 5^{th} LSB layer. They indicate clearly that stego-signals generated by the proposed phase embedding approach have experienced less distortion compared to [15]. Moreover, phase coding is robust to common linear signal manipulation such as: amplification, attenuation, filtering and resampling.

B. Comparative study by steganalysis

To further investigate our steganography algorithm performance, a comparative study by steganalysis is conducted based on a state-of-the-art reference audio steganalysis method [3]. The comparison is performed against our magnitude data hiding [15] and existing audio steganographic software: Steghide, S-Tools and Hide4PGP found respectively in [16]–[18]. The selected reference method was applied successfully in detecting the presence of hidden messages in high capacity LSBs-based steganography algorithms [3]. It is based on

 ISBN: 978-1-61208-376-6

extracting Mel-cepstrum coefficients (or features) from the second order derivative of audio signals. The features are then fed to a support vector machine (SVM) with RBF kernel [22] to distinguish between cover- and stego-audio signals.

For each studied steganography tool and algorithm, two datasets are produced: training and testing. Each dataset contains 270 stego and cover WAV audio signals of 10s length. All signals are sampled at 44.1 kHz and quantized at 16-bits. Each training and testing dataset contains 135 positive (stego) and 135 negative (cover) audio samples. We used online audio files from different types such as speech signals in different languages (i.e, English, Chinese, Japanese, French, and Arabic), and music (classic, jazz, rock, blues).

All stego-audio signals are generated by hiding data from different types: text, image, audio signals, video and executable files. To make a fair comparison between all assessed algorithms, the cover-signals were embedded with the same capacity of data. More precisely, S-Tools's (with hiding ratio of 50%) hiding capacity is used as a reference to embed the candidate steganographic algorithms and tools. The performance of each steganographic algorithm is measured through the levels by which the system can distinguish between stego and cover-audio signals (Table III). In order to analyze the obtained results, we first present the contingency table (see Table II).

TABLE II: THE CONTINGENCY TABLE

	Stego-signal	**Cover-signal**
Stego classified	True positives (tp)	False negatives (fn)
Cover classified	False positives (fp)	True negatives (tn)

The entries of the contingency table are described as follows:

- *tp*: stego-audio classified as stego-audio signal
- *tn*: cover-audio classified as cover-audio signal
- *fn*: stego-audio classified as cover-audio signal
- *fp*: cover-audio classified as stego-audio signal

In subsequent formulas, *all* represents the number of positive and negative audio signals. The value of the information reported in Table II is used to calculate the following measures:

$$Accuracy(AC) = \frac{tp + tn}{all} \quad (2)$$

The receiver operating characteristic (ROC) value is the fraction of true positive (TPR= true positive rate equivalent to Sensitivity) versus the fraction of false positive (FPR= false positive rate equivalent to 1-Specificity). Following the preparation of the training and testing datasets, we used the SVM library tool available at [23] to discriminate between cover- and stego-audio signals. The results of the comparative study are reported in Table III. The accuracy of each studied tool and method is measured by the values of AC and ROC.

In our second experimental work, we assess the performance evaluation of our algorithm and compare it to [15]–[18]. The values presented in Table III are the percentages of stego-audio signals correctly classified. Higher score values are interpreted as high-detection rate. Consequently, the proposed method show a significant improvement over the other, whereby, we were able to add a considerable accuracy to our steganographic algorithm against steganalysis attacks. The fact that the phase embedding scheme was able to perform better than the other algorithms, shows that the distortion amount resulting from embedding similar embedding ratios is much smaller.

TABLE III: OVERALL ACCURACY STEGANALYSIS RESULTS

Hiding methods	**AC**
Stools	0.725
Steghide	0.67
Hide4PGP	0.85
[15]	0.775
proposed	0.575

Further details on the behavior of each algorithm are represented in term of ROC curves in Figure 5. In each graph, a higher curve corresponds to more accurate detection rate while a lower curve corresponds to low accurate detection rate.

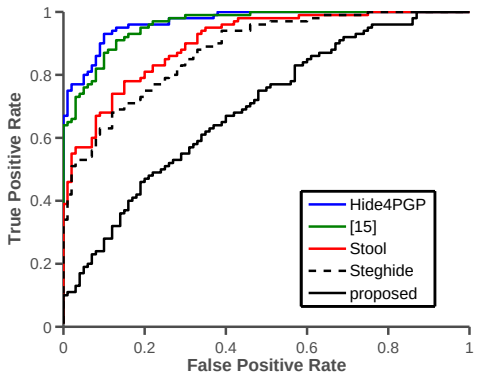


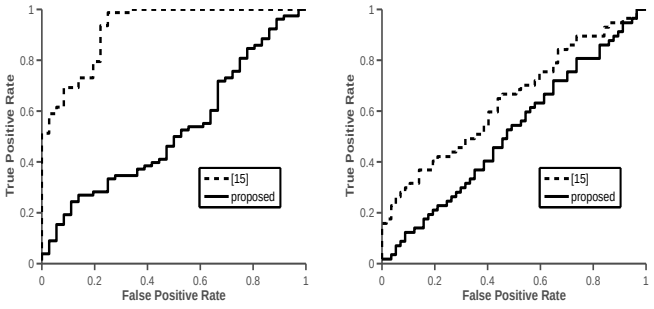
Figure 5: ROC curves for steganographic methods [15]–[18] and the proposed algorithm.

For the second experiment we further investigate the performance of our algorithm when the dataset contains only speech or music signals. The aim of this experiment is to put more emphasis on the behavior of the proposed algorithm when music audio-signals are used to convey hidden data versus those of speech audio-signals. We split the dataset into two sets A (130 speech signal) and B (130 music signal). Each set is further split to 65 stego- and 65 cover-signal to create a training and testing dataset for speech as well as for music. A set up similar to that described for experiment 1 was employed. The overall results in Table IV and Figure 6, show that our method performs better whether for speech- or music-signals. Our finding shows also that data-hiding in music Figure (6b) is less detectable than in speech-signals Figure (6a). In fact, the reference steganalysis method uses features extracted from high frequencies (lower in energy) while in our algorithm we target high energetic frequency components to embed data. In addition, the number of low-energy frequency components in music audio signals is smaller than that in speech signals.

 ISBN: 978-1-61208-376-6

TABLE IV: STEGANALYSIS RESULTS FOR DATA IN SPEECH AND IN MUSIC AUDIO SIGNALS

Hiding methods	Audio signal	AC	ROC
proposed	Music	0.504	0.502
	Speech	0.558	0.558
[15]	Music	0.6	0.598
	Speech	0.84	0.84



(a) (b)

Figure 6: ROC curves for [15] and the proposed method for data-hiding in speech (6a) versus music (6b) audio signals.

VI. CONCLUSION

In this paper, we presented a robust phase audio steganography. This work has a double aim. The first aim is to benefit from the fact that when phase coding can be used, it gives better signal to noise ratio. The second is to address the undetectability issue which is overlooked by most of the presented work in audio steganography. Perceptual and steganalysis study results reveal a great potential to hide large amounts of data, while ensuring their security and preserving the naturalness of the original signals. In the future, we plan to extend our work by investigating steganalysis of audio signals in codec domain.

REFERENCES

[1] F. Djebbar, B. Ayad, K. A. Meraim, and H. Hamam, "Comparative study of digital audio steganography techniques", EURASIP Journal on Audio, Speech, and Music Processing, Dec 2012, pp. 1-16.

[2] Avcibas, "Audio steganalysis with content independent distortion measures", IEEE Signal Process Letter, 2006, vol. 13, no. 2, pp. 92-95.

[3] Q. Liu, A. H. Sung, and M. Qiao, "Temporal derivative-based spectrum and mel-cepstrum audio steganalysis", IEEE Transactions on Information Forensics and Security, 2009, vol. 4, no. 3, pp. 359-368.

[4] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevit, "Digital image steganography: Survey and analysis of current methods", Signal Processing, Marsh 2010, vol 90, issue 3, pp. 727-752.

[5] F. Djebbar, K. Abed-Maraim, D. Guerchi, and H. Hamam, "Dynamic energy based text-in-speech spectrum hiding using speech masking properties", 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), May 2010, vol.2, pp. 422426.

[6] R. Balaji and G. Naveen, "Secure data transmission using video Steganography", IEEE International Conference on Electro/Information Technology (EIT), May 2011, pp. 1-5.

[7] M. Shirali-Shahreza and S. Shirali-Shahreza, "Persian/Arabic Unicode Text Steganography", SIAS Fourth International Conference on Information Assurance and Security, Sept. 2008, pp. 62-66.

[8] G. Handel Theodore and T. Maxwell Sandford II, "Hiding Data in the OSI Network Model", Information hiding: first international workshop, Cambridge, UK. Lecture Notes in Computer Science, 1996, vol. 1174, pp. 23-38.

[9] N. Cvejic and T. Seppanen, "Increasing Robustness of LSB Audio Steganography Using a Novel Embedding Method", Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), 2004, vol. 2, pp. 533537.

[10] M. A. Ahmed, M. L. M. Kiah, B. B. Zaidan, and A. A. Zaidan, "A novel embedding method to increase capacity and robustness of low-bit encoding audio steganography technique using noise gate software logic algorithm", Journal of Applied Sciences, 2010, vol. 10, pp. 59-64.

[11] X. Dong, M. Bocko, and Z. Ignjatovic, "Data hiding via phase manipulation of audio signals", IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings (ICASSP'04), vol. 5, pp. 377-380.

[12] K. Gopalan, "Audio steganography using bit modification", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'2003), vol. 2, pp. 421-424.

[13] S. Shirali-Shahreza and M. Shirali-Shahreza, "High capacity error free wavelet domain speech steganography", Proc. 33rd Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2008), Las Vegas, Nevada, USA, pp. 17291732.

[14] N. Cvejic and T. Seppanen, "A wavelet domain LSB insertion algorithm for high capacity audio steganography", Proc. 10th IEEE Digital Signal Processing Workshop and 2nd Signal Processing Education Workshop, Georgia, USA, October 2002, pp. 5355.

[15] F. Djebbar, H. Hamam, K. Abed-Meraim, and D. Guerchi, "Controlled distortion for high capacity data-in-speech spectrum steganography", International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IEEE-IIHMSP), ISBN: 978-0-7695-4222-5, 2010, pp. 212-215.

[16] Steghide, http://steghide.sourceforge.net/. Retrieved 28 Sept, 2014.

[17] Stools Version 4.0, http://info.umuc.edu/its/online_lab/ifsm459/s-tools4/. Retrieved 28 Sept, 2014.

[18] Hide4PGP, http://www.heinz-repp.onlinehome.de/Hide4PGP.htm. Retrieved 28 Sept, 2014.

[19] G. S. Kang, T. M. Moran, and D. A. Heide, "Hiding Information Under Speech", Naval Research Laboratory, http://handle.dtic.mil/100.2/ADA443638, Washington, 2005.

[20] B. Paillard, P. Mabilleau, S. Morissette, and J. Soumagne, "PERCEVAL: Perceptual Evaluation of the Quality of Audio Signals", journal of Audio Engeneering Society, 1992, vol. 40, pp. 21-31.

[21] Y. Hu and P. Loizou, "Evaluation of objective quality measures for speech enhancement", IEEE Transactions on Speech and Audio Processing, 16(1), 2008, pp. 229-238.

[22] N. Cristianini and J. Shawe-Taylor, "An introduction to Support Vector Machines", Cambridge University Press; 2000.

[23] http://www.csie.ntu.edu.tw/$\sim$cjlin/libsvm. Retrieved 28 Sept, 2014.

 ISBN: 978-1-61208-376-6

Current Issues in Cloud Computing Security and Management

Pedro Artur Figueiredo Vitti, Daniel Ricardo dos Santos,
Carlos Becker Westphall, Carla Merkle Westphall, Kleber Magno Maciel Vieira
Network and Management Laboratory - Department of Informatics and Statistics
Federal University of Santa Catarina - Florianopolis, Santa Catarina, Brazil
{pvitti, danielrs, westphal, carlamw, kleber}@inf.ufsc.br

***Abstract*—Cloud computing is becoming increasingly more popular and telecommunications companies perceive the cloud as an alternative to their service deployment models, one that brings them new possibilities. But to ensure the successful use of this new model there are security and management challenges that still need to be faced. There are numerous threats and vulnerabilities that become more and more important as the use of the cloud increases, as well as concerns with stored data and its availability, confidentiality and integrity. This situation creates the need for monitoring tools and services, which provide a way for administrators to define and evaluate security metrics for their systems. In this paper, we propose a cloud computing security monitoring tool based on our previous works on both security and management for cloud computing.**

Keywords–cloud computing; security management; monitoring; security metrics

I. INTRODUCTION

Cloud computing is a new way to provide computational resources over the Internet in a transparent and easy manner. According to the National Institute of Standards and Technology (NIST), it is a model for enabling on-demand network access to a shared pool of computational resources, comprised of three service models and four deployment models [1].

These service models are: Software as a Service (SaaS), in which the service provided to the user is in the form of an application that runs on a cloud infrastructure; Platform as a Service (PaaS), in which the user can deploy its own applications in the provider's infrastructure; and Infrastructure as a Service (IaaS), in which the user has access to the computational resources themselves, in the form of virtual machines, storage, networks and others.

The deployment models are the private, community, public and hybrid cloud, and refer to the location of the cloud infrastructure, who has access to it and who is responsible for its management. The most used models are the public cloud, when the infrastructure is run by an organization and provisioned to be used by the public; and the private cloud, when an organization provisions its own infrastructure to be used by their business units.

In an era where telecommunication providers face ever greater competition and technology evolution, the basic features of cloud computing such as virtualization, multi-tenancy and ubiquitous access provide a viable solution to their service provisioning problems.

Telecoms are now using their own private clouds, or sometimes public clouds, to host their services and enjoy the benefits of this new model. With a multi-tenant cloud they can support an increasing number of subscribers and maintain the Quality of Experience of their services even when dealing with high demand. The use of the cloud also helps these companies transition from a product based business model to a service based one.

The main advantages of cloud computing are the reduction of IT costs and increased flexibility, scalability and the possibility to pay only for the used resources. The users of the cloud range from individuals to large government or commercial organizations, and each one has their own concerns and expectations about it.

Among these concerns, security and privacy are the biggest ones [2]. This comes from the fact that the data that belongs to users and organizations may no longer be under their absolute control, being now stored in third party locations and subject to their security policies, in the case of public clouds.

But even in private clouds, the most common case in telecom companies, there are new security challenges, such as providing access to an ever growing number of users while maintaining efficient and well monitored access control.

It becomes necessary to characterize what are the new risks associated with the cloud and what other risks become more critical. These risks must be evaluated and mitigated before the transition to the cloud.

It is already possible to find in the literature a lot of work being done in the security aspects of Cloud Computing, describing its challenges and vulnerabilities and even proposing some solutions [3].

In the rest of this paper, we provide some background in security concerns in cloud computing, briefly describe a previous implementation of a monitoring tool for the cloud, show how security information can be summarized and treated under a management perspective in an Service Level Agreement (SLA) and then propose a system for monitoring security information in the cloud.

In Section II, some works, related to security in cloud computing environments, are cited. In Section III, currently existing concerns in cloud computing security area are presented. In Section IV, an architecture for monitoring clouds is described. In Sections V and VI, safety concerns with SLA, and the definition of entities, components, metrics and, actions of security monitoring cloud computing are shown. Section VII shows the case study. In Section VIII, lessons learned from this work are described. Finally, in Section IX, a conclusion is presented and some future work proposals are made.

II. RELATED WORK

Uriarte and Westphall [4] proposed a monitoring architecture devised for private Cloud that focuses on providing data

 ISBN: 978-1-61208-376-6

analytics capabilities to a monitoring system and that considers the knowledge requirements of autonomic systems. While, argue that in the development of an analytical monitoring system for public Clouds, security, privacy and different policies need to be considered, their proposal does not consider specific security metrics and Sec-SLAS.

Fernades et al. [5] surveys the works on cloud security issues. Their work addresses several key topics, namely vulnerabilities, threats, and attacks, and proposes a taxonomy for their classification. Their work, however, does not consider metrics monitoring or any implementation details.

CSA [6] has identified the top nine cloud computing threats. The report shows a consensus among industry experts, focusing on threats specifically related to the distributed nature of cloud computing environments. Despite identifying, describing and analyzing these threats, their work does not consider the monitoring of security metrics related to the identified threats.

Murat et al. [7] proposed a cloud network security monitoring and response system, which is based on flow measurements and implements an algorithm that detects and responds to network anomalies inside a cloud infrastructure. Their proposal however does not take into account security metrics and Sec-SLAs, instead it generates and monitors profiles of network traffic to detect for anomalies, hence it is limited in the scope of security issues it can monitor.

III. Security Concerns in Cloud Computing

A. Technologies

A lot of different technologies are necessary to create and manage a cloud environment, according to the kind of service that this cloud will provide. Cloud computing relies heavily on virtualization and network infrastructure to support its elasticity. Technologies such as Web Services, Service Oriented Architecture (SOA), Representational State Transfer (REST) and Application Programming Interfaces (API) are employed to provide users with access to their cloud resources. Each of these technologies presents some kind of known vulnerability and possible new exploits in the cloud [8].

B. Challenges, Threats and Vulnerabilities

The usual three basic issues of security: availability, integrity and confidentiality are still fundamental in the cloud and remain a big challenge in this scenario. Each sector has its main concerns when it comes to the cloud. Industry services are mostly worried about availability, so that they keep providing services even during peaks of access, while academia may be more concerned with integrity and individual users usually care about the confidentiality of their data. But every security aspect must be considered together to achieve security as a whole in this scenario. Because of the multi-tenant characteristic of cloud computing, one single vulnerable service in a virtual machine may lead to the exploitation of many services hosted in the same physical machine. Also, virtualization has an inherent security threat that a user may escape its confined environment and gain access to the physical machine resources or to other virtual machines. This requires complex attacks, but is possible.

Web applications and web services have a long history of security vulnerabilities, and if not well implemented they are susceptible to a lot of easily deployed and very well-known attacks such as SQL injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF) and session hijacking.

Cryptography is the most important technology to provide data security in the cloud, but problematic implementations and weak proprietary algorithms have been known problems for a long time and are still exploited.

Another important topic in cloud security is Identity and Access Management, because now data owners and data providers are not in the same trusted domain. New mechanisms for authentication and authorization that consider cloud-specific aspects are needed and are being actively researched [9].

The main security management issues of a Cloud Service Provider (CSP) are: availability management, access control management, vulnerability management, patch and configuration management, countermeasures, and cloud usage and access monitoring [10].

To remain effective in this new paradigm, some security tools have to be adapted, such as Intrusion Detection Systems (IDS), which are critical to monitor and prevent incidents in the cloud. Because of its distributed nature, the cloud is an easy target for an intruder trying to use its abundant resources maliciously, and because of this nature, the IDS also has to be distributed, to be able to monitor each node [11].

C. Attacks

While the cloud serves many legitimate users, it may also host malicious users and services, such as spam networks, botnets and malware distribution channels. Cloud providers must be aware of those problems and implement the necessary countermeasures.

Besides that, Distributed Denial of Service (DDoS) attacks can have a much broader impact on the cloud, since now many services may be hosted in the same machine. When an attacker focuses on one service it may affect many others that have no relation with the main target. DDoS is a problem that is still not very well handled. On the other hand, since the cloud provides greater scalability and may allocate resources almost instantaneously it becomes more resilient to denial of service, but it comes with a cost to the users.

D. Data Security

The security and privacy of the data stored in the cloud is, perhaps, the most important challenge in cloud security. To maintain data security a provider must include, at least: an encryption schema, an access control system, and a backup plan [12].

However, data encryption can be a hindrance in the cloud because of the current impossibility to efficiently process or query over encrypted data [2]. There is active research in these areas, with techniques such as Searchable Encryption and Fully Homomorphic Encryption, but their applications are still limited and they cannot yet be used in large scale environments.

When moving to the cloud it is important that a prospective customer knows to what risks its data are being exposed. Some of the key points a user must consider in this migration are [13]: The cloud administrators will have privileged access to user data, and possibly bypass access controls; The provider must comply to legal requirements, depending on the kind of data the user intends to store; The location of the user's data may now be unknown to them; How the data of one user are kept separate from others; The provider must have a capacity to restore a system and recover its data through backup and replication; The provider must formally ensure full support in

 ISBN: 978-1-61208-376-6

the case of an investigation over inappropriate activities; and The data must be in a standardized format and be available to the user even in the case the provider goes out of business.

E. Legal Compliance

Legal compliance is fundamental when dealing with cloud computing. In the cloud world, it is possible that data cross many jurisdiction borders and have to be treated in compliance to many different laws and regulations. This is one of the reasons why security plays such an important role in cloud adoption and development, especially for the CSPs.

To achieve compliance both providers and users must be held responsible for how data is collected, stored and transmitted, especially sensitive data, such as Personally Identifiable Information (PII).

Among the most important tools to ensure legal compliance are external audits and security certifications.

F. Telecommunications

The deployment and provisioning of telecommunication services becomes easier in the cloud, and it empowers telecom providers with greater scalability and flexibility. Those advantages, however, come with the cost of new security challenges.

Security plays such a vital role in telecommunications that many telecommunication networks are built from the ground-up with security requirements in mind. This, however, is not true for many Internet protocols. When transitioning to the cloud, telecom providers must be aware that their services are being deployed in a different scenario, one that has to be well understood before this transition is considered.

Availability, for instance, is critical to the telecom business and if services are being deployed in a public cloud without a proper SLA, server downtime will cause a lot of trouble. Confidentiality is also fundamental, since telecoms collect and store a lot of data from their clients, from personal data to information about their communications.

IV. Cloud Monitoring

The provisioning of cloud services represents a challenge to service monitoring. It requires complex procedures to be well accomplished, which leads to the development of new management tools. Our team has previously proposed and implemented an open-source cloud monitoring architecture and tool called the Private Cloud Monitoring System (PCMONS) [14].

The architecture of the system is divided in three layers (see Figure 1):

- Infrastructure - Consists of basic facilities, services and installations and available software, such as operating systems and hypervisors;
- Integration - Responsible for abstracting the infrastructure details for the view layer; and
- View - The interface through which information is analyzed.

The main components of the architecture are (see Figure 1):

- Node information gatherer: Gathers local information on a node;
- VM monitor - Injects scripts into the virtual machine (VM) that send data to the monitoring system;
- Configuration Generator - Generates the configuration files for the tools in the view layer;
- Monitoring tool server - Receives data form different resources and take actions such as storing it;
- Database - Stores data needed by the Configuration Generator and the Monitoring Data Integrator.

V. Security Concerns in SLA

Security is not only a matter of preventing attacks and protecting data, it also has to be considered in a management perspective. Providers must have ways to ensure their clients that their data is safe and must do so by monitoring and enhancing security metrics.

A SLA formally defines the level of service a provider must guarantee. SLAs are a fundamental part of network management, and are also applied in cloud computing. They are defined in terms of metrics that must be monitored to ensure that the desired levels of service are reached.

SLAs may also be used in the definition, monitoring and evaluation of security metrics, in the form of Security SLAs, or Sec-SLAs [15]. In this case, the SLA considers security service levels.

To accomplish this, the first step is to define a set of security metrics, which in itself is not easy. Though there is not a definitive set of security metrics that is considered relevant in every case, researchers tend to use or adapt concepts gathered from international standards such as ISO 27002. Some issues that are usually considered are cryptography, packet filtering, redundancy, availability, and backup.

VI. Cloud Security Monitoring

Security monitoring is inherently hard, because the agent-manager approach normally used in the monitoring of other kinds of SLA, does not fit easily to every security characteristic [15].

Cloud computing has been evolving for many years and so, only now we are able to have a broader view of what exactly it is and hence what are its security requirements, based on recent definitions and publications.

With this new perspective it is now possible to define good security metrics that can be used to provide a clear view of the level of security being employed in a CSP and its virtual machines.

We now propose an extension to the PCMONS architecture and tool to enable security monitoring for cloud computing. We also present the security metrics which we consider adequate to be monitored in a cloud infrastructure and which provide a good picture of security as a whole in this environment.

The tool uses data and logs gathered from security software available in the monitored systems, such as IDSs, anti-malware software, file system integrity verification software, backup software and web application firewalls, and presents these data to the cloud administrators.

Besides providing to administrators reliable metrics and information about the security of their systems, this monitoring architecture can also be used in the auditing and outsourcing of security services.

The main components of the proposal can be seen in Figure 1 and are described below.

A. Entities

The entities involved in the definition, configuration and administration of the security SLAs and metrics are:

- Cloud users - The users of the cloud infrastructure. They negotiate the SLAs with the CSP and expect

 ISBN: 978-1-61208-376-6

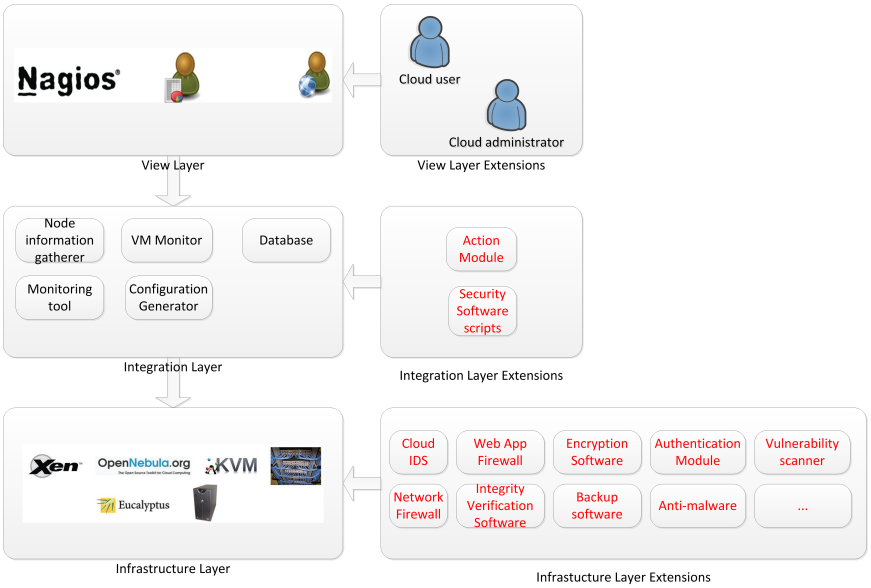


Figure 1. Three Layers Monitoring Architecture

them to be accomplished;
- Cloud administrators - The administrators of the CSP. Their role is to monitor the cloud infrastructure; and
- Security applications - The applications which produce the security information that will be gathered.

The two first entities were a part of the previous PCMONS, while the third one was inserted in our extension.

B. Components

Since PCMONS is modular and extensible, the components used in the new architecture are the same already available, but with extensions that allow the monitoring of security metrics.

The extensions are new scripts to gather the security data from the many sources needed and an extension to the visualization tool to show this data.

C. Metrics

As mentioned in Section IV, the definition of security metrics is not an easy task, and it becomes even harder in the cloud.

Here, we present the basic metrics we intended to monitor. These metrics were chosen because we consider they cover a great part of what was considered critical in a cloud provider, based on the survey presented in Section II.

We divided the set of metrics into subsets related to each security aspect that will be treated. There are four subsets of metrics. The first three are related to each individual virtual machine. Data Security Metrics, Access Control Metrics and Server Security Metrics are shown in Table I, Table II, and Table III, respectively.

D. Actions

We decided to introduce a new module to take actions based on the monitored metrics and possible violations to the Sec-SLA. As an example, if a virtual machine has had a huge number of failed access attempts in the last hours we may want to lock any further access to it and communicate the possible issue to the administrator of that machine. Also, if malware was detected on a machine we may want to shut it down to prevent it from infecting other VMs in the same physical machine. These actions will be predefined scripts available to cloud administrators and may be enabled or disabled by them at any time.

VII. Case Study

We have implemented the metrics presented in Tables I-III and gathered the data generated in a case study. The implementation of the data gathering scripts was done in Python and the data shown in the Nagios interface.

Our infrastructure consisted of two physical servers, one hosting the OpenNebula cloud platform and another hosting the virtual machine instances.

Several virtual machines running the Ubuntu operating system and the security software needed to provide the security metrics were instantiated. The following software were used to gather the security information: dm-crypt (encryption), rsync (backup), tripwire (filesystem integrity), ssh (remote access), clamAV (anti-malware), tiger (vulnerability assessment) and uptime (availability).

The VMs were automatically attacked by brute force login attempts and malware being downloaded and executed, as well as access attempts to ports blocked by the firewall. During the tests there were also simulations of regular usage, encompassing valid accesses and simple user tasks performed on the machines, such as creating and deleting files. The malware scans, vulnerability scans, integrity checks and backups were performed as scheduled tasks on the operating system using latest versions of Linux Malware Detect [16], OpenVAS [17], AFICK [18] and, Amanda [19] respectively. We did not stress the environment to test for scalability issues because it had already been done with the previous versions of PCMONS.

Figure 2 shows an example of an early snapshot of the monitoring environment. It represents how the metrics are shown in Nagios and it is possible to see the vision that a network administrator has of a single machine. The metrics HTTP_CONNECTIONS, LOAD, PING, RAM and SSH are from the previous version of PCMONS and are not strictly related to security, but they are show combined.

 ISBN: 978-1-61208-376-6

TABLE I. DATA SECURITY METRICS

Metric	Description
Encrypted Data?	Indicates whether the data stored in the VM is encrypted
Encryption Algorithm	The algorithm used in the encryption/decryption process
Last backup	The date and time when the last backup was performed
Last integrity check	The date and time when the last file system integrity check was performed

TABLE II. ACCESS CONTROL METRICS

Metric	Description
Valid Accesses	The number of valid access attempts in the last 24 hours
Failed access attempts	The number of failed access attempts in the last 24 hours
Password change interval	The frequency with which users must change passwords in the VM's operating system

TABLE III. SERVER SECURITY METRICS

Metric	Description
Malware	Number of malware detected in the last anti-malware scan
Last malware scan	The date and time of the last malware scan in the VM
Vulnerabilities	Number of vulnerabilities found in the last scan
Last vulnerability scan	The date and time of the last vulnerability scan in the VM
Availability	Percentage of the time in which the VM is online

It is important to notice that the accuracy of the obtained information depends on the security software being monitored. Our solution aggregates these data to present it in a way that is more clear and easy to monitor. The tool helps network and security administrator perceive violations to Sec-SLAs and actively respond to threats.

In this case study, considering the automatic attacks previously described, the most violated metrics were the failed access attempts and the anti-malware events, as well as availability, because of malware that would cause denial of service.

Since we obtained a high number of violations in an environment that was supposed to be under constant attack, it suggests that the chosen metrics are good indicators of overall security for the virtual machines.

VIII. KEY LESSONS LEARNED

A. Background

Monitoring and managing security aspects remains a challenge that has to be faced to enable the full potential of the cloud and only now, with a recent agreed upon definition of exactly what is cloud computing, this can be achieved. The major piece of technology used to provide security in the cloud is cryptography.

Data leakage and data loss are possibly the greatest concerns of cloud users. If the CSP acts unfaithfully the users may not even become aware of incidents that compromise their data. There must be ways to verify data integrity, so that users are certain their data were not corrupted. Backup and recovery are also fundamental tools to ensure the availability of customer data.

The greatest challenge to security monitoring in a cloud environment is the fact that the cloud provides services on demand, creating a highly dynamic and flexible system to which the metrics have to be adapted.

SLAs are fundamental to provide customers with the needed guarantees that the service they are hiring will be adequately provided, their machines will be secure and their data will be securely stored, transmitted and processed.

Security metrics and a more quantitative approach to security, in both the definition of requirements and their monitoring, remain an important open research topic.

There are other important security metrics that are related to the security processes of the CSP, such as employee training, physical security and contingency plans. These were not taken into account in this work because they cannot be automatically gathered and monitored.

B. Design and Implementation

The design of a software project and related architectural decisions may consume a great time before the implementation is even started. Building an extension over a previous architecture, as was our case, may greatly reduce this time.

Nevertheless, many important decisions have to be made to achieve a final functioning software. The major decisions in this case were related to the security metrics and the software used to provide the necessary security data.

As already stated in this paper, defining and quantifying security is no easy task, therefore it was the most time consuming aspect of the project. Trying to come up with a simple set of metrics that represent the state of security of a whole cloud not only seems, but definitely is a daunting task.

Something that became clear with this implementation is that no single set of metrics can embrace every security need, and so to define the metrics we based our approach on the common security issues described in the literature, as well as issues that are consistently cited as the most critical by industry users. It is also important to note that the definition and monitoring of metrics must be flexible enough to accommodate different potential uses of the software.

After defining what is going to be measured it is necessary

 ISBN: 978-1-61208-376-6

oneadmin i-322 stratus	AVAILABILITY	OK	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	99.93%
	CIPHER	OK	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	AES
	SSH_VALID	OK	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	25
	IS_ENCRYPTED	OK	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	yes
	LAST_BACKUP	OK	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	2014-08-14 18:30:48
	LAST_INTEGRITY	OK	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	2014-08-14 10:23:50
	LAST_MALWARE_SCAN	OK	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	2014-08-14 10:02:42
	VULNERABILITIES	CRITICAL	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	111
	MALWARE_FOUND	CRITICAL	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	5
	PASSWORD_INTERVAL	OK	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	180 days
	SSH_FAIL	WARNING	2014-08-14 15:58:51	4d 1h 16m 40s	1/4	545

Figure 2. Nagios simplified interface of the monitored cloud services

to focus on how to do it. The idea of analyzing logs to obtain security data is classical in information security and it seemed like a natural approach to our challenge.

To read, parse and present the data we chose to use the Python programming language because it already formed the base of PCMONS and it fits very well these kinds of tasks.

An important aspect of the proposed solution is its modularity. Because of this feature we were able to introduce the new metrics and adapt it to our needs without changing anything that was already done in terms of basic monitoring. We believe the same can be achieved anytime it becomes necessary to adapt the software to new particular monitoring needs.

Modularity and extensibility are necessary approaches when you deal with such dynamic and highly scalable environments, because you have to be certain that you will be able to adjust the software to your future needs, which may be very different from current ones. The most challenging metrics in terms of implementation were those that gathered data from non-standard security software, such as tripwire, because we had to understand the data they generated to interface them with PCMONS. The analysis of our results shows that PCMONS was able to comply with our defined set of metrics, since their implementation relied on established security software, and that the definition and implementation of new metrics may be done in the future without the need for a great architectural redesign.

C. Testing Environment

Setting up a reliable testing environment was also extremely important to the success of the project. Setting up a private cloud is often advertised as being simple and convenient, but that is not always true when we have to deal with specificities of architectures, operating systems and hypervisors.

Our private cloud has been evolving for some years and through the realization of other projects we were able to gather experience on deploying, managing and monitoring it, which allowed us to choose tools we already knew would work well together.

Since the whole cloud infrastructure is built upon a piece of software, it is important to know that it is stable, reliable, well documented and provides available support. Our choice for the OpenNebula platform came from previous experience with it and its widespread use by many big players in the industry, such as Telefonica, Akamai and IBM.

An important feature of this extension of PCMONS is that it can run over Eucalyptus, OpenNebula and OpenStack, monitoring virtual machines in every platform. The support for different cloud platforms reflects the evolution of cloud tools and a greater effort being made in terms of standardization, interoperability and portability, all of which are big issues in cloud computing.

The use of scripting languages in the development process, such as Python and Bash Script allowed us to define the metrics, implement and test them on the fly on the testing environment, without needing to stop services, compile software, test it, compile it again and so on. This approach required less intensive use of the testing environment during development and accelerated the whole process.

IX. Conclusion and Future Work

This paper described a few of our previous works in the field of Cloud Computing and how to bring them all together in order to develop a cloud security monitoring architecture.

The use of cloud computing is a great option for telecommunications companies that want to reduce OPEX and CAPEX costs and still improve their service provisioning. Security, nevertheless, must be accurately planned and monitored to ensure that the transition to the cloud runs smoothly.

The paper described the design and implementation of a cloud security monitoring tool, and how it can gather data from many security sources inside VMs and the network in which the physical machines are to give administrators a clear view of the security of their systems and allow Cloud Service Providers to give users guarantees about the security of their machines and data.

Currently, there are not many solutions to cloud security monitoring, and this paper shows it is possible to build such a system based on previous work.

As future work, we can point to the definition and implementation of new metrics and a better integration with existing Security SLAs, planning to include a new module to treat possible actions to be taken in response to metric violations, such as locking a virtual machine or shutting it down.

Also, it would be important to study the integration of the

 ISBN: 978-1-61208-376-6

security monitoring model with other active research fields in cloud security, such as Identity and Access Management and Intrusion Detection Systems.

ACKNOWLEDGEMENT

We would like to thank Salvatore Loreto, Saverio Niccolini and Vijay Gurbani for their prior review and for their help in improving the paper.

REFERENCES

[1] P. Mell and T. Grance, The nist definition of cloud computing. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (2011) [retrieved: Sept, 2014]

[2] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," Internet Computing, IEEE, vol. 16, no. 1, jan.-feb. 2012, pp. 69 –73.

[3] F. Shaikh and S. Haider, "Security threats in cloud computing," in Internet Technology and Secured Transactions (ICITST), 2011 International Conference for, 2011, pp. 214–219.

[4] R. B. Uriarte and C. B. Westphall, "Panoptes: A monitoring architecture and framework for supporting autonomic clouds," in Network Operations and Management Symposium (NOMS), 2014 IEEE. IEEE, 2014, pp. 1–5.

[5] D. Fernandes, L. Soares, J. Gomes, M. Freire, and P. Incio, "Security issues in cloud environments: a survey," International Journal of Information Security, vol. 13, no. 2, 2014, pp. 113–170. [Online]. Available: http://dx.doi.org/10.1007/s10207-013-0208-7 [retrieved: Sept, 2014]

[6] T. T. W. Group et al., "The notorious nine: cloud computing top threats in 2013," Cloud Security Alliance, 2013.

[7] M. Mukhtarov, N. Miloslavskaya, and A. Tolstoy, "Cloud network security monitoring and response system," vol. 8, no. Special Issue on Cloud Computing and Services. sai: itssa.0008.2012.020 ITSSA, 2012, pp. 71–83.

[8] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," Security Privacy, IEEE, vol. 9, no. 2, march-april 2011, pp. 50 –57.

[9] X. Tan and B. Ai, "The issues of cloud computing security in high-speed railway," in Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on, vol. 8, 2011, pp. 4358–4363.

[10] F. Sabahi, "Cloud computing security threats and responses," in Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, 2011, pp. 245–249.

[11] K. Vieira, A. Schulter, C. Westphall, and C. Westphall, "Intrusion detection for grid and cloud computing," IT Professional, vol. 12, no. 4, 2010, pp. 38–43.

[12] L. Kaufman, "Data security in the world of cloud computing," Security Privacy, IEEE, vol. 7, no. 4, 2009, pp. 61–64.

[13] S. Chaves, C. Westphall, C. Westphall, and G. Geronimo, "Customer security concerns in cloud computing," in ICN 2011, The Tenth International Conference on Networks, 2011, pp. 7–11.

[14] S. De Chaves, R. Uriarte, and C. Westphall, "Toward an architecture for monitoring private clouds," Communications Magazine, IEEE, vol. 49, no. 12, 2011, pp. 130–137.

[15] S. de Chaves, C. Westphall, and F. Lamin, "Sla perspective in security management for cloud computing," in Networking and Services (ICNS), 2010 Sixth International Conference on, 2010, pp. 212–217.

[16] R. M. Ryan MacDonald, Linux malware detect. [Online]. Available: https://www.rfxn.com/projects/linux-malware-detect/ (2014) [retrieved: Sept, 2014]

[17] R. Deraison, Open vulnerability assessment system. [Online]. Available: http://http://www.openvas.org/ (2014) [retrieved: Sept, 2014]

[18] E. Gerbier, Another file integrity checker. [Online]. Available: http://afick.sourceforge.net/ (2014) [retrieved: Sept, 2014]

[19] J. da Silva, Advanced maryland automatic network disk archiver. [Online]. Available: http://http://www.amanda.org/ (2014) [retrieved: Sept, 2014]

[20] D. dos Santos, C. Merkle Westphall, and C. Becker Westphall, "A dynamic risk-based access control architecture for cloud computing," in Network Operations and Management Symposium (NOMS), 2014 IEEE, May 2014, pp. 1–9.

[21] P. Silva, C. Westphall, C. Westphall, M. Mattos, and D. Santos, "An architecture for risk analysis in cloud," in ICNS 2014, The Tenth International Conference on Networking and Services, 2014, pp. 29–33.

 ISBN: 978-1-61208-376-6

N-Gram-Based User Behavioral Model for Continuous User Authentication

Leslie Milton, Bryan Robbins, and Atif Memon,
Department of Computer Science,
University of Maryland, College Park, MD, USA
{lmilton,brobbins,atif}@cs.umd.edu

***Abstract*—We posit that each of us is unique in our use of computer systems. It is this uniqueness that we leverage in this paper to "continuously authenticate users" while they use web software. We build an n-gram model of each user's interactions with software. This probabilistic model essentially captures the sequences and sub-sequences of user actions, their orderings, and temporal relationships that make them unique. We therefore have a model of how each user typically behaves. We then continuously monitor each user during software operation; large deviations from "normal behavior" can indicate malicious behavior. We have implemented our approach in a system called *Intruder Detector* (ID) that models user actions as embodied in the web logs generated in response to the actions. Our experiments on a large fielded system with web logs of approximately 320 users show that (1) our model is indeed able to discriminate between different user types and (2) we are able to successfully identify deviations from normal behavior.**

Keywords–behavioral modeling; continuous authentication; software security; n-grams.

I. INTRODUCTION

The idea of continuous user authentication (CUA) is not new. The basic premise of CUA is that conventional user authentication, usually performed during the initial login session, is insufficient. In conventional authentication, users are not asked to verify their identity during their session, leaving the computer system vulnerable to malicious or unintended use while the user is logged-in. CUA techniques, on the contrary, monitor, verify, and authenticate users during their entire session. Functionally, CUA may be considered as one example of an intrusion detection system (IDS). CUA can be used to check the state of the system by continuously montioring user activity and comparing this activity with stored usage profiles to alert and/or de-authenticate the user when an intrusion is detected or suspected. IDS takes this one step further by adding a second line of defense to pinpoint misuse and initiate proper response [1]. Several studies have used biometrics to continuously authenticate users by the use of cognitive fingerprints, eye scans, color of user's clothing, and face tracking [2][3][4]. However, many of these techniques require additional hardware and cost to operate efficiently. Behavioral modeling addresses these limitations by monitoring how users interact with the system. Evaluating mouse movement, how a user searches for and selects information, and the habitual typing rhythm of users are traits used to continuously observe a user's behavior [5][6]. Although these approaches do not require special hardware, most of them do require the installation of specialized monitoring software.

In this paper, we posit we can obtain a unique behavioral footprint indicating patterns of use for a specific user or group of users of a particular web-based software using web log analysis. It is this footprint that we leverage to "continuously" authenticate the user. We can construct models of any targeted group of sessions based on n-grams. n-gram models have performed somewhat surprisingly well in the domain of language modeling, where researchers have found that a history of only one to two events is necessary to obtain optimal predictive capabilities [7]. An n-gram model captures all sequences and subsequences of a fixed length, N, from previously observed user input, which allows prediction and evaluation of future behavior based on frequencies. If we assume that the event sequences carried out by users of a software system are analogous to natural language, we would expect to find similar predictive power in n-gram models of software event sequences as well. To test the validity of our approach, we seek to answer the following research questions:

1) *Can we build discriminating user models to determine user types?*
2) *Can the model recognize various legitimate users who are operating in the same user session?*
3) *Can usage profiles be used to identify outliers in the user's behavior?*

Our approach is different from that employed by [3] because they focus on device usage and delays, mouse tracking, word usage, etc. Our approach also differs from various biometric approaches [8][4][9][10][11]. We instead focus on a particular layer of the software. With our approach, we evaluate web log data generated by users of a web-based system. The web logs are used to build unique profiles based on a user's role within the system. There is no need for additional hardware since we are using information that is automatically generated by the system. This process provides a continuous verification technique with no interaction from the user which not only improves security but enhances usability of the system.

We feel that our new approach should be used together with existing authentication mechanisms (e.g., passwords) to provide an increased level of security demanded by today's computing environment. ID is just one more tool in the security expert's toolbox.

This work makes the following contributions:

1) We use n-grams to model the behavior of users while they interact with web-based software.
2) We show how keywords are abstracted from web logs to develop user footprints.
3) We develop a continuous user authentication technique with the ability to categorize user sessions into a pre-defined set of roles or potentially finer-grained user profiles.

 ISBN: 978-1-61208-376-6

The rest of the paper is organized as follows: Section II provides details of related work. Section III defines the basis of continuous user authentication and how we model this activity using n-grams. Section IV describes our pilot study with experimental results. The conclusion and future work are discussed in Section V.

II. Background & Related Work

User authentication serves as a prevention-based method to protect malicious access of systems. However, if a malicious user is able to successfully pass the authentication step, there should be a transparent method to detect their behavior. For this reason, CUA is used as a second line of defense to check the state of the system by continuously monitoring user activity and simultaneously compares this activity with stored usage profiles to alert and/or de-authenticate the user. Finally, the IDS takes over and performs misuse and anomaly detection. The former identifies patterns of known attacks and the latter detects intrusions by determining whether there is some deviation from stored normal usage patterns.

Various research studies have explored the use of authentication. Kaminsky *et al*. address challenges for user authentication in a global file system [12]. This approach uses an authentication server to identify users based on local information. Researchers of cloud computing security methods have developed implicit authentication to identify a user's past behavior data to authenticate mobile devices [13]. These two studies have one major limitation worth noting. They lack the ability to continuously monitor user behavior for anomaly detection.

The realm of CUA has be extensively evaluated with the use of biometrics to verify user identity through the use of unique behavioral and/or phyical traits. A study by the Defense Advanced Research Projects Agency (DARPA) uses a combination of metrics that include eye scans and keystrokes to evaluate how the user searches and selects information [3]. In addition, a number of research studies concerning CUA use one or more hard and soft biometric traits to continuously authenticate a user. Niinuma *et al*. propose a CUA framework to automatically register the color of users' clothing and their face as soft biometric traits [4][2]. Results from this study show that the system is able to successfully authenticate the user with high tolerance to the user's posture. Limitations to these studies exist because of the additional hardware that is needed to implement this technique which can become costly if an entire organization uses this feature to authenticate users.

Altinok *et al*. propose a continuous biometric authentication system that provides an estimate of authentication certainty at any given time, even in the absence of any biometric data [10]. However, as the authentication uncertainty increases over time, system usability decreases. In a similar study, Kang *et al*. introduce temporal integration of biometrics and behavioral features to continuously authenticate users [14]. Similar to the previous biometric studies, additional hardware is needed.

A face tracking system that uses color and edge information has been used to compute behavioral features. Shen *et al*. use mouse dynamics when implementing continuous user authentication [5]. This technique is used to observe behavioral features in mouse operations to detect malicious users. However, there are some existing limitations with this emerging approach. Behavioral variability occurs because of human or environmental factors. Such changes could possibly identify the correct user as an impostor.

Our study extends beyond the aforementioned research studies in that: 1) Instead of using traditional biometric traits, we explore the possibility of using web log information that is automatically generated by web applications; 2) Our approach, integrated into a prototype tool, uses a novel and simple n-gram language model to capture user behavior; 3) Our experiments are based on data from users of a government system who are completing day-to-day tasks.

III. Continuous User Authentication

Our approach to CUA involves building n-gram models of user activity by observing sequences of user interaction with a web-based system. Once a model is constructed, we leverage it for the classification of incoming event sequences. However, the models are not without problems. While we have been able to address some challenges with tool support, other challenges remain. Below we formally define n-gram models, then present our approach to dealing with fundamental risks of using this type of model. Finally, we present algorithms for applying this model to the CUA domain.

A. n-gram Models

In general, n-gram models capture a probability distribution over some domain of events. As a simple example, imagine that we would like to track the probability of a Sunny (S) day or Rainy (R) day of weather. To learn the probability of a Sunny day, we could observe days for some period of time (*e.g.*, 10 days), and count the number of Sunny days observed, $nSunny$. Then, we could assign the likelihood of a Sunny day occurring to be $\frac{nSunny}{10}$. When waking up each morning, we assume that the probability of a Sunny day is given by this same fixed rate. Under this interpretation, to find $P(SSSS)$ (*i.e.*, five Sunny days in a row), we would simply solve $P(S)^5$. We can compute probabilities based on a set of given observations. The observations can be mapped to a series of class labels $\{w_0, w_1, w_2, ..., w_n\}$. Applying the chain rule of probability theory yields the probability of a sequence according to some prior context available at each data point:

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2|w_1)...P(w_n|w_1w_2...w_{n-1}) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)...P(w_n|w_1^{n-1}) \\ &= \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \end{aligned} \quad (1)$$

The probability of $P(SSSS)$ would be given by

$$P(S) * P(S|S) * P(S|SS) * ...,$$

where $P(S|S_1..S_n)$ is the probability of S given we have seen the sequence $S_1..S_n$ immediately prior to S. When using this method, however, the number of parameters grow exponentially with the number of keywords in prior context. In some cases, we can reasonably apply the *Markov assumption*, which assumes that the probability of an event occurring is dependent only on the current "state" of a system. This state is defined as a fixed-length context or history, *h*.

n-gram models, then, are Markov models which use $(N-1)$ keywords of context to define the current state of the model.

 ISBN: 978-1-61208-376-6

Constructing, or training, an n-gram model requires the ability to observe example sequences occurring in the domain to be modeled. To train a model well, we need to observe single events from sequences in all relevant contexts.

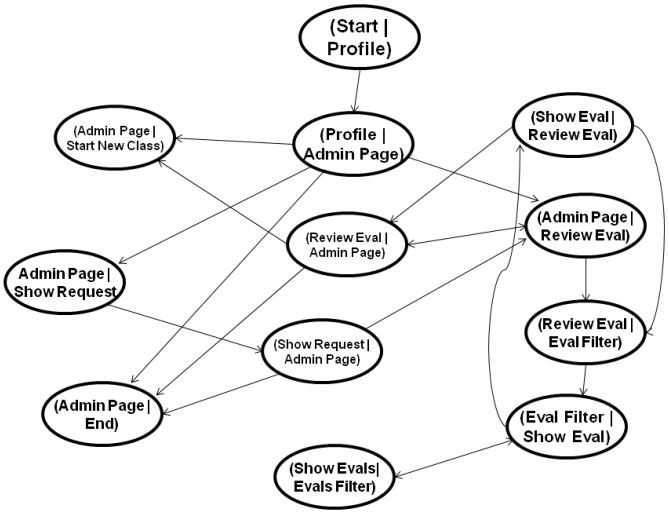


Figure 1. n-gram model of *User1* where n=3

In statistical analysis, it is sometimes difficult to estimate $P(w|h)$. Figure 1 represents an n-gram model of *User1*, where N=3 and *h*=2. In this figure, h is a sequence of the $(N-1)$ previous keywords. Probabilities are not included in this example. We deliberately keep the model simple by showing how current keywords depend on previous keywords when observing a sequence of actions. Since we are only concerned with a history of $h = 2$, the *(Start | Profile)* state is captured instead of *(Start)* as shown in Figure 1. As we transition from *(Start | Profile)*, we lose *Start*, but keep *Profile* to arrive at the *(Profile | Admin)* state. Now, every state that previously came from the *Admin* state, comes out of every *(<X> | Admin)* state. Once we construct an n-gram model in the training phase, we can use it for analysis of new event sequences in the test phase. With the *Markov assumption* incorporated into our construction of the model, we can apply the chain rule for conditional probabilities with a history of length $N-1$ to estimate the probability of any event sequence according to the existing model.

Finally, ID can be configured to incrementally improve its models during execution. Consider that User1 executes a new sequence of actions not contained in their n-gram model. This action can possible occur due to an update of the user interface, additiional duties added to User1, etc. ID will recognize the mismatch and ask for re-authentication (e.g., via a password, supervisoral approval). If this transaction is successful, then the sequence of actions that was not recognized is added to the n-gram model, thereby improving its accuracy. When fielded, we envision ID to be used in a training mode to build baseline models of all users; an then used in a deployment mode.

B. Challenges with n-gram Models

Even from a simple example, we see that additional data typically only stands to improve the reliability of a probabilistic model. With n-gram models, we capture conditional probabilities. The number of possible parameters needed by the model grows not only according to events observed, but also exponentially by the history being considered.

For the best possible model, our training phase requires observation of every possible event in every possible context. Additionally, we need to observe each context multiple times to have confidence that parameters are accurate. Without sufficient training, we may encounter event sequences during the test phase for which we have no knowledge to provide a probability estimate.

Others from the natural language domain have addressed this challenge of insufficient data for n-gram models. A concept known as *smoothing* involves saving some probability mass in an n-gram model's probability distribution for unseen events [15]. For the models in this paper, we have chosen to use the Kneser-Ney smoothing technique [16]. At a high level, Kneser-Ney involves:

- Reserving probability mass for unseen events by reducing all probabilities by a constant discounting percentage
- Estimate missing higher-order conditional probabilities by incorporating the observed frequencies of lower-order prefixes (a concept known as *backoff*)
- More precisely, combining the frequency of lower-order n-grams with a concept called a continuation probability, which estimates the probability that an event completes an n-gram, to estimate the probability of event sequences

For a more complete consideration of Kneser-Ney smoothing, the reader is referred to the original reference [16]. For our purpose, potential risks of applying Kneser-Ney to the domain of events carried out on software are violations of key features above. For example, if very few events are being performed by users, applying a constant discounting to every probability may save too much probability mass for unseen events. We will revisit the appropriateness of the model after gathering evidence in our experiments.

C. User Categorization

During the test phase of our experiments, we assign a probability to a sequence of events. We use binary categorization to judge a sequence as having likely been generated by a specific model (PASS) or not (FAIL). We introduce a probability threshold, t, for this pass/fail type of judgment for a sequence. Any sequence whose probability exceeds this threshold should be considered as a PASS, $+1$, and otherwise considered FAIL, -1.

A decison rule is used to predict the class membership of a given sequence of behavioral keywords, K. When new samples are encountered, the following decision rule is used:

$$\begin{cases} P(K,m) > t, & \text{then } y = +1 \\ P(K,m) < t, & \text{then } y = -1 \end{cases} \quad (2)$$

where $P(K,m)$ is the probability the behavioral keyword sequence is generated by the mth user's n-gram model. The probabilities are estimated using a training set of labeled data, $\{(m_0,y_0),(m_1,y_1),(m_2,y_2),...,(m_n,y_n)\}$, where label $y_i = \pm 1$ and depends on the class of m_i.

 ISBN: 978-1-61208-376-6

A more complex scheme that can also be useful for continuous authentication is multi-class categorization [17] [18]. For effective evaluation, this categorization method requires more training and test data. Under this decision-making approach, we can score an input sequence according to one of many models, and categorize the sequence as belonging to the model which estimates the highest probability. Therefore,

$$u = \arg\max_{m} P(K, m) \quad (3)$$

We use binary categorization by a simple threshold and multi-class categorization by comparing probabilities to translate n-gram models' estimations of sequence probability into decisions. We investigate our ability to make accurate decisions of various types as supported by these algorithms.

IV. Experiment

We now describe a set of experiments carried out on real user data designed to investigate the utility of n-gram models for CUA. In particular, we investigate the following research questions:

RQ1: Can we build discriminating user models to determine user types?

RQ2: Can the model recognize various legitimate users who are operating in the same user session?

RQ3: Can usage profiles be used to identify outliers in the user's behavior?

A. Subject System

We evaluate our proposed approach for CUA on an active government training support website for the high performance computing (HPC) community. This site is a repository for online training material which provides training course registration, course evaluation, information on several domain specific areas, and is a general source of information for its community of users.

Role	# of Users
Users	3775
Admins	6
Management	54
Technologist	2
Total # of users	**3837**

Figure 2. User Roles.

Each user account has an associated role. As shown in Figure 2, approximately 3800 users are in the "users" group which incorporates limited read access as well as the ability to evaluate and register for courses. These users do not access the system often. There are additional roles that provide access to areas of the system that are meant for administrative purposes; (*Admin, Management, Technologist*). The most prominent of these is the Admin role which has access to all areas. These users interact with the system often. Therefore, while we have more individual sessions available for the User role, the Admin role provides more per-user and per-session data. The Admin role also has the greatest risk for unauthorized use.

B. Data Collection

Because the system is web-based, access logs capturing hypertext transfer protocol (HTTP) requests made to the web server can provide sequences of user events. These logs are generated by the Tomcat JavaServer Pages (JSP) and servlet container. We analyzed the web log files for nine months to get accurate usage data for the system. User activity is largely based on role (i.e., access level). A user's role is monitored and used to verify their identity. This is under the assumption that users within the same role are likely to perform similar actions. System trust increases as the user interacts with the system if no outliers are identified in the CUA user model.

To validate data captured in a user's session, the following steps were used for preprocessing [19]:

1) *Data Cleaning*: The process of data cleaning is very important to generate an accurate picture of user activity when navigating a web application. For web logs of the subject system, various graphics and scripts are generated which add several entries to the log file. However, with our experiements, only JSP entries show user behavior and are important for logging purposes. Therefore, we remove all entries that are not related to user activity.
2) *User Identification*: Once the data is clean, the remaining file entries are grouped by individual user. Each user-id is associated with a role. User-ids are not continuously captured in the web log file. To solve this limitation, we check the session-id in a separate database table to capture unique user-ids.
3) *Session Identification*: Session identification is used to divide user accesses into individual sessions [19]. For the web access log, sessions are clearly identified. After checking the database for the role of each user, sessions are then grouped by role.
4) *Keyword Generation*: For each relevant JSP entry in the individual user log, a portion of the string is captured as a keyword. We are not considering parameters in this process because user entries would be too unique to categorize in a model.

When predicting individual user behavior (i.e., fine-grained approach), we filter the web logs to abstract only those users who have at least two sessions of activity and at least 80 keywords to ensure we have enough keyword data. After applying this filter, 31 users met this criteria and were used for this study. When predicting user types, 320 users were identified. In addition, at least two user roles must be present when predicting user types (i.e., user roles). To maintain the purity of the evaluation, we separate data into training and test sets, such that no model should be evaluated on its ability to classify data which was used during its own training.

C. Multi-class Categorization

For RQ1, we first want to consider whether unique profiles can indeed be constructed. To evaluate this research question, we develop models for each user role and use the multi-class categorization approach. Because we have four groups of users, a random model with no observation would achieve 25% accuracy in categorizing users. If our models are effectively capturing unique details about the behavior of users within roles, we would expect to see much greater overall accuracy

 ISBN: 978-1-61208-376-6

in categorization. We test the approach by comparing ID's recommended category to the actual category of each test session.

For the purpose of cross-validation, we performed various data splits for training and test data as seen in Figure 3 (50/50, 60/40, 70/30, 80/20, 90/10) to predict user roles. We reserve a precentage of sessions based on the data split to represent the testing set, E, and use the remaining data as the training set, R, to train an N order n-gram model for the specified class. We calculate $P(K, m)$ for each sample of keywords K from E. This represents the probability that the keywords are generated by the mth users n-gram model. Finally, m is selected with the highest probability for the behavorial keyword sequence, $P(K, m)$, and used as the prediction value. The y-axis, accuracy in Figure 3, shows this value for each data split.

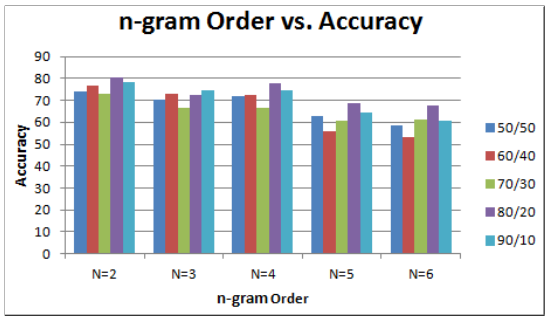


Figure 3. Prediction accuracy vs. n-gram order.

RQ1 results: Overall, the history order does not play a significant role. However, we do find that accuracy is highest when N=2 for each data split under evaluation when correctly categorizing test sequences into roles. Among each data split tested, the 80/20 data split has the highest accuracy, 80.5%, when N=2. The reported accuracies mark the mean performance over 10 separate trials, each of which chose random training data. For the 154 test sessions identified under this split, ID classified anywhere from 110 to 124 sessions correctly, depending on the sessions randomly chosen as training data. In most data splits, the accuracy began to decrease as the model order increased.

Based on these observations, we perform at a much greater rate of accuracy than a random selection of 25%, suggesting that at least some unique features of roles are effectively detected by the models. This shows using the models are both feasible and appropriate moving forward.

Instead of focusing on the four pre-defined user roles, RQ2 focuses on the ability of n-gram models to capture the behavior of specific users regardless of role. To evaluate this research question, we first filter the data under consideration to include only those users which have at least two sessions and at least 80 total keywords of input to ensure we have enough data to capture sequences. For the users meeting this criteria, we train models according to the n-gram approach.

RQ2 results: We achieved an overall accuracy rate of only 46% on the task of correctly categorizing test sequences by specific users. After filtering, we were left with only 28 test sessions. The tool correctly classified anywhere from 11 to 13 sessions correctly, depending on the sessions randomly chosen as training data. As in RQ1, the 46% overall mark represents the mean performance over 10 separate trials that were generated using random training data. We achieved our best results on this data when tuning model length to N=2 and implementing the 90/10 data split.

Working with so few test and training sessions, we have very little confidence in our evaluation of user specific profiles. In the future, we will be required to obtain much more data per user to effectively consider the construction of user-specific profiles.

D. Binary Categorization

When addressing RQ3, we want to consider whether the role-specific profiles constructed as part of RQ1 are capable of detecting outliers in user behavior. To evaluate this research question, we use the models from RQ1 independently in a binary categorization approach, as described in Section III-C. Because this task uses models independently, we use both training and test data from the remaining three roles to evaluate the model's ability to reject uncharacteristic sequences (i.e, negative examples). In addition, we use only test data from the model's own role to evaluate its ability to accept valid sequences (i.e., positive examples). Additionally, we would like to consider the effect of test sequence length (i.e., the number of keywords in an input sequence) on the performance of this task.

Finally, we track only a single threshold value for this task even though sequence length varies. A threshold which performs well for input sequences of length two would likely overestimate the threshold for longer lengths. As an alternative, we adapt the model's output probability to be an indicator of entropy, a measure of uncertainty in a system. This gives us the ability to normalize by the length of the input sequence. By doing so, we maintain a single threshold value for the binary classification task.

RQ3 results: Of all three research questions we consider, RQ3 relates most directly to the CUA goal of ID. Efficient experimental results are observed when accepting or rejecting snippets of user sessions based on role. We tested each model against every available subsequence of user keywords, from lengths two to nine with a probability threshold of -0.6. We obtained our best performance on this data when using N=9. Recall that backoff and smoothing in the model allow for the assignment of probabilities to a sequence of any length, regardless of the maximum history considered. Figures 4, 5, 6, and 7 show our findings. Note that accuracy is plotted separately for positive and negative test samples to analyze the models ability to reject uncharacteristic sequences and accept valid sequences.

We summarize the results as follows:

- As expected, the length-of-session examples provided to models significantly affects the ability to correctly classify the example.
- The effect of length was much greater on negative examples, as failures due to rejecting a positive sample were rare after lengths greater than four.
- The User model performed at a level greater than 90% on all samples for lengths greater than three.

 ISBN: 978-1-61208-376-6

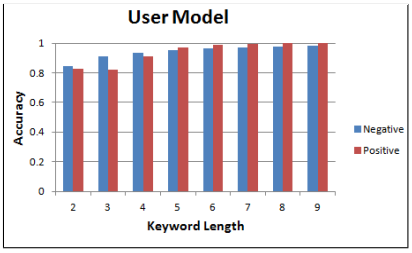


Figure 4. User n-gram model.

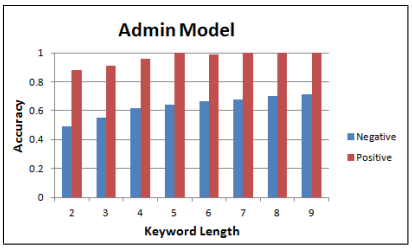


Figure 5. Admin n-gram model.

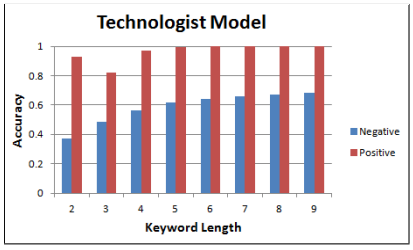


Figure 6. Technologist n-gram model.

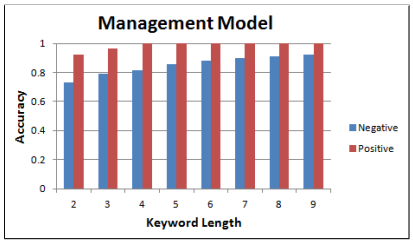


Figure 7. Management n-gram model.

- The Management model eventually achieved performance of 92%, though this took sessions of length nine.
- The Admin and Management models averaged 71% and 68% on negative examples, respectively, even at the maximum considered length of nine.

From these results, we can conclude that binary categorization proves to be much more effective than a random baseline at detecting uncharacteristic user behavior. For two of the four models considered, we achieved above 90% correct identification of negative samples and 100% correct acceptance of positive samples. In particular, the finding that User sessions can easily be protected against uncharacteristic usage is promising. Due to a large data set and elevated level of privileges for tasks, we expected the Admin user role to have one of the strongest models but this was not observed. In general, n-gram models seem much better suited for binary categorization tasks such as this one, especially given limited amounts of available data. In the future, perhaps multi-class categorization problems could be restated as a series of binary decisions.

Alternatively, the improvement in performance could be due to the use of shorter sessions which are less likely to contain unseen events. In this case, we rely on the validity of smoothing assumptions for accurate probability estimation. In the future, we will consider the effect of test example length on other tasks performed by ID as well.

V. Conclusions & Future Work

In this paper, we propose a continuous probabilistic authentication approach to model the behavior of users that interact with web-based software. A significant advantage of this mode of authentication is that it can be employed throughout the period of interaction, and hence, provide a natural way to continuously authenticate users. We have built a prototype, *Intruder Detector*, which keeps track of user actions, builds user profiles based on the n-gram language model and use these models to discriminate between user roles, and potentially finer-grained user profiles. Results show that *Intruder Detector* achieves 80.5% accuracy in user role classification tasks and nearly perfect accuracy when identifying categorization errors.

Although our pilot study has shown promising results, much work remains. In the immediate short term, we intend to work with a larger data set and compare various smoothing techniques. This will help improved the accuracy of RQ2 to correctly categorizing individual users. Capturing more data will help to better understand the characteristics of good, well-trained user models. We also plan to work out the details of fielding ID and evaluate two modes of operation: *training* in which models get built; and *deployment* in which the models get used for CUA. Afterwards, we will evaluate an alternative fielding strategy, one in which we have another level of authentication, using conventional means, in case ID identifies an intruder; this strategy will allow the models to iteratively get better (i.e., more accurate) with time.

Acknowledgment

This material is based on research sponsored by DARPA under agreement number FA8750-14-2-0039. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

[1] J. Liu, F. Yu, C.-H. Lung, and H. Tang, "Optimal combined intrusion detection and biometric-based continuous authentication in high security mobile ad hoc networks," Wireless Communications, IEEE Transactions on, vol. 8, no. 2, Feb 2009, pp. 806–815.

[2] K. Niinuma, U. Park, and A. K. Jain, "Soft biometric traits for continuous user authentication," Trans. Info. For. Sec., vol. 5, no. 4, Dec. 2010, pp. 771–780. [Online]. Available: http://dx.doi.org/10.1109/TIFS.2010.2075927

[3] R. P. Guidorizzi, "Security: Active authentication," IT Professional, vol. 15, no. 4, 2013, pp. 4–7.

[4] K. Niinuma, A. K. Jain, J. B. Kumar, S. Prabhakar, and A. A. Ross, "Continuous user authentication using temporal information," SPIE., vol. 7667, 2010.

 ISBN: 978-1-61208-376-6

[5] C. Shen, Z. Cai, and X. Guan, "Continuous authentication for mouse dynamics: A pattern-growth approach," in Proceedings of the 2012 42Nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), ser. DSN '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1–12. [Online]. Available: http://dl.acm.org/citation.cfm?id=2354410.2355184

[6] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," Future Gener. Comput. Syst., vol. 16, no. 4, Feb. 2000, pp. 351–359. [Online]. Available: http://dx.doi.org/10.1016/S0167-739X(99)00059-X

[7] D. Jurafsky and J. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd ed. Pearson Prentice Hall, 2009.

[8] S. Zhang, R. Janakiraman, T. Sim, and S. Kumar, "Continuous verification using multimodal biometrics," in Proceedings of the 2006 International Conference on Advances in Biometrics, ser. ICB'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 562–570. [Online]. Available: http://dx.doi.org/10.1007/11608288_75

[9] A. Azzini and S. Marrara, "Impostor users discovery using a multimodal biometric continuous authentication fuzzy system," in Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Part II, ser. KES '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 371–378. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-85565-1_47

[10] A. Altinok and M. Turk, "Temporal integration for continuous multimodal biometrics," in In Multimodal User Authentication, 03, pp. 11–12.

[11] A. J. Klosterman and G. R. Ganger, "Secure continuous biometric-enhanced authentication," Tech. Rep., 2000.

[12] M. Kaminsky, G. Savvides, D. Mazieres, and M. F. Kaashoek, "Decentralized user authentication in a global file system," in Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, ser. SOSP '03. New York, NY, USA: ACM, 2003, pp. 60–73. [Online]. Available: http://doi.acm.org/10.1145/945445.945452

[13] R. Chow, M. Jakobsson et al., "Authentication in the clouds: A framework and its application to mobile users," in Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop, ser. CCSW '10. New York, NY, USA: ACM, 2010, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/1866835.1866837

[14] H. B. Kang and M. H. Ju, "Multi-modal feature integration for secure authentication," in Proceedings of the 2006 International Conference on Intelligent Computing - Volume Part I, ser. ICIC'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 1191–1200. [Online]. Available: http://dx.doi.org/10.1007/11816157_148

[15] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in Proceedings of the 34th annual meeting on Association for Computational Linguistics, 1996, pp. 310–318.

[16] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on, vol. 1, 1995, pp. 181–184 vol.1.

[17] Y. Liu, Z. You, and L. Cao, "A novel and quick svm-based multi-class classifier," Pattern Recogn., vol. 39, no. 11, Nov. 2006, pp. 2258–2264. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2006.05.034

[18] P. Honeine, Z. Noumir, and C. Richard, "Multiclass classification machines with the complexity of a single binary classifier," Signal Processing, vol. 93, no. 5, 2013, pp. 1013 – 1026. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168412004045

[19] R. Cooley, B. Mobasher, and J. Srivastava, "Data preparation for mining world wide web browsing patterns," KNOWLEDGE AND INFORMATION SYSTEMS, vol. 1, 1999, pp. 5–32.

 ISBN: 978-1-61208-376-6

GAIA-MLIS: A Maturity Model for Information Security

Roger W. Coelho
Computer Science Department
State University of Londrina, UEL
Londrina, Brazil
rogercoelho04@uol.com.br

Gilberto Fernandes Jr.
Computer Science Department
State University of Londrina, UEL
Londrina, Brazil
gil.fernandes6@gmail.com

Mario Lemes Proença Jr.
Computer Science Department
State University of Londrina, UEL
Londrina, Brazil
proenca@uel.br

Abstract— **Information security management has become one of the most important areas for organizations in recent times. This is due to the increased need to protect data which is, in turn, one of the most important assets for any organization nowadays. Managing security risks is an ardous task which requires investments in support and technology management in order to succeed. Thus, there is great demand for a tool which is able to demonstrate the maturity level of an information security system, with the main objective of identifying key strengths and weaknesses in IT processes utilized by an organization. The GAIA-MILS model presented in this article has, as its main goal, to analyze the maturity level of an organization's information security system and supply them with key data on how they can improve. This proposed model presents descriptions of each different level in the areas of hardware, software, people and facilities. Its main objective is to diagnose and aid in the improvement of any identified weaknesses in the management of each specific area.**

Keywords - Maturity Level; Information Security; IT Governance.

I. INTRODUCTION

In the business world, asset information is seen as one of the most important within organizations. There are three distinct types which are considered most valuable: people, facilities and information [1]. Thus, security risk management is usually based on technology support and investment management [2].

The risks posed by information systems are not only complex but also difficult to quantify, since the damage can directly impact on the goal of the organization [5].

Organizations and service providers must develop protection tools in order to avoid misappropriation of user data. Thus, security threats such as viruses, worms, denial of service, submission of data by third parties, among others, cause concern for both users and service providers [3].

The Governance of Information Technology, aligned with good information security, is vital to the organization and service providers, since its credibility and reliability are tested every day. In addition, assessment methods can provide prescriptive data on how to improve the company management, as well as define who is responsible for the information and how it will be transmitted or maintained [15].

In conjunction with IT (Information Technology) governance, information security means keeping three main pillars: confidentiality, as information must be accessible only to authorized persons; integrity, to ensure that information is entirely transmitted; and usability, to guarantee authorized personnel access to the information and related resources when needed [4].

Organizations should assess their level of safety maturity through a formal model and utilize it as a parameter to measure the security risk. The model GAIA Maturity Level Information Security (GAIA-MLIS) aims to assess the maturity level of information security used in the evaluated network. For the purpose of implementing improvements in these processes, GAIA-MLIS enables companies to identify weaknesses in security processes, like hardware, software, human resources, facilities and information.

This article is organized as follows: Secion II deals with IT Governance and Information Security; Section III presents GAIA-MLIS Maturity Model Information Security; Section IV shows tests and results; and finally, Section V concludes the article.

II. IT GOVERNANCE AND SECURITY OF INFORMATION

Technological infrastructure is critical to daily operations within an organization and should be managed with defined processes. Accordingly, IT governance should focus on risk and resource management and strategic alignment to ensure that the technology and the active information adopt corporate objectives, maximizing benefits and opportunities as a means of acquiring competitive advantage [1].

IT governance has emerged as an auxiliary tool for managers, both in IT and other sectors of an organization, to help them comprehend the importance of all sectors working in alignment and, therefore more efficiently, in order to achieve their common goal [6]. IT is a strategic sector for an organization and it aids in revenue generation, contributing to the development of new technologies and technical support for other sectors. The Chief Information Officer

 ISBN: 978-1-61208-376-6

(CIO) must establish an effective governance, to improve the performance and success of the organization, supporting business strategies and plan of action [5].

Effective governance requires that the managers set standards and regulations for information assets. Information security is not only restricted to minimizing risks and failures, but it also affects the reputation of the organization, depending on how it acts on disaster recovery. The recovery organization defines the values and access permission information, thus everyone involved, customers, employees, among others, come to rely on the credibility of the organization [7]. Almost all organizations have their automated processes in their information systems, in order to ensure the efficient delivery of their services [17].

It is know that security is a method to protect the information against various types of threats ensuring continuity of business, higher return on investment and minimized risk. It is also the practice of ensuring the information can only be read, heard, altered or transmitted by people or organizations that have the right to do so. The main goals are confidentiality, integrity and availability. Confidentiality is the protection against theft and espionage. Integrity is the protection against non-authorized changes. Availability is the automated and secure access to the information users [12] [18].

Information security is achieved by means of an appropriate set of controls, which might include, policies, procedures, software, hardware, among others. All these controls need to be established, implemented, monitored, reviewed and improved in order to achieve the company's business targets. Likewise, security metrics have attracted the attention of the community for many years. However, the field is still lacking a formal model [16].

It is necessary that these controls are carried out in conjunction with security metrics to measure and compare the value of the security provided by different systems and settings [8].

The organization should always conduct audits at intervals of predetermined time in order to ascertain whether the control objectives, processes and procedures are meeting the security requirements of information identified, and if all objectives are maintained and implemented by executing them as expected. Control Objectives for Information and Related Technology (COBIT) aims to help businesses create an ideal value, referring to the IT sector, balancing and maintaining the resources from this area. Thus, COBIT version 5 allows organizations to manage their resources in a holistic way, with the goal of an end-to-end IT and functional areas considering both internal and external interest business [9].

For the development of a model of maturity level in information security, COBIT serves as a helper tool. Thus, the asset information gains importance in verifying the actual efficiency of the resources used for protection and obtaining a level of acceptance that is risky or not for the organization, since the information and its security must be established during the process of governance. The COBIT maturity model is used as basis for the GAIA-MLIS maturity model.

Information, systems, processes that support the organization, and even computer networks, are important assets to the organization's business. With the view to ensure greater competitiveness and visibility, the security information assets should be reviewed each time period and verified whether the initial planning is under execution or, the initial idea does or does not comply with the reality of the organization [7].

It is a fact that organizations often undergo various types of threats to their systems and computer networks, which may include, espionage, malicious persons within the enterprise and electronic fraud [11]. It is well known that organizations should understand the need for improvements in regards to risks they face and what targets and plans are in place [10].

Information security is important for any organization, whether a public agency with a model of electronic government (e-gov), or for a private enterprise [11].

Many systems are not designed for security. Some organizations do not have appropriate processes and procedures. It is essential that the requirements of information security are identified, analyzed and monitored, so that through continuous improvement, targets relating to information and its security are being met.

It is important to evaluate and establish a standard on an enterprise maturity level, so that both can be used to research through questionnaire or the construction of baselines about characteristics related to the use of technology. The use of baseline, or digital signature, has been used, for example, for establishment of standard and profile to network usage, as may be viewed in [21] and [22].

The standards ISO / IEC 27001:2005 and 27002:2005 aim to help IT managers and others, to establish what the security requirements are for the information which should be adopted. The standards serve as a guideline to develop practices and procedures for information security and assist in confidence building activities focusing on inter-organizational guidelines [19] [20].

III. MODEL OF GAIA-MLIS

Information is considered by many organizations as the asset which causes the most concern [13]. Defined processes help managers and employees to identify the requirements for decision making in order to protect all assets related to information [14].

The GAIA-MLIS maturity model aims to evaluate the level of maturity in information security and examines five areas, which are: Hardware, Software, Staff, Facilities and Information. All these areas are related to information. Through this model, organizations can verify the level of maturity in information security, identify if there is any deficiency and correct it in order to implement the improvement.

 ISBN: 978-1-61208-376-6

Figure 1 shows that the information has a centralizing role among all assets. Keeping information secure is one of the most difficult challenges that organizations have. Given that, many resources and processes should be measured by GAIA-MLIS model.

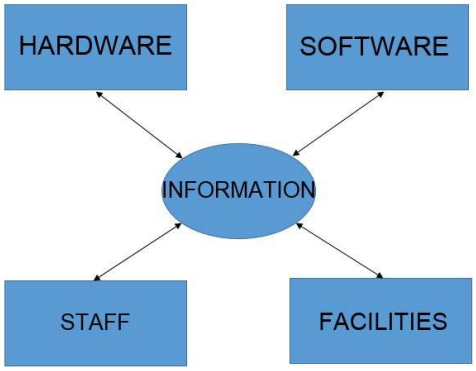


Figure 1. Relationship Areas.

A. Maturity Level GAIA-MLIS

Organizations are concerned with constant intrusions into computer systems. Processes in information security should be stored in environments that require more efficient security not only in computational media, but also in the physical environment with committed employees and a series of rules and procedures laid down in order to protect their information assets.

Since this procedure is not always carried out by the companies, along with the lack of knowledge of the importance of information, or non-commitment from the directors to the other employees, the creation of tools able to verify the security level of information is necessary for organizations. Thus, the GAIA-MLIS, model aims to analyze the level of maturity in information security in a particular company.

Through GAIA-MLIS, companies can verify what their weaknesses are in relation to information security and what targets they need to meet to achieve a certain level of information security. Through continuous planning, corporations can use the model in order to check whether goals are being met. The proposed model has five levels of maturity, which are goals and objectives describing what should be achieved by companies regarding the information security with a fully managed process.

The maturity model GAIA-MLIS is based on recommendations of COBIT 5 [9] and ISO / IEC 27001 [10] and 27002 [11] standards. The GAIA-MLIS maturity levels are described below.

Level 0, no insurance: Processes are not defined in information security. There are no defined responsabilities for information security policies. Employees and partners are unaware or are not trained with awareness programs on the importance of information security. Employees, partners and third parties do not suffer disciplinary proceedings upon the discovery of an information security incident. Shutdown policy of employees, partners and third parties policies are not applied upon termination and the return of organization's information assets. There is no security or access control defined process. Physical facilities are unsecured. There is no protection of equipment against external threats, whether human or environmental. There is no an efficient management for the network, avoiding or minimizing loss, damage or theft to information assets. Asset information is not encrypted. There is no backup policy with copies stored in monitored environments with access control in an environment protected against external threats. Inventories of assets are not identified and there are not established or documented. There are no classificaions of the importance and values of information.

Level 1, entry level insurance: Some processes are defined in information security. There are no defined sets for information security. Staff and partners are unaware or are not trained with awareness programs on the importance of information security. Employees, partners and third parties do not face disciplinary proceedings upon the discovery of a security incident information. Shutdown of employees, partners and third parties policies are applied haphazardly when closing the active. There is no security and access control process defined. Physical facilities are unsecured. There is some equipment protection against external threats, whether they are human or environmental. There is a basic management for the network without defined processes to avoid or minimize loss, theft or damage to information assets. Asset information is not encrypted. There are backup policy, but there are no copies stored in environments with access control, monitored and protected from outside threats. Assets inventory are not identified and are no established or documented. There are no classifications of the importance and values of information assets.

Level 2, regular insurance: Processes are defined in information security. There are few sets of defined responsibilities for information security. Staff and partners know, but they are not trained in awareness programs on the importance of information security. Employees, partners and third parties do not suffer disciplinary proceedings when some information security incidents are discovered. Shutdown of employees, partners and third parties policies are applied haphazardly when closing the active. There are some control access security set. Physical facilities are unsecured. There is some equipment protection against external threats, whether they are human or environmental. There is a basic management for the network without defined processes to avoid or minimize loss, theft or damage to information assets. Asset information is not encrypted. There are backup policy, but there are copies stored on environments without monitoring, access control and external threat. Inventories of assets are identified and established, but are not documented. There are no

 ISBN: 978-1-61208-376-6

classifications of the importance and values of information assets.

Level 3, partially safe: Processes are defined in information security and there are sets of defined responsibilities for information security. Staff and partners are trained in awareness programs on the importance of information security. Employees, partners and third parties sufferers disciplinary proceedings when an information security incident is discovered. Shutdown of employees, partners and third parties are partially documented. There is security and access control procedures defined. Physical facilities are protected. There is some equipment protection against external threats, whether they are human or environmental. There is an efficiently network managed, with some defined processes to avoid or minimize loss, theft or damage to information assets. Asset information is encrypted. There are backup policies and the copies are stored in monitored environments with access control and with protected against external threats to the environment. Inventories of assets are identified and established, but they are partially documented. There are classifications of the importance and values of information assets are partially documented.

Level 4, fully insured: Processes are defined in information security. Sets of responsibilities defined by security policy information. Staff and partners are trained in awareness programs on the importance of information security. Employees, partners and third parties sufferers disciplinary proceedings when an information security incident is discovered. Shutdown policies of employees, partners and third parties are totally documented. Access control are defined. Physical facilities are protected. The facilities are protected against external threats, both human and environmental. There is an efficient network management, avoiding or minimizing loss, damage or theft to information assets. Asset information is encrypted. There are backup policies and the copies are stored in monitored environments with access control and with protected against external threats to the environment. Inventories of assets are identified, established and registered. There are classifications established and the importance and values of information assets fully documented.

The maturity levels possess the following percentages: Level 0 has a percentage from 0% to 35%; Level 1 from 36% to 55%; Level 2 from 56% to 75%; Level 3 from 76% to 85%; and Level 4 above 85%. The percentages were assigned as described metrics of security levels. The empirical study was carried out to create an evaluation model for information security by analyzing the areas (hardware, software, staff, facilities and information), and these weights are an adaptation to what is suggested in the groups of ISO/IEC 27002. As observed, the levels are described as the overall organizational structure an organization might have, due to their maturity in information security. It is noteworthy that, through measurements of the formal model to assess GAIA-MLIS, organizations can plan and check the weaknesses in security processes.

The five ares (Hardware, software, facilities, staff and information) on GAIA-MLIS is addressed as in ISO/IEC 27002 standard. We may relate the areas of ISO/IEC 27002 (security policy, organizing information security, asset management, human resources security, physical and environmental security, communications and operations management, access control, information system acquisitions, development and maintenance, information security incident management, business continuity management and compliance) with five areas of GAIA-MLIS.

The evaluation will provide all companies, whether public or private, the ability to measure, manage and verify the asset information and use metrics to target higher levels by structuring its processes according to their needs and realities. Thus, the results obtained by supplementation of data areas provide greater control of the process used in information security, as well as manage the risks that organizations are subjected to every day.

IV. TESTS AND RESULTS

As means to verify and validate the maturity model GAIA-MLIS, three organizational structures were analyzed. The companies were not divided into sectors groups (service provider, bussiness company, etc), because we wanted to have a general sampling.

A questionnaire with thirty questions was administered in order to identify strengths and weaknesses in the processes of the five areas. The objective of the questions is to perform a diagnostic analysis of each area (hardware, software, people and facilities). The questions were developed based on the suggested groups of ISO/IEC 27002. There are five questions for the groups hardware, software, people and facilities, and ten questions related to the information area. The diagnose performed involves the application evaluation of security requirements related to policies and rules on the five suggested areas, assessing the investment degree and the use of technologies to guarantee each one of these areas. The weights of the questions were defined in an empirical way, and the information area has a higher number of questions than the other areas due to the fact that it is the analysis focus of the model. The mentioned areas have an assigned weight of: 30% for information, 25% for hardware, 25% for software, 15% for employees and 5% for facilities. These weights are an adaptation to what is suggested in the groups of ISO/IEC 27002.

Figure 2 below is a comparison of results from the analysis of different companies.

 ISBN: 978-1-61208-376-6

Figure 2. Results.

According to figure 2, Company 1 and 2 are at Level 1 maturity in information security. Meanwhile, Company 3 is at Level 2. Results show that the software area has more investment than others and facilities area has the lowest investment. A monitored environment may be able to inhibit harmful actions caused by employees or people who do not work in the organization. However, if the company does not provide training in accordance with the rules and punishments applied to employees, they face the risk of information security threats caused by internal factors.

These results indicate that there are more weaknesses than strengths in processes of the assessed networks, leaving companies with a level of information security level which is fragile and more susceptible to certain information security situations. Thus, companies should check and improve their processes, and directors may have GAIA-MLIS system as an analysis tool.

The system has proved to be efficient in indicating what level of maturity in information security the companies fall under. Figure 3 shows the trend lines for the three companies analyzed. These lines show their current status. Thereby, the results obtained in the tests enable defined strategies for improving processes and also indicate what their weaknesses are.

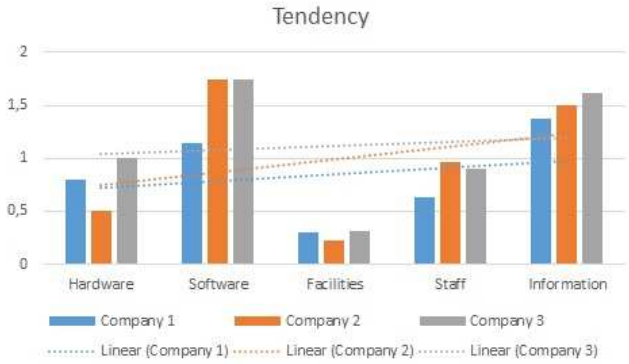


Figure 3: Tendency.

The GAIA-MLIS model contributes to a better management of information assets, analyzing five areas (hardware, software, staff, facilities and information), aiming to formalize metrics and levels of security. It creates value, in the sense that it allows for planned investments and formal documentation, defining standards and procedures for IT processes.

V. CONCLUSION

We presented the GAIA-MLIS model that aim to analyze maturity level for information security in enterprise, observing five areas (hardware, software, staff, facilities and information) through a diagnostic evaluation. We used three enterprise as object of analysis and we may see strengths and weaknesses in their areas of safety. With the results, we may evaluate what are the strengths and weaknesses of enterprise in each area, and what needs investments to improve the information security level.

The model helps organizations focus their efforts to solve specific problems in each one of the areas where the diagnostic evaluation identified a problem. The questionnaire application allows the exact identification of the area that needs investments in order to strengthen the security and, thus, improve the maturity level of the organization.

The flexibility of the analysis demonstrates that GAIA-MLIS system is able to state clearly the needs of each evaluated area. With the obtained results, the CIOs discuss the investment needs for all evaluated areas. Therefore, the CEO knows that the organization must change or create new policies and targets in order to aim at a better standard for the level of information security, demonstrating to partners and customers their concern with the integrity of all company assets, mainly with information.

Companies should establish policies and goals to aim for a higher level of security. GAIA-MLIS system provides companies metrics to identify the strengths and weaknesses of the processes. Investment in equipment and software techniques are important. However, if employees are not committed and if there is no a physical infrastructure able to protect the information assets, the organization will not be able to provide security for its network.

The proposed model achieved its objective of performing a security diagnosis evaluation, more specifically in hardware, software, people, facilities and information. It also helps the organizations on focus efforts to solve specific problems in each one of the areas in which the diagnostic evaluation found a problem.

An advantage of the model is the simplicity and the fast way with which it evaluates and diagnoses security maturity levels on the proposed subareas.

The corrective actions are directed according to the result of the diagnostic evaluation, and they aim to define policies of investment and adjustment on the analyzed areas in order to improve the information security.

In future works, we intend to analyze other companies separated by sector (service provider, public agencies, etc), aiming to adjust and improve the results according to characteristics common to organizations.

 ISBN: 978-1-61208-376-6

ACKNOWLEDGEMENTS

This work was supported by: SETI/Fundação Araucária and MCT/CNPq for Betelgeuse Projects' financial support.

REFERENCES

[1] R. V. Solms, K. L. Thomson and P. M. Maninjwa, "Information security governance control through comprehensive policy architectures", Proc. IEEE ISSA, IEEE Press, Aug 2011, pp 1-6.

[2] X. Yuan, Y. Zhou and Z. Qian, "Information Security of Power Corporations and its Reinforcement Measures", Proc. IEEE CICED, IEEE Press, Sep 2012, pp 1-7.

[3] P. I. Wang, "Information Security Knowledge and Behavior: An Adapted Model of Tecnology Acceptance", Proc. IEEE ICETC, IEEE Press, June 2010, pp v2-364 – v2-367.

[4] L. Qingguo and Z. Wei, "Strengthen Militaru Academy's Information Security Management", Proc. IEEE MINES, IEEE Press, Nov 2009, pp 182 – 186.

[5] J. Zhang, W. Yuan and W. Qi, "Research on Security Management and Control System of Information System In IT Governance", Proc. IEEE CSSS, IEEE Press, Jun 2011, pp 668-673.

[6] P. Weill and J.W. Ross, IT Governance: How top performers manage IT decision rights for superior results, Boston: Harvard Business Press, 2004.

[7] M. Sajko and N. Hadjina, "Information Security Governance and How to Accomplish it", Proc. IEEE MIPRO, IEEE Press, May 2011, pp 1516 – 1521.

[8] K. Sun, S. Jajodia, J. Li, Y. Cheng, W. Tang and A. Singhal, "Automatic Security Analysis Using Security Metrics", Proc. IEEE MILCOM, IEEE Press, Nov 2011, pp 1207-1212.

[9] ISACA, COBIT 5, A Business Framework for the Governance and Management of Enterprise IT. ISACA. 2012.

[10] ISO/IEC, Information technology – Security techniques – Information security management system - Requirements. ISO/IEC. 1ed. 2005.

[11] ISSO/IEC, Information technology – Security techiniques – Code of practice for information security management. ISO/IEC. 1ed. 2005.

[12] M. Moyo, H. Abdullah and R. C. Nienaber, "Information Security Risk Management in Small-Scale Organisations: A Case Study of Secondary Schools Computerised Information Systems", Proc. IEEE ISSA, IEEE Press, Aug, 2013, pp 14 – 16.

[13] L. Hong-li and Z. Ying-ju, "Measuring effectiveness of information security management", Proc. IEEE CNMT, IEEE Press, Jan, 2009, pp 1 -4.

[14] M. Ratchakom and N. Prompoon, "A Process Model Design and Tool Support for Information Assets Access Control using Security Patterns", Proc. IEEE JCSSE, IEEE Press, May, 2011, pp 307 – 312.

[15] M. Simonsson and P. Johnson, "The IT organization modeling and assessment tool: Correlating IT governace maturity with the effect of IT", Proc. IEEE HICSS, IEEE Press, Jan, 2008, pp 1 – 10.

[16] L. Krautsevich, F. Martinelli and A. Yautsiukhin, "Formal Analysis of Security Metrics with Defensive Actions", Proc. IEEE UIC/ATC, IEEE Press, Dec, 2013, pp 458 – 465.

[17] A. Chakraborty, A. Sengupta and C. Mazumdar, "A Formal Approach to Information Security Metrics", Proc. IEEE EAIT, IEEE Press, Dec, 2012, pp 439 – 442.

[18] B. Karabey and N. Baykal, "Information Security Metric Integrating Enterprise Objectives", Proc. IEEE ICCST, IEEE Press, Oct, 2009, pp 144 – 148.

[19] J. Anttila, K. Jussila, J. Kajava and I. Kamaja, "Integrating ISO/IEC 27001 and other managerial discipline standards with processes of management in organizations", Proc. IEEE ARES, IEEE Press, Aug, 2012, pp 425 – 436.

[20] A. Iqbal, D. Horie, Y. Goto and J. Cheng, "A Database for Effective Utilization of ISO/IEC 27002", Proc. IEEE FCST, IEEE Press, Oct, 2009, pp 607 – 612.

[21] E. Gomede, M. L. Proença JR and R. M. Barros, "Networks Baseline and Analytic Hierarchy Process: An Approach to Strategic Decisions", IADIS International Conference Applied Computing 2012, 2012, Madrid. Processing of IADIS International Conference Applied Computing 2012. Madrid, 2012. p. 34-41.

[22] M. L. Proença JR, C. Coppelmans, M. Bottoli and L. S. Mendes, "Baseline to help with network management", ICETE 2004 – Springer. (Org.). e-Business and Telecommunication Networks. Dordrecht: Springer, 2006, v. 1, p. 158-166.

 ISBN: 978-1-61208-376-6

Security of Vehicular Networks: Static and Dynamic Control of Cyber-Physical Objects

Vladimir Muliukha, Vladimir Zaborovsky, Sergey Popov

Telematics department St.Petersburg Polytechnic University
Saint-Petersburg, Russia
Email: vladimir@mail.neva.ru, vlad@neva.ru, popovserge@spbstu.ru

***Abstract*—The modern vehicle has long ceased to be a pure mechanical device. Each year data-processing component of the car is becoming more important. Considering the vehicle as the dynamic cyber-physical object in non-deterministic environment, we propose to use the methods of cloud services information security to solve the problem of access control to the cars telematics network. We propose to use a real-time control for each of these aspects, which is a complex technical challenge with static and dynamic interactions. The paper proposes a solution for implementing access control for vehicular networks. It is done by firewalls using dynamic access approach, based on virtual connections management, and algebra of filtering rules with mechanism of traffic filtering in "stealth" mode. The proposed security monitor architecture allows to enforce dynamic access policy depending on static set of firewall filtering rules and current condition of virtual connections and network environment.**

Keywords–Security; Vehicular network; Cyber-physics objects; Dynamic access control; Virtual connections.

I. INTRODUCTION

Information systems are deeper entering our lives, integrating with various purely physical systems. For example, a modern car is no longer a mechanical device. After enabling cyber component to all internal circuits and vehicular communications, it can be assigned to the new class – Cyberphysical objects. And each year this "cyber" component of the car is becoming more and more important.

In order to simplify the driving, more and more systems in the car become automated. Lots of the remaining mechanical systems in a modern car are controlled by computer via the Controller Area Network (CAN), but not directly by the driver. According to the researches, modern vehicles comprise up to 60-70 Electrical Control Units (ECUs). The ECUs serve a multitude of purposes like monitoring and controlling the different subsystems of a car [1][2].

Many of ECUs are connected together by the controller area network bus. Now, CAN is the most frequently used protocol in automotive networks, other protocols, designed to fit specific uses may also be used, such as Local Interconnect Network (LIN), Media Oriented Systems Transport (MOST) or FlexRay [1]. Such bus and ECUs form telematics network and serve as information and communication system of the modern vehicle. In modern vehicles, most of important functions are realized by telematics network. It measures vehicle's speed and revolutions per minute or informs the driver and other systems when an accident is about to occur and so on.

The world's largest automobile manufacturers are developing further, integrating in modern vehicles, more and more software and hardware to provide the owner and the driver of the car a maximum number of different digital services, including the remote ones.

According to the joint project of the Ford Motors Company and Saint-Petersburg State Polytechnical University, we suggest that in the near future, all new cars should be integrated into a single information service network and should be able to communicate with other cars and external sources, via USB, Bluetooth, WiFi, 3Gm and Long-Term Evolution (LTE) networks.

Digital revolution allows vehicles to significantly extend their functionality. Security means have to evolve together with cars. From 1960s to 2010s, vehicular security devices developed from mechanical through electromechanical and electronical to software based systems [3]. In the next few years, the car will be part of a single information and service space – cyber-physical object operating in the information space, which will result in a new class of security threats.

For several years, experts concerned with vehicular information security by hacking CAN network and replacing data from controllers. But, while maintaining speed and trends for Automotive Research in the near future, the hacking would be done remotely. This can lead to very bad consequences from data theft to a carjacking or damage the vehicle itself.

Thus, the issue of cars information security as the new class of cyber-physical systems that combine mechanical and electronic components is one of the most important issues of the vehicular networks.

The article describes cars as the new class of systems that combine the mechanical part and logical information, so-called cyber-physical objects. The security of information services for networks of cyber-physical objects is based on the access control technology.

The paper is organized as follows: In Section II, we consider the vehicle as the cyber-physical object. In Section III, we discuss security aspects of mobile cyber-physical networks using cloud computing security approaches. Section IV contains main aspects of the dynamic access control enforcement in computer networks. And in Section V, we suggest an architecture of a secure cloud computing environment. Section VI concludes the paper.

 ISBN: 978-1-61208-376-6

II. VEHICLES AS THE CYBER-PHYSICAL OBJECTS

In the near future, new generation of vehicles will be created. Such cars would be able to receive, store, and transmit information about their surrounding environment, which will be used during their operations. Information will be transmitted between such objects, between car and information center and also between the vehicle and the driver.

In our work, for the formalization of vehicular networks we use Cyber-Physical (CPh) approach, which extends the range of engineering and physical methods for a design of complex technical objects by researching the informational aspects of communication and interaction between objects and with an external environment.

The selection of CPh systems as a special class of designed objects is due to the necessity of integrating various components responsible for Computing, Communications, and Control (3C) processes. Although in modern science there are different approaches to the use of information aspects of the physical objects, but only within cybernetics, such approaches have had structural engineering applications. The conceptual distinction between closed and open systems in terms of information and computational aspects requires the use of new models, which take into account the characteristics of information processes that are generated during the driving of the vehicle and are available for monitoring, processing, and transmission via computer network.

According to Figure 1, a CPh model of a vehicular control system can be represented as a set of components, including following units: information about the characteristics of the environment (Observation), analysis of the parameters of the current state for the controlled object via CAN or telematics network (Orientation), decision-making according to the formal purpose of functioning (Decision), organization and implementation of the actions that are required to achieve the goal (Action). The interaction of these blocks using information exchange channels allows us to consider this network structure as a universal platform. Such platform allows us to use various approaches, including new algorithms and feedback mechanisms for the goals restrictions entropy reduction or the reduction of the internal processes dissipation.

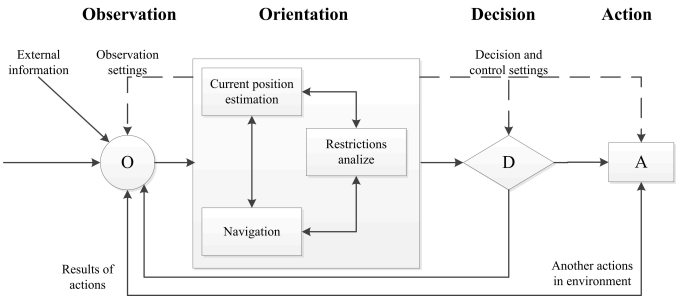


Figure 1. Cyber-physical model of vehicular control system.

Centric solutions allow using universal means for the organization of information exchange to integrate different technologies for the both observed and observable components of the control system. The main differences between these observed and observable components are the property ”part whole” (for observed components) and the ratio ”system environment” (for the observable ones). The parameters and the structure of such control system can quickly be adjusted according to the current information about the internal state of the object and the characteristics of the environment, which are in a form of digital data. Reported features open up the new prospects for the development of intelligent vehicular cyber-physical systems that will become in the near future an integral part of the human environment in the information space ”Internet of Things.” According to the estimates [4], network-centric cyber-objects in the global information space of the Internet will fundamentally change the social and productive components of people’s lives. That will accelerate of the knowledge accumulation and the intellectualization for all aspects of the human activity. However, this process requires not only innovative engineering ideas, but also the development of scientific concepts uniting universal scientific paradigm. Within this paradigm, for every CPh object like car, the information should be considered as a fundamental concept of objective reality, in which physical reality has ”digital” basis and therefore is computable. The idea of integrating the physical concepts with the theory of computation has led to the new conceptual schema for nature descriptions, known as ”it from bit” [5]. In this scheme, all physical objects, processes, and phenomena of nature, which are available to be read and understood by a person, are inherently informational and therefore they are isomorphic to some digital computing devices. Within this paradigm information acts as an objective attribute of matter that characterizes the fundamental distinctiveness of the potential states of the real object. The distinctiveness, according to the Landauers principle [6], is an energy factor of the objects states and that is why it gives an explanation of what are the states and how they are perceived by other objects. This distinctiveness appears while creating the systems that are capable to ensure the autonomy of the existence during the interaction with the external environment by the self-reproduction of its characteristics. It should be noted that on the way to the wide-spread use of ”digital reality” for the control problems, there are some limitations that reflect the requirements for the existence of the special state of physical objects reflecting its changes as a result of the information exchange processes. So, cyber-physical approach now often used to describe the properties of the so-called non-Hamiltonian systems in which the processes of self-organization are described by dissipative evolution of the density states matrix. However, the cyber-physical methodology may be successfully used to create complex robotic systems, the components which are capable for reconfiguration as the result of transmitting and processing digital data or metadata. The control and security tasks that are considered in this paper cover the actual scope of the cyber-physical approach, which is the basis of cloud computing technology and develop the methodology of cybernetics in the direction of the metadata control.

III. SECURITY ASPECTS OF CYBER-PHYSICS SYSTEMS

Modern technical systems have clear boundaries separating them from the environment or other objects and systems.

Therefore, the description of the processes in such systems is local and the change of its state can be described by the laws of physics, which are, in its most general form, the deterministic form of the laws of conservation, for example, energy, mass, momentum, etc. The mathematical formalization

 ISBN: 978-1-61208-376-6

of these laws allows to computationally determine the motion parameters of the physical systems, using position data on the initial condition, the forces in the system and the properties of the external environment. Although the classical methodology of modern physics, based on abstraction of "closed system" is significantly modified by studying the mechanisms of dissipation in the so-called "open systems", but such an aspect of reality as the information is still not used to build the control models and to describe the properties of complex physical objects. In the modern world, where the influence of the Internet, supercomputers, and global information systems on all aspects of the human activity becomes dominant, accounting an impact of information on physical objects cannot be ignored, for example, while realizing sustainability due to the information exchange processes. The use of cyber-physical methods becomes especially important while studying the properties of systems, known as the "Internet of Things" [6][7], in which robots, network cyber-objects, and people interact with each other by sharing data in the single information space for the characterization of which are used such concepts as "integrity", "structure", "purposeful behavior", "feedback", "balance", "adaptability", etc.

The scientific bases for the control of such systems were called Data Science. The term "Big Data" describes the process of integration technologies for digital data processing from the external physical or virtual environment, which are used to extract useful information for control purposes. However, the realization of the Data Science potential in robotics requires the creation of new methods for use the information in control processes, based on sending data in real time at the localization point of moving objects (the concept of "Data in motion").

In general, the "Big Data" are characterized by a combination of four main components (four "V"): volume, variety, velocity, and value. The general "V" is visibility of data and it is also a key defining characteristic of Big Data.

As a result, "Big Data" in modern science has become a synonymous for the complexity of the system control tasks, combining such factors of the physical processes that characterize the volume, velocity, variety, and value of data generated by them.

The security of CPh systems like vehicles is more complex task than access control in stationary local network. The cars move constantly changing the network configuration. Security policy enforcement requires data about permissions and prohibitions, as well as the current localization of the car and the route to it. Thus, while ensuring the information security of the vehicular network, we have to consider the static and dynamic aspects. This task is very similar to the information security of cloud services, where is regular migration of virtual machines from one physical platform to another to optimize the structure of the cloud and a hardware load balance.

The virtual computing environment allows us to create applications' service-oriented network of virtual devices. Any vehicle involved in the information exchange has its own virtual "avatar" in a high cloud environment. This virtual "avatar" of the car has all the information from the real object and the data obtained from other "avatars" in a virtual environment. Information exchange and required calculations are done in the secure virtual environment.

Localization of computing and data collected can accelerate the process of information processing and decision making. After that, the data is transmitted to the driver on the car for a final decision.

IV. Dynamic access policy using firewall filtering rules

The implementations of vehicular network security are far from simple due to the dynamic nature of network environment and users activity [7][8][9].

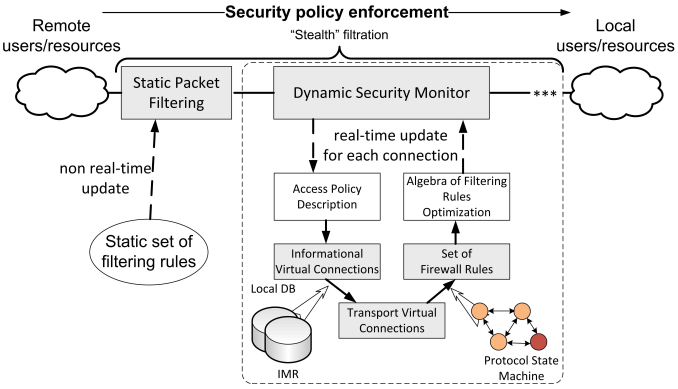


Figure 2. Security conveyor architecture.

Specific threats for the vehicular network are attacks affecting data integrity and availability of the car systems. An attacker often prefers not to steal any information out of the car but to modify it, thus tricking the various systems of the auto. Attack on availability can significantly hamper the work of the car, cutting off some of its devices from the information exchange process.

This specificity requires the use of specialized protective equipment. The main thread is while attacking the integrity of the information, the attacker can spoof the signals from the remote control signal, for example, acting as the owner and steal the vehicle.

In vehicular network, every second, hundreds of users and telematics devices establish new virtual connections [10] with distant resources and services. According to the mandatory security policy, if we have N users, M resources and these numbers are big, than we have to create a huge access N*M matrix. And each element of this matrix will be a vector of parameters and attributes, describing one virtual connection. Vehicles and resources of course can be grouped according to their rights and context, but in either case such matrix is too big to be processed efficiently in real-time.

We propose the architecture of security conveyor (see Figure 2), which consists of several filtering stages. At the first stage when a connection is established there is a classical static packet filter, which reject prior harmful traffic that corresponds local telematics devices of the car. The second stage enforces more accurate dynamic aspect of the access policy. Doing it the dynamic firewall have to take into account that the network resources can change their context any moment without informing our security conveyor. That is why we propose to use some prior information about remote users and resources in firewall to enforce security policy. Such information should be received from databases and services outside of our security monitor.

 ISBN: 978-1-61208-376-6

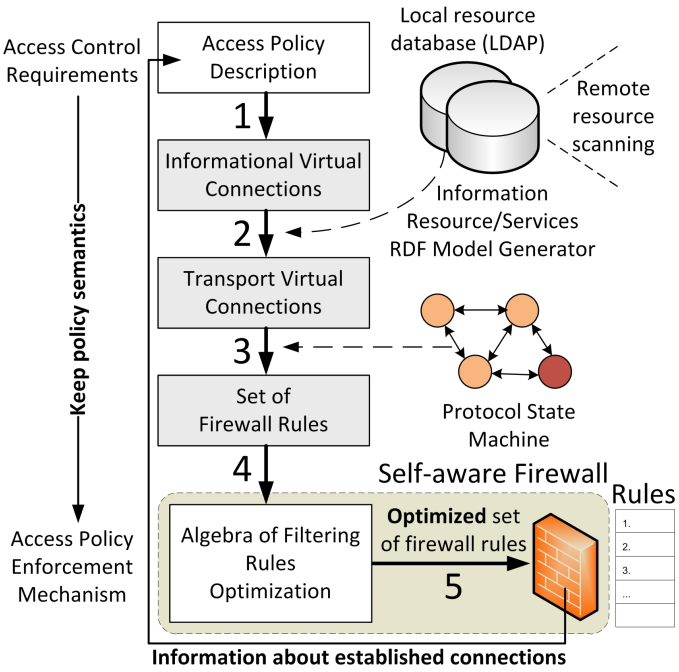


Figure 3. Dynamic access control approach.

In computer networks, information is transmitted in the form of packets. Each of these packets has some part of message in it. All network devices such as vehicular transmitters and firewalls have to operate with these packets. According to Figure 3, every access rule can be considered as one or more informational virtual connections (1) specifying action as ”access”, ”read”, ”write”, and so on. Then, we have to determine what does this record mean to telematics devices, how can the vehicle and requested information resource be described. To answer these questions, our security monitor has to use some prior outside information from specialized databases and services (2). Using such information we receive one or more Technological Virtual Connections (TVCs) from initial informational virtual connections [10]. Then, all these TVCs rules should be transformed into the requirement to the packet filter. At this stage, we use different transport protocols state machines to receive information about packet sequences (3). If all these procedures will be applied to each established virtual connection well receive huge amount of filtering rules. That is why at the next stage, we propose to optimize the set of filtering rules using specialized algebra of filtering rules (4) [11]. Only optimized set of filtering rules can be processed in real-time by firewall to enforce access policy (5).

V. Architecture of a Secure Cloud Computing Environment

During the researches at the Telematics department of SPbSPU, we proposed architecture of a secure cloud computing environment. This architecture considers dynamic nature of cloud environment and is suitable for description of vehicular networks.

A distributed computing environment (cloud system) consists of the following software and hardware components:

1) Virtualization nodes;
2) Storage of virtual machines and user data;
3) Cluster controller;
4) Cloud controller.

The distributed computing environment intended for solving scientific and engineering problems is a set of various computing resources, such as virtual machines, and has the following features [12]:

1) The environment is used by a wide range of users, who are solving problems of different classes;
2) Virtual machines of different user groups can operate within one hypervisor;
3) Wide range of software components (Computer-Aided Design (CAD)/Computer-Aided Engineering (CAE) applications, development tools) and operating systems is used;
4) Different hardware configurations are used, including virtual multicore computing machines and virtual machines, which allow performing computations using the streaming technology Compute Unified Device Architecture (CUDA).

Virtualization node is the hypervisor software which runs on powerful multicore computing node. In virtualization, the domain level 0 (dom0 in terms of hypervisor XEN or service console in terms of other hypervisors) and virtual computing machines (domain level U, domU) operate.

For information security and Access Control (AC) between the virtual machines that operate under a single hypervisor, the internal (”virtual”) traffic and the external traffic (incoming from other hypervisors and from public networks) must be controlled. The solution of the access control problem could be achieved through the integration of a virtual firewall into the hypervisor; this firewall would functions under the hypervisor, but separately from the user virtual machines. The virtual firewall domain can be defined as ”security domain” (domS). Invisible traffic filtering is an important aspect of the network monitoring; the firewall must not change the topology of the hypervisor network subsystem. This can be achieved by using ”Stealth” [12] technology, which is a packet traffic control invisible to other network components.

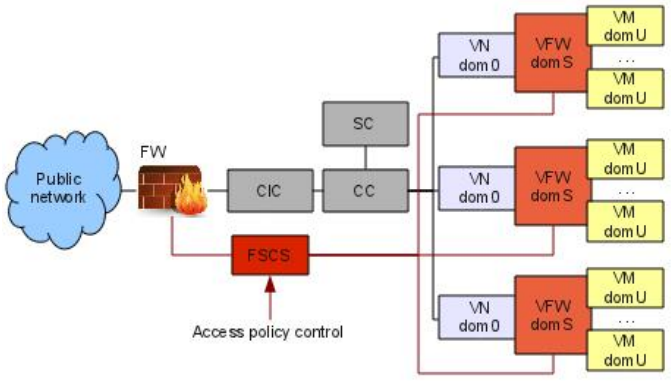


Figure 4. Secure cloud architecture.

Figure 4 shows the common architecture of a distributed cloud system with integrated AC components. The following abbreviations are used: hardware FireWall (FW); Virtual FireWall (VFW); the Central Control System of all Firewalls in

the cloud (FSCS); Virtual Machine (VM); CLoud Controller (ClC); Cluster Controller (CC); Storage Controller (SC).

The FSCS distributes the access control policies to all firewalls in the system. When the information security policy changes, new access rules are replicated to all components. The security domain isolates virtual machines from the hypervisor, which prevents the possibility of attack against the hypervisor inside the cloud. The hardware firewall isolates the private cloud components from the external threats.

The joint use of hardware and software firewall and intrusion detection system, based on the prediction of the driver's and vehicular's behavior and the vehicular network state will reduce the risks of invasion in a car network. Using a virtual machine "avatar" in a cloud computing environment, allows a better control of the processes of information exchange and the current status of all road users.

The task of finding an optimal allocation of virtual machines to minimize the number of nodes used by the cloud system is similar to the N–dimensional problem of packing containers (Bin Packing Problem), where N corresponds to the number of virtual machine's selected characteristics taken into account in the allocation. In [13], specialists of our department proposed a new approach for virtual machines distribution. A new virtual machines scheduler is able to place a virtual machine on the optimal compute node and migrate it to another node if resource consumption state has been changed. In [13], the proposed algorithm allows to optimize the structure of high-performance computing cloud system, facilitates localization of data and computing resources, and reduces the time required to provide a user requested services.

Cloud can improve system performance through the use of parallelization technology. When a large multi–node cluster needs to access large amounts of data, task scheduling becomes a challenge. Apache Hadoop is a cluster computing framework for distributed data analytics. However, the performance of each job depends on the characteristics of the underlying cluster and mapping tasks onto Central Processing Unit (CPU) cores and Graphics Processing Unit (GPU) devices provides significant challenges. Spark provide interesting advantages to the typical Hadoop platform [14]. Spark is an open source cluster computing system provides primitives for in–memory cluster computing. Job can load data into memory and query it repeatedly much quicker than with disk–based systems.To make programming faster, Spark integrates into the Scala language. Scala is statically typed high–level programming language designed to express common programming patterns in a concise, elegant, and type–safe way. Scala runs on the Java Virtual Machine (JVM) so it integrates features of object–oriented and functional languages. Spark is built around distributed datasets that support types of parallel operations: transformations, which are lazy and yield another distributed dataset (e.g., map, filter, and join), and actions, which force the computation of a dataset and return a result (e.g., count) [15]. In our work, we propose to use Deep Content Inspection (DCI) that reconstructs, decompresses, and decodes network traffic packets into their constituting application level objects. DCI examines the entire object and detects any malicious or non-compliant intent.

While solving information security problems for vehicular networks, we rely on our expertise in the field of robots control, for example during the space experiment "Kontur-2" [16]. Using DCI for network traffic monitoring enables us to provide the required level of security for on-surface robots, and traffic prioritization methods in packets processing allow us to provide the required level of Quality Of Service (QoS) [17] [18].

VI. Conclusion

Considering the vehicle as the dynamic cyber-physical object in non-deterministic environment, we propose to use the methods of cloud services information security to solve the problem of access control to the cars telematics network.

Vehicular security devices developed from mechanical through electromechanical and electronical to software based systems.

From the viewpoint of information security, the vehicle can be regarded as the dynamic virtual machine in the cloud environment.

In this paper, we propose an architecture of a secure cloud computing environment, which involves the use of static and dynamic access control methods.

It is necessary to mention that proposed solution doesnt solve all security problems of vehicular networks. The model described above can be merged easily with other methods of security control, for example with encryption or obfuscation. The prototype of the proposed system is currently developing for the Ford Motors Company at the Telematics department of the Saint-Petersburg State Polytechnical University.

Acknowledgment

This paper funded by RFBR grant 13-07-12106 and is done in the framework of the project with the Ford Motor Company.

References

[1] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaâniche, and Y. Laarouchi, "Security of embedded automotive networks: state of the art and a research proposal," in SAFECOMP 2013 - Workshop CARS (2nd Workshop on Critical Automotive applications : Robustness & Safety) of the 32nd International Conference on Computer Safety, Reliability and Security, Toulouse, France, 2013, J. Fabre, P. Quéré, and M. Trapp, Eds. HAL, 2013. [Online]. Available: http://hal.archives-ouvertes.fr/SAFECOMP2013-CARS/hal-00848234

[2] B. Donohue, "Hacking the Modern Automobile," 2013, URL: http://blog.kaspersky.com/car-hacking/ [accessed: 2014-10-01].

[3] A. Weimerskirch, "Automotive Data Security," 2012, URL: http://www.sae.org/events/gim/presentations/2012/ weimerskirch_escrypt.pdf [accessed: 2014-10-01].

[4] A. L. Fradkov, Cybernetical Physics: Principles and Examples. Nauka, Saint-Petersburg, Russia, 2003, ISBN: 5-02-025028-7.

[5] J. A. Wheeler, "Information, physics, quantum: the search for links," in Proceedings of the 3rd International Symposium Foundations of Quantum Mechanics in the Light of New Technology, Kokubunji, Tokyo, Japan, August 28-31, 1989, N. B. Gakkai, Ed. Hitachi, Ltd., 1989, pp. 354–368.

[6] G. Niemeyer and J.-J. E. Slotine, "Telemanipulation with time delays," The International Journal of Robotics Research, vol. 23, no. 9, 2004, pp. 873–890. [Online]. Available: http://ijr.sagepub.com/content/23/9/873.abstract

[7] V. Zaborovsky, O. Zayats, and V. Mulukha, "Priority queueing with finite buffer size and randomized push-out mechanism," in Networks (ICN), 2010 Ninth International Conference on, April 2010, pp. 316–320.

[8] V. Zaborovsky and V. Mulukha, "Access control in a form of active queuing management in congested network environment," in Networks (ICN), 2011 Tenth International Conference on, 2011, pp. 12–17.

 ISBN: 978-1-61208-376-6

[9] V. Zaborovsky, A. Gorodetsky, and V. Muljukha, “Internet performance: Tcp in stochastic network environment,” in Evolving Internet, 2009. INTERNET ’09. First International Conference on, Aug 2009, pp. 21–26.

[10] V. Zaborovsky, V. Mulukha, A. Silinenko, and S. Kupreenko, “Dynamic firewall configuration: Security system architecture and algebra of the filtering rules,” in Evolving Internet, 2011. INTERNET ’11. Third International Conference on, Jun 2011, pp. 40–45.

[11] V. Zaborovsky, V. Mulukha, and A. Silinenko, “Access control model and algebra of firewall rules,” in WORLDCOMP11: Proceedings of the 2011 International Conference on Security and Management (SAM2011). CSREA Press, Jul 2011, pp. 115–120.

[12] V. Zaborovsky, A. Lukashin, S. Kupreenko, and V. Mulukha, “Dynamic access control in cloud services,” in Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on, Oct 2011, pp. 1400–1404.

[13] A. Lukashin and A. Lukashin, “Resource scheduler based on multi-agent model and intelligent control system for openstack,” in Internet of Things, Smart Spaces, and Next Generation Networks and Systems, ser. Lecture Notes in Computer Science, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer International Publishing, 2014, vol. 8638, pp. 556–566.

[14] T. Kumawat, P. K. Sharma, D. Verma, K. Joshi, and K. Vijeta, “Implementation of spark cluster technique with scala,” in International Journal of Scientific and Research Publications (IJSRP), vol. 2, 2012. [Online]. Available: http://www.ijsrp.org/research-paper-1112/ijsrp-p1181.pdf [accessed: 2014-10-01]

[15] A. Lukashin, L. Laboshin, V. Zaborovsky, and V. Mulukha, “Distributed packet trace processing method for information security analysis,” in Internet of Things, Smart Spaces, and Next Generation Networks and Systems, ser. Lecture Notes in Computer Science, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer International Publishing, 2014, vol. 8638, pp. 535–543.

[16] V. Zaborovsky, M. Guk, V. Muliukha, and A. Ilyashenko, “Cyber-physical approach to the network-centric robot control problems,” in Internet of Things, Smart Spaces, and Next Generation Networks and Systems, ser. Lecture Notes in Computer Science, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer International Publishing, 2014, vol. 8638, pp. 619–629.

[17] A. Ilyashenko, O. Zayats, V. Muliukha, and L. Laboshin, “Further investigations of the priority queuing system with preemptive priority and randomized push-out mechanism,” in Internet of Things, Smart Spaces, and Next Generation Networks and Systems, ser. Lecture Notes in Computer Science, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer International Publishing, 2014, vol. 8638, pp. 433–443.

[18] V. Muliukha, A. Ilyashenko, O. Zayats, and V. Zaborovsky, “Preemptive queueing system with randomized push-out mechanism,” Communications in Nonlinear Science and Numerical Simulation, 2014, p. in print. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1007570414004031

 ISBN: 978-1-61208-376-6

Digital Signature of Network Segment Using Genetic Algorithm and Ant Colony Optimization Metaheuristics

Paulo R. G. Hernandes Jr.*, Luiz F. Carvalho†, Gilberto Fernandes Jr.†, Mario Lemes Proença Jr.†
*Security Information Department, São Paulo State Technological College (FATEC), Ourinhos, Brazil
†Computer Science Department, State University of Londrina (UEL), Londrina, Brazil
{paulogalego, luizfcarvalhoo, gil.fernandes6}@gmail.com, proenca@uel.br

***Abstract*—Every day computer networks have more significance in our lives, and these network's complexity is still growing. To help customers achieve maximum productivity and avoid security risks, network administrators have to manage network resources efficiently. Traffic monitoring is an important task, which describes the network's normal behavior. Thereby, we present a Digital Signature of Network Segment using Flow Analysis (DSNSF) as a mechanism to assist network management and information security through traffic characterization. Our new approach uses a genetic algorithm to optimize the process. In order to accomplish this task, we compared the novel model with another similar method, Ant Colony Optimization for Digital Signature (ACODS), using a real data set of traffic for bits and packets. We also evaluate these models to measure their accuracy.**

Keywords–Traffic Characterization; Traffic Monitoring; Network Management; Genetic Algorithm, sFlow.

I. INTRODUCTION

Network Management is a complex task which utilizes different tools and techniques. These tools and techniques aim not only to help network administrators in their daily tasks, but also to provide them with mechanisms which enable them to detect information regarding security events in order to avoid security incidents.

Since the first networked computers, the administrators required all the necessary information about their equipments, so they could understand the behavior of their network by observing information, such as an interface's traffic or which port in a remote switch are being used. Thereby management protocols and tools emerged.

The Simple Network Management Protocol (SNMP) became popular because of its simplicity and its ability to be used by most equipment manufacturers [1]. Using SNMP, we can monitor whether the equipment is functioning, its traffic average and other additional information when required. Nevertheless, with the increase of the complexity of applications that run on networks, such as VoIP, P2P, video on demand, and also the increase of mobile equipment and the Internet of Things, an SNMP protocol alone was not enough for all information required by the network administrators. With the use of data flow, network administrators could have more detailed information to take decisions quicker and more efficiently.

A flow record reports at least the endpoint addresses, time, and volume of information transferred between two sockets. This gives a better view of the traffic than interface-level counters queried by SNMP, and it provides significant data reduction compared to packet traces, allowing it to scale to large networks [2]. The study of flow records can help network administrators identify anomalies in their environments. As a result, researchers are trying to find anomaly detection models based on traffic characterization. These models, as described by Lakhina *et al.* [3], are able to identify an anomalous behavior based on traffic history, learning the standard behavior of an environment, and based on its history detect changes in the network routine.

A network anomaly detection system, first creates a baseline profile of the normal system, network, or program activity. Thereafter, any activity deviating from the baseline is treated as a possible intrusion [4]. It helps administrators to identify any attack or network anomalous behavior, such as users running a P2P application or any other activity which is against company policies.

To reach our target, we use a Genetic Algorithm (GA), a model which simulates the natural evolution process through operators such as selection, crossover and mutation [5]. GA is recognized as an ideal optimization technique to solve large variety of problems. One of the best uses for GA is to optimize search problems, or organize data under some conditions.

Our proposal is to create a Digital Signature of Network Segment using Flow Analysis (DSNSF) utilizing GA to optimize the clustering process and characterize network traffic using flow analysis to permit detection of network anomalies. We use a real set of data to perform the process and evaluate the results to prove the accuracy of our model. Also, we compared this technique with another approach, the Ant Colony Optimization for Digital Signature (ACODS).

This paper is organized as follows: Section II presents the related work. Section III details the novel method DSNSF-GA and also the ACODS approach, both used to characterize network traffic. Section IV delivers the generation of the DSNSF-GA. Section V presents the result of our evaluation tests, and finally Section VI concludes this paper.

II. RELATED WORK

The target of our work is to characterize network traffic and permit network administrators identify anomalous behavior in their environments. For this purpose, we created a DSNSF. This methodology to characterize network traffic was proposed by Proença *et al.* [6] in which a Digital Signature of Network Segment (DSNS) was created using data of each day, usually a workday, based on the history of the previous weeks.

A Firefly Harmonic Clustering Algorithm (FHCA) [7], an optimized clustering algorithm based on the fireflies behavior and its emitted light characteristics, used data acquired using SNMP. To characterize network traffic, certain techniques could be applied such as Holt-Winters for Digital Signature

 ISBN: 978-1-61208-376-6

(HWDS), a modification of the classic statistical method of forecasting Holt-Winters [8] or the K-means for Digital Signature (KMDS) [9], where a DSNS is created using K-Means clustering technique. The ACODS approach presented by Carvalho *et al.* [10] is based on Ant Colony Optimization metaheuristic. For DSNSF creation, the ACODS aims to optimize the clustering process, seeking solutions to make it possible to extract patterns of network traffic.

GA was proposed by Holland [5] to simulate the natural evolution process, and it is recognized as an ideal solution to solve problems with a large solution variation. One of the usages of GA is the optimization of the clustering process. A cluster is a group of data which are organized in groups. Data within the same group should be similar and data within other groups should be different. A group is also called cluster. A genetic algorithm-based clustering technique was proposed by [11] and uses the Euclidean distance as the objective function, to classify in which cluster each point should be. This is an example of a profitable way to organize data among clusters using GA and clusterization.

In Xiaopei *et al.* [12], an Artificial Immune System is used along with GA in order to optimize the process. An immune system produces plenty of antibodies to resist an antigen, which is an attribute much similar to the individual diversity of GA. Applying GA to this memory identifying function can enhance the quality of the generated detectors therefore improving the efficiency of detection. In Guo *et al.* [13], a Network Anomaly Intrusion Detection based on Genetic Clustering uses the cluster centers as binary code to organize data and detect intruders. However, if the number of clusters and the length of chromosomes are too large, a system operation inefficiency will be detected.

III. Generation of DSNSF

In this section, we present two metaheuristic strategies to create a DSNSF using data as bits and packets per second. These data were collected using sFlow to generate flows from the network's assets. Our purpose in this work is to demonstrate that flow attributes bits and packets per second can be used to identify a normal, or expected, traffic pattern. The first model is based on the natural evolution of species theory, implemented in computing as Genetic Algorithm, which simulates the natural process of evolution in a population. The second uses the Ant Colony Optimization process, which is based on ant colonies' behavior. Both methods are appropriate to the DSNSF construction and they will be described ahead.

A. DSNSF-GA

Our DSNSF-GA uses the Genetic Algorithm based approach to organize data in clusters. These data were collected using sFlow in State University of Londrina (UEL). We use the average among cluster centroids to generate a graph that will show the network traffic by bits and packets per second using the last three determined days in the week, and compare them with the data of the current day to detect network anomalies. For example, for a Monday we use data from the last three Mondays (excluding the current day) to plot a graph with the characterized network, and compare with these to detect anomaly behavior.

GA is a technique which manipulates a population of potential problem solutions trying to optimize them. Specifically, they operate with a coded representation of solutions which would be equivalent to genetic material (chromosomes) of individuals in nature. Each individual will be assigned a fitness value that will reflect the individual adaptability in an environment in comparison with others. As in nature, the selection will elect the fittest individuals. These will be assigned for a genetic combination, also called crossover, which will result in the exchange of genetic material (reproduction), generating a new population.

Mutation is a value modification in some genes belonging to a solution with some probability p' (the mutation probability). The function of mutation in GA is to restore lost or unexplored genetic material in the population to prevent a premature convergence of a sub-optimal solution, and also to try to find a better solution. Selection, crossover and mutation will repeated for several generations for a fixed number of times, or until some condition is reached. This cycle is represented in Figure 1.

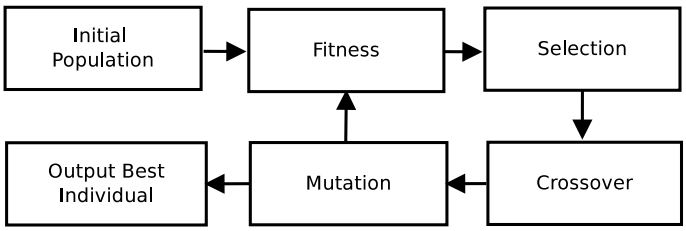


Figure 1. Genetic Algorithm cycle.

To help these mechanisms, there are other operations which must be used to complete the GA process, such as selection engine and crossover point selection. We must also determine our objective and fitness functions, which are the functions being optimized. In our approach we use the Euclidean distance as an objective and fitness function.

The number of clusters is fixed. We use the Euclidean distance to determine the best distance of each point from their respective cluster centers. Our chromosome was represented by a real number, which is the value of the cluster center.

The Euclidean distance is given by

$$J = \sum_{i=1}^{E} \sum_{j=1}^{K} \sqrt{\sum_{n=1}^{N} (x_{in} - c_{jn})^2} \quad (1)$$

in which K is the total number of clusters, E represents the amount of flows to be clustered and N indicates data dimensionality, i.e., number of features to be processed. The collected flows are divided in 5 minute intervals, totaling 288 data sets throughout the day. The variable x_{in} denotes value of the feature n of flow i and c_{jn} stores value of center of cluster j at n dimension.

We chose the Roulette Wheel approach to find the fittest in the population, which conceptually consists of giving each individual a slice of a circular wheel equal in area to the individual's fitness [14]. We spun the roulette for the same number of individuals in the population. Those which were selected more often were the fittest and will breed the new generation.

 ISBN: 978-1-61208-376-6

The crossover operator combines the chromosomes of two parents to create a new one. In our approach, we have chosen a random number to divide both chromosomes in the same point and merge them in another one (child). This process will continue until the old population be replaced by a new population of "children".

To start the process, we generate a random initial population in which we began applying the crossover, selection and mutation. As we have described, our chromosomes are the cluster centroids values. We appointed an initial population of forty parents. They create the new generation, which will replace the old one. It will repeat for a number of sixty iterations. For our purpose we conclude that sixty is an ideal number, because a higher number will increase the effort unnecessarily, and a lower one will not create an ideal solution.

At the end of this process, we have the best chromosome based on its fitness function. This value represents a single point in the DSNSF-GA. We have to apply the clusterization using GA for each point in the graphic, so it will be repeated for 288 times, one point every five minutes.

B. ACODS - Ant Colony Optimization for Digital Signature

Nature has been inspiring men in creating solutions for human problems. Hence, a variety of biologically inspired approaches have appeared in various research fields such as engineering, computer science, medicine and economics. For example, the ants' ability to find the shortest path between their colony and the food source has inspired the creation of a very promising method called Ant Colony Optimization (ACO) metaheuristic [15].

Similarly to real ants, ACO agents are able to organize themselves using pheromone trail. They travel through the search space represented by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of all nodes and $\mathcal{E}$ is the edges set. Agents are attracted to more favorable locations to optimize an objective function, i.e., those in which the concentration of pheromone deposited by ants which previously went through the same path is higher.

According to Jiang *et al.* [16], the ant colonies' habits living in groups are essentially similar to the grouping of data. Algorithms based on the behavior of ants have natural advantages in the application of cluster analysis. Thus, we introduce the ACODS, a modification of the Ant Colony Optimization metaheuristic for DSNSF creation using the clustering approach.

In this paper, we assume that the edges of $\mathcal{G}$ are formed between the center of a cluster (centroid) and each element that will be clustered. ACODS runs iteratively and in each iteration, an agent constructs a solution. Although each ant has the ability to build a viable solution as well as a real ant can somehow find a path between the colony and food source, the highest quality solutions are obtained through cooperation between individuals of the whole colony [17].

Algorithm 1 shows the steps executed by ACODS for DSNSF creation. These activities are classified into 3 categories:

Build solutions: This step consists of the movement of ants concurrently and asynchronously by the states of the problem. It is determined by moving agents from one node to another neighbor in the graph structure.

Algorithm 1 – ACODS used for DSNSF creation

Require: Set of bits and packets, number of clusters
Ensure: X: Vector representing the normal behavior for bits and packet sets of a day arranged in 288 intervals of 5 minute, i.e. the DSNSF

```
for i = 1 to 288 do
    for t = 1 to number of iterations do
        Create solution
        Evaluate solutions through the objective function (1)
        Update the pheromone trail
    end for
    Calculate the center of each cluster of the best solution found
    X_i ← Average among the clusters
end for
return X
```

Local Search: It aims to test and evaluate solutions created by ants through a local search. In our model, we use the objective function to measure how good are the solutions built by the ants.

Pheromone Update: The pheromone trails are modified in this process. The trails' values can be incremented (when ants deposit pheromones in the edge or connections between the used nodes) or can be decremented. The increased concentration of pheromone is an essential factor in the algorithm implementation, since it directs ants to seek new locations more prone to acquire a near-optimal solution.

IV. CREATING A DSNSF-GA

To create our DSNSF-GA, our data were separated in files, one per day. Every file has 86400 lines, which corresponds to the amount of bits and packets per second in each line. As we choose to generate the DSNSF-GA using data from every five minutes, we selected 300 lines to generate a single point in the graphic.

These lines were divided and later grouped in clusters according to Euclidean distance. Using the Silhouette method for interpretation and validation of clusters [18], best results were reached using $K = 3$ and the GA were used to optimize these distribution among the clusters. Each cluster has its centroid, which is the center of its cluster. As we have three centroids, we calculate the average among those three clusters, which in turn should be the point allocated in the middle of these clusters. At this stage we obtained one point in the DSNSF-GA.

The operation of DSNSF-GA is shown in Algorithm 2. The chromosomes are vectors, and they contain the value of the cluster centroids. The length of a chromosome is defined by $N * K$, where N is the number of dimensions of our search space and K is the number of clusters. As we have three clusters, we distribute flow data among those clusters. Each cluster has its centroid and these values will determine a single gene in the chromosome.

For the initial population, we have generated randomly forty chromosomes, and their values should be between the minimum and maximum flow data values. These chromosomes are used to generate new populations of individuals. The next action is to determine the fittest individuals in a population.

 ISBN: 978-1-61208-376-6

Algorithm 2 – using GA to create the DSNSF

Require: Set of bits and packets collected from historic database, number of: clusters, initial population and iterations

Ensure: X: Vector representing the bits and packets of a day arranged in 288 intervals, which means 5 minutes, i.e. the DSNSF

1: **for** $i = 1$ to 288 **do**
2: Initialize population ρ
3: **for** $t = 1$ to number of iterations **do**
4: Compute fitness for ρ
5: Select the fittest using the Roulette Wheel
6: Apply crossover to generate a new population τ
7: Apply mutation if necessary
8: Evaluate solutions through the *objective function* (1)
9: **end for**
10: Calculate the center of each cluster of the best solution found (fittest)
11: $X_i \leftarrow$ average among the clusters
12: **end for**
13: **return** X

As previously described, we used a Roulette Wheel technique to determine the best chromosomes. What will determine if an individual is or is not fitted, is the shorter distance between all points and their respective cluster centroid. If this distance is lower in an individual than in others, it means the data inside that cluster are well organized, i.e., there are more points closer of its central point in a cluster than in others. In our approach, we used the sum of the three clusters distance to determine the fittest individuals which will procreate.

To yield new generations, individuals must mate between each other. As in nature, the fittest individuals have a greater probability of generating a new offspring, who will generate new ones and so on. When two parents generate a new individual, they will combine their genetic material to a new progeny. This probabilistic process of breeding a new population and combining genetic material is the crossover. This is the key process in a Genetic Algorithm based search, because at this moment the exchange of genes will occur and our population will be diversified. Since two parents reproduce, they will switch genetic material and their children will be a copy of both merged chromosomes. It will assure the population diversity. For our purpose, the exchange of chromosomes will improve the solution, where we are finding the shorter total distance in a chromosome. An important part of the crossover process is to define the crossover point, as we have to choose between a single point crossover or a multi point crossover. A single point crossover is indicated when there is a larger population and we choose that technique to divide our chromosomes. This single point is a random point which divides two chromosomes in four (each of them in two), and combines a couple of those to generate a new one [19].

For a initial population of ρ individuals, we choose $\tau = \rho/2$ corresponding to the fittest in these population to generate a new population. These new population will be mixed to the previous ones and will breed others. The Roulette is span to choose τ, which will generate other children. Each one of these iteration is called generation. Since there is no ideal stopping criteria, usually a fixed iteration number is used. The process

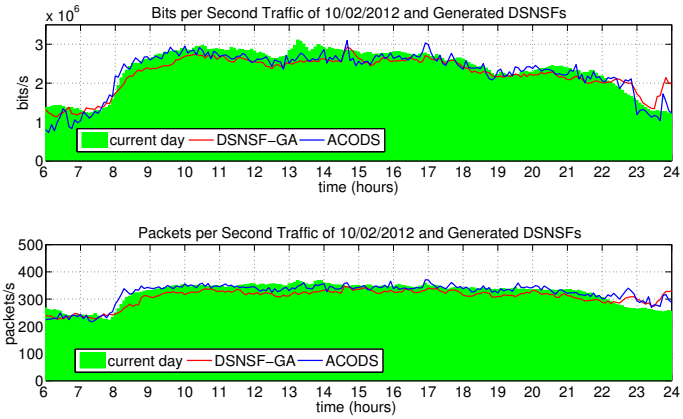


Figure 2. DSNSF-GA and ACODS for 2nd of October

of generating new populations will repeat for a fixed number of times which we have set to sixty [20].

Each chromosome undergoes a mutation probability which is a fixed number. Mutation allows the beginning and the preservation of genetic variation in a population by introducing other genetic structures modifying genes inside the chromosome. We establish a mutation tax of 5%. When the mutation occurs we choose a mutation point, called MP, which will be the point where we are going to change its value. This new value is calculated using $New_i = Old_i + (\delta \times Old_i)$, where New_i is the new individual, Old_i is the old individual, δ is the randomic number $0 < n < 1$ which determine if mutation will or will not occur. The new chromosome will be used to generate a new offspring.

At the end of all these processes we obtained the best population. From this, we choose the best individual, which is the chromosome with the shortest sum of distance between each point in the cluster and its respective centroid. We calculate now the mean among the three cluster centroids. This number will represent a unique point in the DSNSF-GA, as we choose to illustrate a five minute interval in the graphic.

The process of acquiring each value will be repeated for other 288 times to create a graph representing one day in a week. By using data from three days to generate this single point, we now have a network signature of them, or the DSNSF-GA.

V. Tests and Results

We used a real data set for DSNSF creation. These data were collected from State University of Londrina (UEL), and exported using sFlow pattern. Afterwards, these flows are divided in files containing 86400 lines, where each line has the value corresponding to one second in a day. One file is for packets and another one is for bits. As we have two methods to compare, it is important to emphasize that we have set the same number of clusters and iterations for both, ACODS and DSNSF-GA.

In Proença *et al.* [6], a DSNS was created using data of each day based on the history of its previous weeks. This technique was also discussed by Assis *et al.* [21] and Lima *et al.* [22]. Each of them have used a Digital Signature of Network Segment to represent the network behavior efficiently. For our purpose, we used only data between 06:00 and 24:00

 ISBN: 978-1-61208-376-6

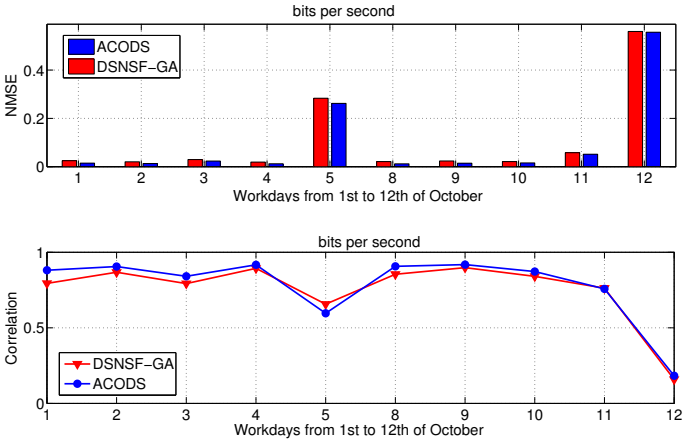


Figure 3. NMSE and CC for bits per second for DSNSF-GA and ACODS

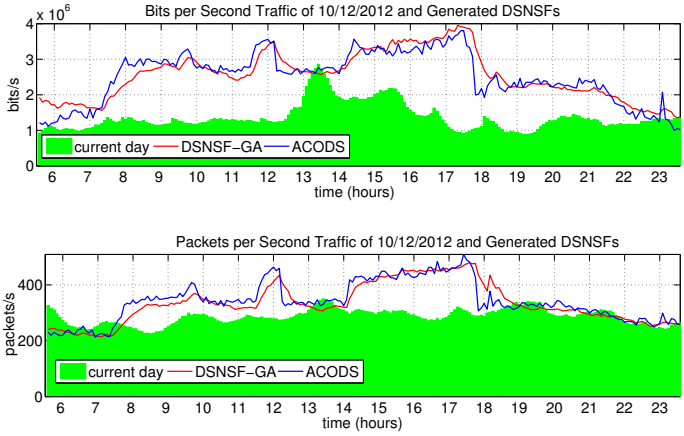


Figure 5. DSNSF-GA and ACODS for 12th of October (national holiday)

since we utilized data from a University and their working hours are between 07:00 and 23:00. It is important to inform that the 12th of October is a national holiday in Brazil, and this is the reason for a different traffic behavior during that day. We decided to keep this day to demonstrate the ability of adaptation of the methods to similar situations.

Figure 2 shows the observed traffic of both, DSNSF-GA and ACODS methods for bits and packets per second. The figure represents the interval described before, October 2nd, 2012, where the green color is the current day, and there are two lines, one for DSNSF-GA and another for ACODS. As shown by this figure, both of them are able to characterize network traffic, displaying the usual observed traffic, the increase and decrease in usage following the same pattern, and also a greater use of network resources during the periods from 07:00 to 12:00 hours and from 13:00 to 23:00 hours.

To measure the accuracy of each method on DSNSFs generation, we decided to adopt two different techniques: Correlation Coefficient (CC) and Normalized Mean Square Error (NMSE).

NMSE compares the mean of a series against certain predicted values. If the NMSE is less than 1, then the forecasts

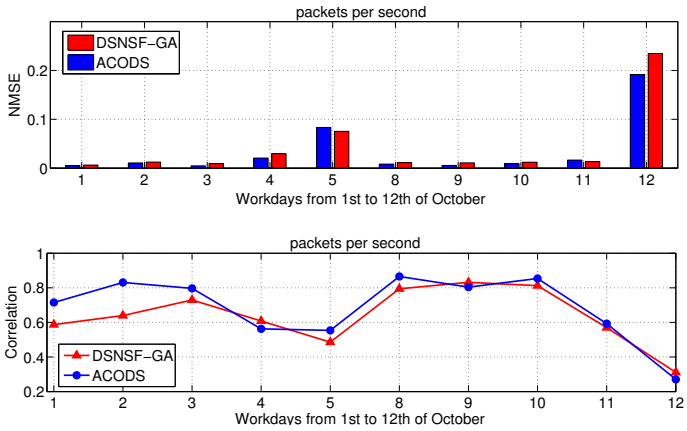


Figure 4. NMSE and CC for packets per second for DSNSF-GA and ACODS

are doing better than the current traffic, but if it is greater than 1, then the predictions are doing worse than the current traffic. The CC measures how suitable a model is, resulting in values varying from -1 to 1. A positive value indicates total correlation, and values close to 0 mean bad or no correlation between data and the adjustable model.

Figure 3 indicates that there is correlation between the observed traffic and the predictions generated by the characterization from DSNSF, as the values are close to 1, both in DSNSF-GA and ACODS. The 12th of October has a bad correlation, and we can see in Figure 5 that the traffic was lower than predicted. This day was a national holiday and there was a little activity at the University. Also in Figure 3, we observe values up to 0.7, meaning that the real traffic and the DSNSF are correlated, except on 12th of October again and on 5th of October.

We investigated what the cause was for the bad correlation 5th of October. The network administrator discovered, analyzing log files, that there was an input HTTP traffic in large scale, caused by students applying for new classes of postgraduate courses. The University was offering 53 new classes that semester, and it was not only the first day but also the only way to apply for a class via the Internet. Figure 4 shows NMSE and CC for packets per second with the same results, indicating that both methods present good error rates, achieving averages NMSE values of 0.4 including the national holiday, and less than 0.1 excluding it. In addition to this, we get an average correlation of 0.75 with the holiday and 0.8 without it.

Both methods are able to characterize network traffic efficiently, as we can see a small difference between the predicted traffic and the real traffic in a normal day. Although we have no threshold to distinguish anomalous from normal behavior, we can see in figures that when good values were observed for CC and NMSE , a normal traffic pattern was also observed. Abnormal values, after investigation, were found to be derived from anomalous traffic.

A. *Computational complexity*

The methods computational complexity are presented as asymptotic notation based on amount of executed instructions. Initially, the ACODS algorithm partitions a set E of data by

 ISBN: 978-1-61208-376-6

K centers of N dimensions, resulting in $O(EKN)$. Using the population of ants to assist the search of the best centers for the collation of data and, as all the ants are compared with each other in pursuit of the final solution, a quadratic complexity is added, ensuring $O(EKNM^2)$. Taking the number of iterations T into account as stopping criterion of the algorithm, we have a final complexity of $O(EKNM^2T)$. Although a maximum of interactions T is defined, ACODS quickly converges to a solution.

Selection operator is executed by the Roulette Wheel, one of the simplest and most traditional methods. As the choice of the roulette slices is done by a linear search from the beginning of the list, each selection requires $O(\rho)$ because, on average, half of the list will be searched. In general, the roulette method uses $O(\rho^2)$ steps, since an offspring will be bred from the crossover between ρ parents. Crossover and mutation operations present linear behavior of $O(\rho)$. All these processes are executed for the N dimensions (bits and packets). Thereby, the activities number performed by DSNSF-GA during iterations are given by $O(EKNH^2T)$.

VI. Conclusion and future works

In terms of computational complexity, both methods use meta-heuristics algorithms to find an optimal solution.There are a number of iterations until a certain condition is reached, and both of them used the same value.

The DSNSF-GA method, introduced in this paper, uses the Genetic Algorithm technique to improve data clusterization and it characterizes network traffic using data collected by sFlow. To estimate network traffic, we organize data simulating the natural evolution process of the nature. Using natural operators like selection of the fittest, crossover and mutation we can obtain the best individuals in a population. We used the shortest distance among each point in a cluster and their respective centroid to determine who are the fittest, which represents a single point in the DSNSF-GA. These digital network signatures can be used, in the future, by network administrators to identify anomalous traffic in their environments, by comparing the real current traffic with the predicted traffic. As previously described, when we identified a flash crowd traffic caused by new classes of postgraduate students, a large input traffic was associated with the beginning of online applications, and the management of network resources could be done automatically.

In future works, we intend to increase the number of dimensions, including new flow data. This multidimensional approach will improve traffic characterization, as we will use more detailed information about the network behavior. Also, we plan to develop a model to establish a threshold for the DSNSF, to distinguish anomalous from normal behavior, being possible to identify, in real time, network anomalies.

Acknowledgment

This work was supported by SETI/Fundação Araucária and MCT/CNPq for Betelgeuse Project financial support. Also the authors would thanks São Paulo State Technological College (Fatec Ourinhos).

References

[1] W. Stallings, “Snmpv3: A security enhancement for snmp,” Communications Surveys Tutorials, IEEE, vol. 1, no. 1, First 1998, pp. 2–17.

[2] B. Trammell and E. Boschi, “An introduction to ip flow information export (ipfix),” IEEE Communications Magazine, vol. 49, no. 4, 2011, pp. 89–95.

[3] A. Lakhina, M. Crovella, and C. Diot, “Characterization of network-wide anomalies in traffic flows,” in Internet Measurement Conference, 2004, pp. 201–206.

[4] A. Patcha and J.-M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” Computer Networks, vol. 51, no. 12, 2007, pp. 3448–3470.

[5] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. Cambridge, MA, USA: MIT Press, 1992.

[6] M. L. Proenca Jr., C. Coppelmans, M. Bottoli, and L. Souza Mendes, “Baseline to help with network management,” in e-Business and Telecommunication Networks. Springer Netherlands, 2006, pp. 158–166.

[7] M. Adaniya, M. Lima, J. Rodrigues, T. Abrao, and M. Proenca Jr., “Anomaly detection using dsns and firefly harmonic clustering algorithm,” in Communications (ICC), 2012 IEEE International Conference on, June 2012, pp. 1183–1187.

[8] M. V. O. Assis, L. F. Carvalho, J. J. P. C. Rodrigues, and M. L. Proenca Jr., “Holt-winters statistical forecasting and ACO metaheuristic for traffic characterization,” in Proceedings of IEEE International Conference on Communications, ICC 2013, Budapest, Hungary, June 9-13, 2013, 2013, pp. 2524–2528.

[9] G. Fernandes, A. Zacaron, J. Rodrigues, and M. L. Proenca Jr., “Digital signature to help network management using principal component analysis and k-means clustering,” in Communications (ICC), 2013 IEEE International Conference on, June 2013, pp. 2519–2523.

[10] L. F. Carvalho, A. M. Zacaron, M. H. A. C. Adaniya, and M. L. Proenca Jr., “Ant colony optimization for creating digital signature of network segments using flow analysis,” in 31st International Conference of the Chilean Computer Science Society, SCCC 2012, Valparaíso, Chile, November 12-16, 2012, 2012, pp. 171–180.

[11] U. Maulik and S. Bandyopadhyay, “Genetic algorithm-based clustering technique,” Pattern Recognition, vol. 33, no. 9, 2000, pp. 1455–1465.

[12] J. Xiaopei, W. Houxiang, H. Ruofei, and L. Juan, “Improved genetic algorithm in intrusion detection model based on artificial immune theory,” in Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium on, Jan 2009, pp. 1–4.

[13] H. Guo, W. Chen, and F. Zhang, “Research of intrusion detection based on genetic clustering algorithm,” in Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on, April 2012, pp. 1204–1207.

[14] M. Mitchell, An introduction to genetic algorithms. MIT Press, 1998.

[15] M. Dorigo, G. D. Caro, and L. M. Gambardella, “Ant algorithms for discrete optimization,” Artificial Life, vol. 5, 1999, pp. 137–172.

[16] H. Jiang, Q. Yu, and Y. Gong, “An improved ant colony clustering algorithm,” in Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on, vol. 6, oct. 2010, pp. 2368 –2372.

[17] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” Computational Intelligence Magazine, IEEE, vol. 1, no. 4, nov. 2006, pp. 28 –39.

[18] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” Journal of Computational and Applied Mathematics, vol. 20, no. 0, 1987, pp. 53 – 65.

[19] W. M. Spears and V. Anand, “A study of crossover operators in genetic programming,” in ISMIS, 1991, pp. 409–418.

[20] C. A. Murthy and N. Chowdhury, “In search of optimal clusters using genetic algorithms,” Pattern Recognition Letters, vol. 17, no. 8, 1996, pp. 825–832.

[21] M. V. d. Assis, J. J. Rodrigues, and M. L. Proenca Jr., “A seven-dimensional flow analysis to help autonomous network management,” Information Sciences, vol. 278, no. 0, 2014, pp. 900 – 913.

[22] M. Lima, L. Sampaio, B. Zarpelão, J. Rodrigues, T. Abrao, and M. L. Proenca Jr., “Networking anomaly detection using dsns and particle swarm optimization with re-clustering,” in Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, Dec 2010, pp. 1–6.

 ISBN: 978-1-61208-376-6

DeadDrop-in-a-Flash: Information Hiding at SSD NAND Flash Memory Physical Layer

Avinash Srinivasan and Jie Wu
Temple University
Computer and Information Sciences
Philadelphia, USA
Email: [avinash, jiewu]@temple.edu

Panneer Santhalingam and Jeffrey Zamanski
George Mason University
Volgenau School of Engineering
Fairfax, USA
Email: [psanthal, jzamansk]@gmu.edu

***Abstract*—The research presented in this paper, to the best of our knowledge, is the first attempt at information hiding (IH) at the physical layer of a Solid State Drive (SSD) NAND flash memory. SSDs, like HDDs, require a mapping between the Logical Block Addressing (LB) and physical media. However, the mapping on SSDs is significantly more complex and is handled by the Flash Translation Layer (FTL). FTL is implemented via a proprietary firmware and serves to both protect the NAND chips from physical access as well as mediate the data exchange between the logical and the physical disk. On the other hand, the Operating System (OS), as well as the users of the SSD have just the logical view and cannot bypass the FTL implemented by a proprietary firmware. Our proposed IH framework, which requires physical access to NAND registers, can withstand any modifications to the logical drive, which is accessible by the OS as well as users. Our framework can also withstand firmware updates and is 100% imperceptible in the overt-channels. Most importantly, security applications such as anti-virus, cannot detect information hidden using our framework since they lack physical access to the NAND registers. We have evaluated the performance of our framework through implementation of a working prototype, by leveraging the OpenSSD project, on a reference SSD.**

Keywords*—*Anti-forensics; Covert Communication; Information Hiding; Security; Solid State Drives.

I. Introduction

With IH, a majority of the research has primarily focused on steganography, the concept of hiding information in innocuous existing files. There has also been considerable research on hiding information within file systems. The advent of SSDs, among other things, has also created new attack vectors from the view point of IH. However, little research exists in regards to IH on SSDs. From an IH view point, the combination of simplicity, standardization, and ubiquity of the traditional Hard Disk Drive (HDD) poses a major challenge. The lack of complex abstraction between physical and logical drive, detailed information on the structure of almost all file systems in use, along with open source tools enabling physical access to HDDs as well as analyzing and recovering both deleted and hidden data makes IH on the HDDs futile. This can be noted from the file system-based IH technique proposed in [1], which utilizes fake bad blocks. Although, this sounds like a generic solution since it is specific to file systems and is independent of storage media. In reality, since the mapping of logical blocks to physical flash memory is controlled by the FTL on SSDs, this cannot be used for IH on SSDs. Our proposed solution is 100% filesystem and OS-independent, providing a lot of flexibility in implementation. Note that throughout this paper, the term "physical layer" refers to the physical NAND flash memory of the SSD, and readers should not confuse it with the Open Systems Interconnection (OSI) model physical layer.

Traditional HDDs, since their advent more than 50 years ago, have had the least advancement among storage hardware, excluding their storage densities. Meanwhile, the latency-gap between Random Access Memory (RAM) and HDD has continued to grow, leading to an ever-increasing demand for high-capacity, low-latency storage to which the answer was SSDs. While flash memory has served this purpose for many years in specialized commercial and military applications, its cost has only recently decreased to the point where flash memory-based SSDs are replacing the traditional HDDs in consumer class Personal Computers (PCs) and laptops. Our proposed framework can operate under two different scenarios – 1) A user hides information strictly for personal use; 2) A group of users collude with the SSD as the *DeadDrop* [2], and any user can hide a secret message which can later be retrieved by another user, with the appropriate map-file.

In contrast to traditional HDDs, SSDs introduce significant complexities [3] including: 1) An inability to support in-place data modification; 2) Incongruity between the sizes of a *"programmable page"* and an *"erasable block"*; 3) Susceptibility to data disturbances; and 4) Imposing an upper bound on their longevity due to progressive wear and/or degradation of flash cells. While these complexities are inevitable to provide the expected performance and longevity from SSDs, they also pair well with the notion of IH. These complexities, if exploited effectively, can provide highly secure and robust IH.

Another technique, as noted by Wee [1], is similar to our proposed data hiding methodology, and is to provide the file system with a list of false bad clusters. Subsequently, the file system discounts these blocks when hiding new data, and as such, can be safely used for IH. Nonetheless, in all of the aforementioned techniques, the entire HDD can be easily read and analyzed for the existence of hidden information using both open source and commercial tools. However, SSDs as such have posed to be the biggest hurdle faced by the digital forensics community [4][5]. Hence,

 ISBN: 978-1-61208-376-6

our proposed approach is very robust to detection and/or destruction, depending on the motive of the user.

A. Assumptions and Threat Model

We assume that there is a key exchange protocol as well as a Public Key Infrastructure (PKI) in place and known to all participants including the adversary. Figure 1 captures the basic idea of our IH framework. *Alice* and *Bob* are two users, who wish to exchange secret messages in presence of the adversary *Eve*. With our proposed framework, shown in Figure 1, they can securely exchange secret messages using the SSD as the *DeadDrop*. Additionally, they need to exchange the corresponding map-file generated during the hiding process. While several different approaches exist that can be used during the above steps, we employ the very popular PKI approach in our discussions. The tools for hiding and retrieving secret messages on the SSD are available to both Alice and Bob. If Eve wishes to extract the secret message from the SSD, then she will need both of these – the tool for retrieving the secret message and the corresponding map-file. While obtaining the tool is very hard, it cannot be completely disregarded. On the other hand, obtaining the map-file can be safely ruled out, particularly owing to the strong security properties of the underlying PKI system that strengthens session initiation with end user authentication with the help of digital certificates. If Eve, somehow, gets access to the SSD physically, she might try the following attacks. We discuss our defense mechanisms to these attacks in section VI-A.

Attack-1: Get a complete logical image of the drive using any available disk imaging tool.

Attack-2: Try to destroy the drive by erasing it.

Attack-3: Get a physical image of the entire drive.

Attack-4: Launch a Man-in-the-Middle attack to sniff the map-file, and subsequently apply it to the drive.

For any IH scheme to be effective, below are the two key features expected to be satisfied: 1) Confidentiality of the hidden message; and 2) Integrity of the hidden message. Most importantly, a very critical feature for an effective IH scheme is that it should conceal the very fact that a secret message is hidden. Our proposed framework indeed achieves all of the above three properties. We use Elliptic Curve Cryptography (ECC) algorithms for encryption, decryption, digital signatures, and for key exchange. ECC algorithms are chosen because of the strong cryptographic properties they meet with small keys sizes. As noted by Rebahi et al. [6], a 160-bit ECC key provides equivalent security to RSA with 1024-bit keys.

Our proposed "Hide-in-a-Flash" IH framework for SSDs differs from existing techniques proposed for traditional HDDs in several key aspects that have been discussed throughout this paper. However, we have identified ones that are pivotal to our research and present them in our list of contributions below and in Section III-A.

B. Our Contributions

Our contributions, at the time of this writing and to the best of our knowledge, can be summarized as follows:

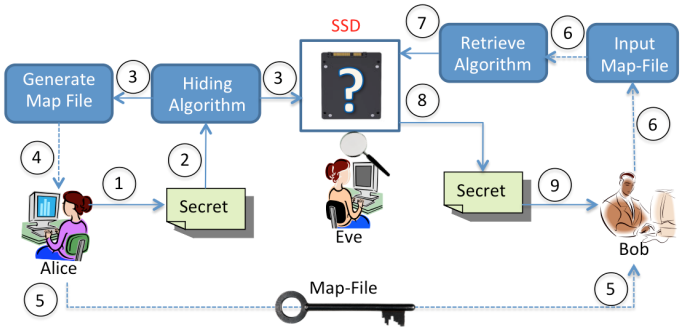


Fig. 1. Hide-in-a-Flash Threat Model.

- This is the first attempt at IH on SSDs at the NAND flash memory physical layer. We have successfully implemented our secure IH technique on the reference *OCZ Vertex Series SATA II 2.5"* SSD. The algorithms used in our framework are *wear leveling* compliant and do not impact the SSD's longevity. Additionally, our implementation of the framework does not affect data integrity and performance of the SSD.
- We have adapted the code from the *OpenSSD* project to bypass the FTL with *Barefoot Controller* firmware, which otherwise completely prevents access to physical flash memory.
- The proposed IH framework is very robust and secure – it can be implemented to be 100% undetectable without prior knowledge of its use, and 100% resistant to manufacturer's updates including destructive SSD firmware updates that completely thwart the proposed IH framework. Most importantly, it is 100% transparent to the user, the OS, and even the FTL.
- Our approach hides information within "false bad blocks" tracked by the FTL in its bad blocks list, thereby preventing any operations on those blocks and making it resistant to updates or overwriting.
- Our framework does not exploit the filesystem's data structure to hide information nor does it hide data in various slack spaces and unallocated space of a drive [7]. Consequently, it does not break the information to be hidden into bits and pieces.
- We have successfully identified functionalities of firmware which are not part of Open-SSDs documentation through firmware reverse engineering. We are working toward making it publicly available through the OpenSSD project website, so that others can continue their research using our information as the baseline.
- Finally, we have designed a tool, by leveraging OpenSSD framework, which can get a physical image of an SSD (with Barefoot flash controller) which we have tested during our evaluations. However, a few minor firmware functionalities are yet to be built into this tool.

 ISBN: 978-1-61208-376-6

C. Road Map

The remainder of this paper is organized as follows. We begin with a review of relevant related works in Section II. In Section III, we provide a background discussion on SSDs, specifically focusing on their departure from traditional HDDs. We also discuss the OpenSSD platform in this section. Section IV investigates various design choices we had to make in designing our system. Later, Section V presents details of the proposed IH framework followed by evaluations methods used and an analysis of the results in Section VI. Finally, in Section VII, we conclude this paper.

II. Related Work

All existing work on IH are proposed for HDDs and nothing specific to SSDs. Those that are for HDDs, the notable ones revolve around hiding information within existing file systems within slack space and unallocated space. Verhasselt [8] examines the basics of these techniques. Another technique, as noted in [1], is similar to our proposed data hiding methodology, and is to provide the file system with a list of false bad clusters. Subsequently, the file system discounts these blocks when hiding new data, and as such can be safely used for IH. Nonetheless, in all of the aforementioned techniques, the entire HDD can be easily read and analyzed for the existence of hidden information using both open source and commercial tools. However, SSDs as such have posed to be the biggest hurdle faced by the digital forensics community [4][5]. Hence, our proposed approach is very robust to detection and/or destruction, depending on the motive of the user.

According to McDonald and Kuhn [9], cryptographic file systems provide little protection against legal or illegal instruments that force data owners to release decryption keys once the presence of encrypted data has been established. Therefore, they propose *StegFS*, a steganographic file system, which hides encrypted data inside unused blocks of a Linux *ext2* file system.

RuneFS [10] hides files in blocks that are assigned to bad blocks *inode*, which happens to be *inode 1* on *ext2*. Forensic programs are not specifically designed to look at bad blocks *inode*. Newer versions of *RuneFS* also encrypt files before hiding them, making it a twofold problem. On the other hand, *FragFS* [11] hides data within Master File Table (MFT) of an New Technology File System (NTFS) volume. It scans the MFT table for suitable entries that have not been modified within the last year. It then calculates how much free space is available and divides it into 16-byte chunks for hiding data.

Khan et. al. [12] have applied steganography to hard drives. Their technique overrides the disk controller chip and positions the clusters according to a code, without which, hidden information cannot be read. In [13], authors propose a new file system vulnerability, *DupeFile*, which can be exploited for IH. The proposed approach hides data in plain sight in the logical disk by simply renaming malicious files with the same name as that of an existing good file. Since the renaming is done at the raw disk level, the OS does not complain to the end user of such file hiding. In another IH method, in [14], authors propose information hiding in file slack space. This technique, called HideInside, splits a given files into chunks, encrypts them, and randomly hides them in the slack space of differ files. The proposed technique also generates a map-file that resides on a removable media, which will be used for retrieval and reconstruction of the randomly distributed encrypted chunks.

In [15], Nisbet et al. analyze the usage of TRIM as an Anti-Forensics measure on SSDs. They have conducted experiments on different SSDs running different operating systems, with different file systems to test the effectiveness of data recovery in TRIM enabled SSDs. Based on their experiments it can be concluded that, with TRIM enabled, Forensic Investigators will be able to recover the deleted data only for a few minutes from the time TRIM was issued.

Wang et. al. [16] have successfully hidden and recovered data from flash chips. Here, authors use the term "flash chips" to refer to removable storage devices like USB flash drives. They use variation in the program time of a group of bits to determine if a given bit is a 0 or a 1. They convert the data to be hidden into bits, and determine the blocks required. Authors have come up with a method to program a group of bits overcoming default page-level programming. While their method is quite robust, it suffers from a significant downside, which is the amount of information that could be hidden. Their method can hide up to 64 MB of data on a 32 GB flash drive, while our proposed IH can hide up to 2 GB of information on a 32 GB SSD, which is an increase in hiding capacity of the channel, by a factor of 16.

III. SSD Background

In contrast to the mechanical nature of the traditional HDDs, an SSD is more than just a circuit board containing numerous flash memory packages and a controller to facilitate the interface between the OS and the physical flash memory. SSDs may utilize either the NOR flash or the NAND flash memory. As the latter is relatively cheap it is highly used for consumer SSDs.

A. Salient Features

Below is a list of salient features of SSDs:

1. Flash Memory: At the lowest level, each flash memory package contains thousands of cells, each capable of holding one or more bits. While read and write operations on a cell are relatively fast, physical limitations imposed by the storage medium necessitate cell erasure before overwriting it with new information. Flash memory cells are logically grouped into pages. A page is the basic unit of reading and writing. Pages are grouped into blocks, which is the basic unit of erasure. Blocks are grouped into dies, and dies are grouped into flash memory packages, aka banks. Within a SSD, multiple flash memory packages are grouped to provide the total capacity of the drive.

2. Flash Translation Layer (FTL): In order to manage the complexities of the physical layout, optimize the use and endurance of the flash memory, and provide the OS with the traditional block device interface of storage devices, SSDs contain a controller which implements an additional layer of abstraction beyond traditional HDDs

 ISBN: 978-1-61208-376-6

TABLE I. REFERENCE SSD OCZ VERTEX SPECIFICATION

Total number of banks	8	Dies per Bank	2
Blocks per Die	4096	Pages per Block	128
Cell Type	2-level cells	Cells per Page	17,256
Bits per Cell	34,512	Total Size	32 GB
Advertised capacity	30 GB	Over-provisioning	2 GB

known as the FTL. Below are the three fundamental operations of the FTL – 1) *logical to physical block mapping*; 2) *garbage collection*; and 3) *wear leveling*.

3. Pages Size, Spare Bytes & Error Correction: Traditional HDDs implement storage based on a predefined allocation unit called a *sector*, which is a power of two. To facilitate the mapping of logical blocks to physical flash memory, the flash memory is manufactured with page sizes also being powers of two. However, since flash memory is susceptible to data disturbances caused by neighboring cells, it is critical for the FTL to implement an error correction mechanism. To accommodate the storage requirements of the Error Correction Code (ECC), the flash memory is manufactured with additional spare bytes, in which FTL can store the ECC. For instance, a flash memory page may consist of 8192 bytes with 448 additional bytes reserved for ECC.

4. Bad Blocks: Due to the physical characteristics of the flash memory as well as cell degradation over time, flash memory packages may ship with blocks that are incapable of reliably storing data, even with an ECC employed. These blocks are tested at the factory and marked in a specific location within the block to identify them as initial bad blocks. During SSD manufacturing, the flash memory is scanned for bad block markings and an initial bad block list is stored for use by the FTL. Beyond this initial list of bad blocks, the FTL must keep the list updated with the inclusion of newly identified bad blocks at runtime.

5. Over-Provisioning: Write amplification, a serious concern with SSDs, is an inevitable circumstance where the actual amount of physical information written is greater than the amount of logical information requested to be written. On SSDs, this occurs for several reasons, including but not limited to: *need for ECC storage*, *garbage collection*, and *random writes to logical blocks*. In order to maintain responsiveness when the drive is near capacity and longevity when flash memory cells begin to fail, SSDs may be manufactured with more flash memory than they are advertised with, a concept known as over-provisioning. For example, an SSD containing 128GB of flash memory may be advertised as 100GB, 120GB, or with 28%, 6.67%, or 0% over-provisioning, respectively.

B. OpenSSD

The *OpenSSD Project* [17] was created by *Sungkyunkwan University* in Suwon, South Korea in collaboration with *Indilinx*, to promote research and education on SSD technology. This project provides the firmware source code for the *Indilinx Barefoot Controller* used by several commercial SSD manufacturers including *OCZ*, *Corsair*, *Mushkin*, and *Runcore IV*. The firmware code provided in this project is an open source implementation, and a version of research implementation of a complete SSD known as *Jasmine Board*, is available for purchase.

Table I summarizes the specifications of the reference SSD. During the course of our research, we learned that the *Jasmine Board* uses the same Indilinx Barefoot controller firmware as our reference SSD, which is an *OCZ Vertex Series SATA II*. We also learned that the firmware installation method used by the *OCZ Vertex* SSD. Furthermore, the *Jasmine Board* involved setting of a jumper on the SSD, to enable a factory or engineering mode. Upon setting the jumper on the reference SSD and compiling and running the firmware installation program adapted from the OpenSSD Project, we were able to connect to the SSD in factory mode with physical access to NAND flash memory chips, bypassing the FTL.

IV. Framework Design Choices

We have successfully identified the following critical pieces of information from the OpenSSD code about firmware functionalities through reverse engineering, information which was otherwise not available on OpenSSD documentation:

- Block-0 of each flash memory package is erased and programmed during the firmware installation process.
- First page of block-0 contains an initial bad block list before the firmware installation, which will be read by the installer, erased along with the remainder of block-0, and programmed with identical information as part of the installation process.
- In addition to page-0, a minimal set of metadata such as the firmware version and image size is programed into pages-1 through 3, and the firmware image itself is programmed into sequential pages starting with page-4.

Based on our analysis of the firmware, we came up with the following storage covert channels that can be used in designing our IH framework.

1) *Manipulating the FTL data structure:* We considered the possibility of modifying the firmware and utilizing it for IH. One possibility was to redesign the wear leveling algorithm such that some blocks will never be considered for allocation.
2) *Utilizing the spare bytes:* All spare bytes available per page are not completely used. Some of the spare bytes are used for storing ECC. Thus the remaining spare bytes can be used for IH.
3) *Using the blank pages in block zero:* Only a few pages of block zero were used during firmware installation; the remaining were free for IH.
4) *Manipulating the initial bad block list:* By inserting new bad blocks in the bad block list in block zero, and using them for IH.

With due diligence, we decided against the first three methods because OpenSSD's implementation is a stripped down version of the actual firmware. This means, the full-blown version of the firmware could easily overwrite

 ISBN: 978-1-61208-376-6

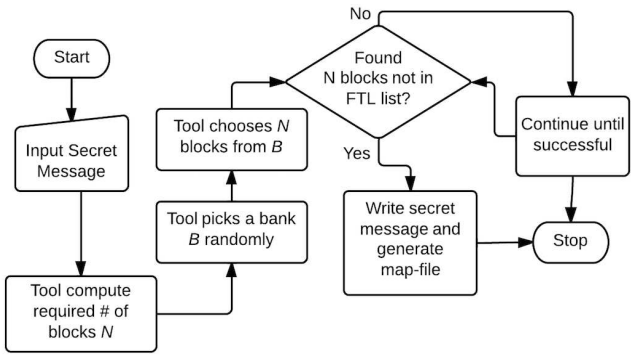


Fig. 2. Flow chart illustrating the initial system design.

any modifications we make to the FTL data structure. Therefore, the first method is not very useful. Though the unused spare bytes can be used, it was uncertain whether or not the firmware would use them during the life of the SSD. Hence, the second method was decided against. As with the blank pages on block 0, they did not provide much room for hiding information because of which third method was ruled out. Finally, we had narrowed down our choice to one stable and robust method – *manipulation of the initial bad block list*, details of which follow in the next section.

V. Information Hiding Framework

A. Process Overview

Scenario: Alice and Bob could be friends or total strangers communicating over an insecure channel. Their goal is to exchange secret messages over the insecure channel in the presence of Eve, the adversary. As noted in Section I-A, we will not discuss the details of key negotiation as plenty of known works exist on this subject. For simplicity, we assume the use of PKI in our discussions.

Step-1: Alice has the secret message M_{sec} she wishes to share with Bob.

Step-2: She generates a random session key K_{rand}^{Alice} that she inputs to the Hiding Algorithm along with M_{sec}.

Step-3: M_{sec} is encrypted with K_{rand}^{Alice} generating the following message:

$$E[M_{sec}]_{K_{rand}^{Alice}} \tag{1}$$

This message is then written into fake bad blocks. Simultaneously, a map-file F_{map} is generated. The purpose of F_{map} is identifying blocks holding the secret message.

Step-4: Alice then encrypts F_{map} and K_{rand}^{Alice} with her private key K_{prv}^{Alice} generating the following message. This is necessary to provide Bob with a message integrity verification service.

$$M_{sec}^{verify} = E[F_{map}||K_{rand}^{Alice}]_{K_{prv}^{Alice}} \tag{2}$$

She then encrypts the M_{sec}^{verify} message with Bob's public key K_{pub}^{Bob}, generating the following message.

$$_{conf}M_{sec}^{verify} = E[(F_{map})||(K_{rand}^{Alice})]_{K_{pub}^{Bob}} \tag{3}$$

```
1: file ⇐ user input secret message
2: blocksRequired = sizeOf(file) / sizeOf(block)
3: leastBadBlockBank = 0
4: for i = 0 to bank.count do
5:     if (i.badBlocksEraseCount <
6:         leastBadBlockBank.badBlocksEraseCount)
   then
7:         leastBadBlockBank = i
8:     end if
9: end for
10: while (leastBadBlockBank.blocksinbadBlockList)
   &&
11: (leastBadBlockBank.blocks.metadata==keyword)
   &&
12: (count < blocksRequired) do
13:     newBadBlock.count = leastBadBlockBank.block;
14:     count++
15: end while
16: Payload = Encrypt(metadata, file)
17: payload.Write()
18: newKey = encode(leastBadBlockBank, newBadBlock)
```

Fig. 3. Algorithm for Hiding

The message $_{conf}M_{sec}^{verify}$ encrypted with Bob's public key provides confidentiality service for message exchange. Note that, the use of encryption keys in this specific order also provides communication endpoint anonymity.

Step-5: Alice sends $_{conf}M_{sec}^{verify}$ to Bob. On receiving this message, Bob uses his private key K_{prv}^{Bob} to decrypts the message extracting M_{sec}^{verify}. Then, Bob uses K_{pub}^{Alice} to extract the F_{map} and K_{rand}^{Alice}. Note that Alice and Bob could use either a client-server or P2P architecture to eliminate the need for physical access to the SSD.

Steps-6 & 7: Bob extracts F_{map} and K_{rand}^{Alice}. He inputs F_{map} to the Retrieving algorithm, presented in Figure 4, which applies it to the SSD to retrieve and reconstruct the encrypted secret message $E(M_{sec})_{K_{rand}^{Alice}}$.

Steps-8 & 9: Bob uses K_{rand}^{Alice} to decrypt $E(M_{sec})_{K_{rand}^{Alice}}$ and finally extracts the secret message M_{sec}.

B. Initial Design

In the first phase of our research, we modified the open SSD framework to our specific requirements and tried to hide a test file. As illustrated in the flowchart in Figure 2, we designed a simple tool with a command line interface that receives *filename* as input from the user. Subsequently, the tool decides the number of bad blocks to be allocated for hiding that file based on the file size. Finally, the tool chooses a random bank on the SSD and allocates the required number of blocks. While allocating the blocks, we made sure that the blocks are not part of the initial bad block list the SSD was shipped with. If the allocation is successful, copy the file to the specified blocks and create the map-file (used to identify the bank and block). The map-file is used for the retrieval process.

C. Challenges with the Initial Design

In this section, we address some of the challenges we face with the initial design of our IH framework.

 ISBN: 978-1-61208-376-6

```
1: map-file ⇐ file received from sender
2: bankAndBlock = decode(map-file)
3: metadata = bankAndBlock.Read()
4: decrypt(metadata)
5: decrypt(file.Read())
6: file.Write()
7: if then(ReadandErase)
8:     Erase(bankAndBlock)
9:     eraseCount++
10:    eraseCount.Write()
11: end if
```

Fig. 4. Algorithm for Retrieving

1) As the number of bad blocks increase, firmware installation fails, rendering the drive useless.
2) If we hide data in blocks that hold the firmware, then firmware reinstallation would rewrite these blocks, irrespective of their inclusion in the bad block list.
3) As part of experiment, we did a complete physical image of drive, including the bad blocks, and were able to find that the hidden files signature was visible along with the metadata.
4) Every time we added a new bad block and hid data, we had to reinstall the firmware. This was required because the firmware would only keep track of the bad blocks that were in the list when the firmware was installed.

D. Enhanced design

We shall now discuss our enhanced design with improvements to overcome the challenges of the initial design as delineated above.

- Uninstall the SSD firmware.
- Enable the user to specify the banks on which bad blocks have to be assigned and the number of bad blocks to be allocated on each bank.
- Have the tool allocate the user specified number of blocks on user specified banks and append these blocks to the bad block list maintained by the FTL.
- Reinstall the firmware on the SSD.
- Add metadata to user-created bad blocks to distinguish them from firmware identified bad blocks. In our prototype implementation of the IH framework on the reference SSD, we reserve the very first byte of user-specified bad blocks to keep track of its erase count, which serves as the metadata. The erase count variable is initialized to 0, and is incremented every time new data is written to the corresponding block, since write operation on as SSD is preceded by an erase operation.

Note that, as shown in the Table IV, the first byte is the erase count which is followed by the data. This helps to select the least-erased block every time we pick a block for hiding, and make sure the block is empty.

As can be seen in Figure 3, we pick blocks from banks that have the least erase count. We achieved this by keeping track of the cumulative erase count, comparing the cumulative erase count of all the bad blocks in the banks, and finally pick the one with the least value. Next, in order to address the firmware overwrite problem, we started excluding the first 32 blocks (This was done during the pre-allocation of bad blocks) in every bank. Finally, in order to escape from the physical imaging, we started to encrypt both the metadata and the file. While retrieving the file, we gave an option for the user to erase and retrieve the file or just retrieve the file alone. If user chooses to erase and retrieve the file, we erased the block and increased the erase count by one such that the block was not used until all the other bad blocks have reached a similar erase count.

TABLE II. Information retrieval under different scenarios.

Condition	Hidden File Retrieved
Firmware Reinstallation	Yes
NTFS Formatted Drive	Yes
Partitioned Drive	Yes
Populate Disk to Full Capacity	Yes
Factory Mode Erase	No

VI. Evaluation of Enhanced Design

We confirm through evaluations that our framework is 100% undetectable and robust to firmware updates.

Experiment-1: We test the conditions under which the secret message is retained and retrievable. Table II summarizes the different scenarios under which we evaluated our framework on the reference SSD. As can be seen, we were able to retrieve the secret message in every scenario except when erased in the factory mode. However, this operation requires access to the SSD in factory mode as well as knowledge of factory commands specific to the firmware in use, without which it is impossible to wipe the physical memory of the SSD.

Experiment-2: With this experiment, our objective was to determine the maximum number of blocks that can be tagged as bad, both firmware-detected and user-specified, before the firmware installation starts to fail. This would give us the total amount of data that can be hidden safely, using our IH framework, without causing drive failure. We also wanted to know if hiding data would result in any changes, as perceivable by a normal user. For this, we gradually increased the bad block count in each bank, in increments of 10. With every increment, we did the following – 1) increment the counter tracking the bank's bad block count ; 2) re-install the firmware; 3) install the file system on top of the firmware; and 4) check the available logical disk space. During the experiments, we determined that the threshold for number of blocks, as a fraction of the total number of blocks per bank, that can be tagged as bad, is approximately 2.6% per bank. This is equivalent to 218 blocks per bank. Beyond this, the firmware installation fails. We have summarized the results in Table III. Furthermore, based on the results, we conclude that bad block management is a part of over-provisioning and hence, a typical user won't notice any changes to the logical structure of the disk when information is hidden, proving that our system is 100% imperceptible by end users.

Experiment-3: Finally, we wanted to test if any of the existing computer forensic tools would be able to discover the

 ISBN: 978-1-61208-376-6

TABLE III. Drive size with different bad block count.

Bad Block Count	Drive Size
25	29.7GB
50	29.7GB
75	29.7GB
100	29.7GB
109	29.7GB

secret messages hidden on an SSD using our IH framework. We used freeware tools like WinHex and FTK imager. Both the tools, though very popular and powerful, were unsuccessful in getting past the logical disk. We confidently conclude that none of the existing computer forensics tools, at the time of this writing and to the best of our knowledge, have the capability to access the physical layer of an SSD.

TABLE IV. Manipulated bad block layout.

Number of Bytes	Information
1	Erase Count
Remaining bytes	Hidden data

A. Defense against attacks

In Section I-A, we presented four possible attacks that an adversary, Eve, can potentially try against our IH framework. We shall discuss why none of these attacks will be successful against our IH framework.

- ***Attack-1 Defense:*** The hidden information is not part of the logical drive. Hence, Eve will not benefit from a logical image of the DeadDrop SSD.
- ***Attack-2 Defense:*** SSD blocks that are tracked as bad blocks by the FTL firmware are never accessed and erased by the firmware. Hence, this attack will not be successful.
- ***Attack-3 Defense:*** Currently, it is impossible for Eve to make a physical image of the SSD without our modified OpenSSD software. Additionally, Eve should have the ability to access the SSD into factory mode with appropriate jumper settings, and should know the firmware functionalities that we have identified beyond those provided in the OpenSSD documentation. Beyond this, she would still need the random session key generated by Alice that was used to encrypt the secret message. Additionally, she would need Bob's (recipient of the map-file) private key to decrypt the random session key and the map-file, without which the secret message cannot be reconstructed. Therefore, the feasibility of this attack can be safely ruled out.
- ***Attack-4 Defense:*** Assuming Eve is able to sniff the map-file from the traffic between Alice and Bob, as discussed in Section V, she still needs Bob's private key to decrypt the map-file. Bob's private key, however, is not accessible to anyone other then Bob himself. Hence, this attack is ruled out.

VII. Conclusion and Future Work

In this paper, we have presented the design, algorithms, and implementation details of secure and robust IH framework that can hide information on SSDs at the physical layer. We have presented multiple methods for IH highlighting their strengths and weaknesses. Finally, we have evaluated the proposed framework through real world implementations on a reference SSD running *Indilinx's Barefoot flash controller*, which is used by various SSD manufacturers including, *Corsair, Mushkin,* and *Runcore IV* [18]. Consequently, this IH framework can be used on SSDs from different manufacturers, making it quite pervasive and ubiquitous.

The ability to interface SSDs with the OpenSSD platform and bypass the FTL has significant impact on the Digital Forensics community. Also, this is the first step toward potential antiforensics techniques. Having discovered this possibility, law enforcement agencies can now focus on potential information theft and antiforensics attacks on SSDs, which otherwise was deemed near impossible. As part of our future work, we would lim to investigate the potential of integrating more support for other popular proprietary firmware. This will enable to expand the existing project to support forensics investigation of SSDs from a wide array of manufacturers.

Acknowledgment

The authors would like to thank the Defense Cyber Crime Centre, Linthicum, Maryland, USA, for the reference SSD drive used in the experiments.

References

[1] C. K. Wee, "Analysis of hidden data in NTFS file system," 2013, URL: http://www.forensicfocus.com/hidden-data-analysis-ntfs [accessed: 2013-04-25].

[2] "DeadDrop," 2014, URL: http://en.wikipedia.org/wiki/Dead_drop [accessed: 2014-07-26].

[3] L. Hutchinson, "Solid-state revolution: in-depth on how SSDs really work," 2014, URL: http://arstechnica.com/information-technology/2012/06/inside-the-ssd-revolution-how-solid-state-disks-really-work/2/ [accessed: 2014-07-25].

[4] G. B. Bell and R. Boddington, "Solid state drives: the beginning of the end for current practice in digital forensic recovery?" vol. 5, no. 3. Association of Digital Forensics, Security and Law, 2010, pp. 1–20.

[5] C. King and T. Vidas, "Empirical analysis of solid state disk data retention when used with contemporary operating systems," vol. 8. Elsevier, 2011, pp. S111–S117.

[6] Y. Rebahi, J. J. Pallares, N. T. Minh, S. Ehlert, G. Kovacs, and D. Sisalem, "Performance analysis of identity management in the session initiation protocol (sip)," in *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on.* IEEE, 2008, pp. 711–717.

[7] E. Huebnera, D. Bema, and C. K. Wee, "Data hiding in the ntfs file system," vol. 3, 2006, pp. 211–226.

[8] D.Verhasselt, "Hide data in bad blocks," 2009, URL: http://www.davidverhasselt.com/2009/04/22/hide-data-in-bad-blocks/ [accessed: 2009-04-22].

[9] A. D. McDonald and M. G. Kuhn, "Stegfs: A steganographic file system for linux," in *Information Hiding.* Springer, 2000, pp. 463–477.

[10] Grugq, "The art of defiling: Defeating forensic analysis on unix file systems," *Black Hat Conference*, 2005.

[11] I. Thompson and M. Monroe, "Fragfs: An advanced data hiding technique," 2004, URL: http://www.blackhat.com/presentations/bh-federal-06/BH-Fed-06-Thompson/BH-Fed-06Thompson-up.pdfTrueCrypt(2006) [accessed: 2014-6-02].

[12] H. Khan, M. Javed, S. A. Khayam, and F. Mirza, "Designing a cluster-based covert channel to evade disk investigation and forensics," vol. 30, no. 1. Elsevier, 2011, pp. 35–49.

 ISBN: 978-1-61208-376-6

[13] A. Srinivasan, S. Kolli, and J. Wu, "Steganographic information hiding that exploits a novel file system vulnerability," in *International Journal of Security and Networks (IJSN)*, vol. 8, no. 2, 2013, pp. 82–93.

[14] A. Srinivasan, S. T. Nagaraj, and A. Stavrou, "Hideinside – a novel randomized & encrypted antiforensic information hiding," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*. IEEE, 2013, pp. 626–631.

[15] A. Nisbet, S. Lawrence, and M. Ruff, "A forensic analysis and comparison of solid state drive data retention with trim enabled file systems," in *Proceedings of 11th Australian Digital Forensics Conference*, 2013, pp. 103–111.

[16] Y. Wang, W.-k. Yu, S. Q. Xu, E. Kan, and G. E. Suh, "Hiding information in flash memory," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy, S&P'13*. IEEE Computer Society, 2013, pp. 271–285.

[17] "OpenSSDWiki," 2013, URL: http://www.openssd-project.org [accessed: 2013-04-25].

[18] "Barefoot," 2014, URL: http://en.wikipedia.org/wiki/Indilinx [accessed: 2013-10-02].

 ISBN: 978-1-61208-376-6

Saving Privacy in Trust-Based User-Centric Distributed Systems

Alessandro Aldini

Dipartimento di Scienze di Base e Fondamenti
University of Urbino "Carlo Bo"
Urbino, Italy
Email: alessandro.aldini@uniurb.it

***Abstract*—User-centricity is a design philosophy subsuming new models of Internet connectivity and resource sharing, whose development is mainly driven by what users offer and require. To promote user-centric services and collaborative behaviors, incentives are needed that are typically based on trust relations and remuneration. In this paper, we show that privacy-preserving mechanisms can favor user's involvement if privacy can be traded with trust and cost. In particular, we present and evaluate formally a model ensuring an adequate level of flexibility among privacy, trust, and cost in the setting of distributed systems.**

Keywords–Cooperation incentives; trust; privacy; remuneration; user-centric networks; model checking.

I. INTRODUCTION

Nowadays, user-driven services, like personal hotspot and peer-to-peer, are playing a fundamental role in the reshaping of the Internet value chain [1]. Essentially, they focus on the user experience, related needs, expectations, and attitude to cooperation. One of the key factors behind the success of community-scale user-centric initiatives is given by the user involvement as a *prosumer*, i.e., an actor combining the roles of service producer and consumer. Such an involvement must be strengthened through the adoption of incentive mechanisms stimulating the willingness to collaborate. In particular, even if cooperation is a natural consequence of sense of community and synergy, it cannot be taken for granted because of typical obstacles like, e.g., selfishness and, even worse, cheating, which represent a threat keeping users from trusting other community members.

Establishing trust relations among users is the objective of explicit trust and reputation systems, among which we concentrate on those aiming at providing computational estimations of user's trustworthiness as perceived by the community [2]. Basically, these estimations work effectively as an incentive to collaborate if they represent parameters influencing access to services at favorable conditions, among which we include the service cost as one of the most important aspects affecting the perceived quality of experience. At the same time, remuneration is another kind of incentive used to stimulate cooperation [3]. Whenever combined with trust, it enables a virtuous circle for the proliferation of user-centric services.

Trust is a concept that may involve and justify the collection of personally identifiable sensitive information, which in many real situations contrasts dramatically the idea of privacy and plays a deterrent role when users are getting involved in interactions. In particular, the lower the attitude to expose sensitive information is, the higher the probability of being untrusted when negotiating a service. Trading privacy for trust is thus a way for balancing the subjective value of what is revealed in exchange of what is obtained.

The above considerations suggest that an efficient cooperation infrastructure depends on the tradeoff among trust, privacy, and cost. As shown recently [4], these three dimensions can be balanced in order to favor collaborative behaviors depending on specific user's needs in terms of social (e.g., personal sensibility to trust and privacy issues) and economical (e.g., in terms of costs that can be afforded) requirements. More precisely, in the model proposed in [4], a balanced tradeoff is guaranteed by a centralized system in which reputation is managed by a trusted third party (TTP) collecting information about every transaction completed, while keeping the desired level of privacy for every user involved. In this paper, we provide a twofold contribution. On one hand, we show how to implement the model of [4] in the setting of distributed systems that cannot rely on TTP. On the other hand, we validate formally such a model through model checking based analysis [5]. This validation is done in the setting of a cooperation system that has been recently proposed to implement trust and remuneration based incentive mechanisms [6].

In the rest of this section, we comment on related work. In Section II, we briefly recall the model of [4] and then we illustrate a distributed solution for its implementation. In Section III, we estimate the validity of such a model in the setting of a real-world case study. Finally, some conclusions terminate the paper in Section IV.

A. Related Work

Making trust and service cost mutual dependent is a winning strategy if the aim is to stimulate honest behaviors while keeping users from cheats and selfishness [6]–[8], as also proved formally by means of formal methods, like game theory and model checking [9]–[13].

The contrast between privacy and trust is investigated in [14], where it is shown that these two aspects can be traded by employing a mechanism based on *pseudonyms*. In practice, users create freely pseudonyms identified by the so-called *crypto-id*, i.e., the hash of the public key of a locally generated asymmetric cryptography key pair. Then, in different environments, a user can use different pseudonyms to carry out actions logged as events signed with the private key of the chosen pseudonym. If needed to acquire more reputation, several pseudonyms can be linked together in order to augment the number of known actions and potentially increase the trust

 ISBN: 978-1-61208-376-6

in the linked entity. Notice that in approaches such as this one the link is irrevocable.

Incentive mechanisms are proposed in [15] to achieve a balanced tradeoff between privacy and trust in the setting of data-centric ad-hoc networks. In [16], such an interplay is formulated as an optimization problem in which both privacy and trust are expressed as metrics. In [17], trust towards an entity is used to take decisions about the amount of sensitive information to reveal to the entity. Further works on unlinkability [18] and pseudonymity [19] [20] provide insights on the tradeoff between privacy and trust.

With respect to previous work, the novelty of the approach proposed in [4] is twofold. On one hand, the analysis of the tradeoff between privacy and trust takes into account also the service cost. On the other hand, it overcomes the limitations of the existing approaches, in which sensitive information linking is irrevocable and the privacy disclosure is incremental.

II. A Model for Independent Release of Privacy

In a classical view of privacy, a user exposes (part of) personal information in order to be trusted enough to get access to the service of interest. In other words, privacy disclosure is traded for the amount of reputation that the user may need to be considered as a trustworthy partner in some kind of negotiation in which, e.g., service cost may depend on trust. Typically, once different pieces of sensitive information (e.g., credentials, virtual identities, or simply the proof of being the user involved in a transaction previously conducted), say I_1 and I_2, are linked and exposed to be trusted by someone else, then such a link is irrevocably released. In this sense, we say that the disclosure of sensitive information is incremental along time.

In order to exemplify, as discussed in [14], I_1 and I_2 may identify two different transactions conducted by the user under two different pseudonyms, each one revealing different personal information about her. The user is obviously able to show that both I_1 and I_2 are associated with the same user and, if such a proof is provided, I_1 and I_2 become irrevocably linked together. As opposite to this scenario, in [4] an alternative, independent model of privacy release is proposed in which the link is not definitive. In order to work properly, such a model requires some form of uncertainty associated with the owners of specific actions. Basically, this is obtained by sharing pseudonyms among different users. Similarly as in [14], a virtual identity is represented by a crypto-id, which can be calculated using the SHA-3 cryptographic hash function over the public key of the user. Then, the basic idea of the independent model of privacy release is that trust and transactions are mapped to pieces of the crypto-id rather than to the crypto-id as a whole.

Let us explain such a mechanism through a typical handshake between Alice, who issues a service request, and Bob, who offers the service. Instead of revealing to be Alice, she accompanies the request with a portion of her crypto-id identified by applying a bitmask to the crypto-id through the bitwise AND operation. For the sake of presentation, consider a 8-bit crypto-id, e.g., 10010101, from which we obtain the portion 000$\underline{1}$00$\underline{0}$0, called *chunk*, when applying the bitmask 00010010. Hence, a chunk is a subset of bits of the crypto-id, of which we know value and position. Amount and position of 1's occurrences in the bitmask are under Alice's control.

The transaction is then identified by the chunk chosen by Alice, together with the proof (which can be validated in Zero Knowledge) of being a proper owner of the chunk exposed. Therefore, a trust value (and related variation due to the feedback following the transaction execution) is not associated with Alice directly, but is related to the chunk of bits extracted from Alice's crypto-id through the chosen bitmask. In general, the same chunk is potentially shared by other crypto-ids belonging to several different users. In future interactions, Alice may select other chunks of her crypto-id. Moreover, she can also spend a set of different chunks of the crypto-id in order to exploit a combination of the trust levels associated with each of these chunks. Ideally, the overall trust associated with a crypto-id shall result from a combination of the trust values accumulated by every chunk of such a crypto-id spent in previous interactions. Thanks to the uncertainty relating chunks and associated owners, every time Alice exposes a chunk to Bob in order to negotiate a transaction, Bob cannot link the current transaction to any of the previous transactions conducted (by Alice or by other users) by using the same (or a portion of the current) chunk.

While in [4] the model above relies on the presence of a TTP managing chunk's reputation, in the following we tackle the problem of implementing the same idea in the setting of distributed systems without central authority and any prior knowledge about crypto-ids, which represent a more realistic scenario in several user-centric networks.

A. Design for Distributed Systems

Handling trust towards users by tracing the usage of (possibly shared) chunks is a hard task in the absence of a centralized reputation system. To deal with this problem, in order to estimate user's trustworthiness we define a local trust structure that allows any user offering a service to associate a trust value with every chunk received to negotiate the service.

Let $\mathcal{C}$ be the set of chunks with which the user has interacted in completed transactions and $\mathcal{T}$ be the trust domain, which we assume to be numeric and totally ordered. Chunks are ranged over by $C, C', \ldots$. Sometimes, we use the notation C_B to define a chunk identified by bitmask B and $C_B[i]$ (resp., $B[i]$) to denote the value of the i-th bit of the chunk (resp., bitmask). Set $\mathcal{C}$ forms a partially ordered set (*poset*, for short), $(\mathcal{C}, \leq)$, where the refinement operator $\leq$ is defined as follows.

Definition 1 (Chunk refinement): Let n be the crypto-id size. Given chunks $C_B, C_{B'}$, we say that $C_{B'}$ refines C_B, denoted $C_B \leq C_{B'}$, if and only if:

- for all $1 \leq i \leq n$: $B[i] \leq B'[i]$;
- for all $1 \leq i \leq n$: if $B[i] = 1$ then $C_B[i] = C_{B'}[i]$.

Notice that if $C_B \leq C_{B'}$ then B is a submask of B' and the information exposed by $C_{B'}$ includes that revealed by C_B. If two chunks are related through $\leq$ then they could be originated from the same crypto-id. As we will see, maintaining the poset structure provides the means to approximate the trust towards any (possibly unknown) crypto-id by employing the trust related to the potential constituting chunks.

 ISBN: 978-1-61208-376-6

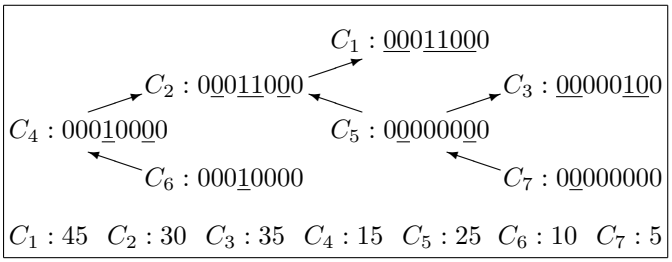


Figure 1. Example of a local trust structure.

Each element of the poset $(\mathcal{C}, \leq)$ is labeled by a value of the trust domain $\mathcal{T}$. Such a value represents the trust of the user towards the related chunk resulting from interactions associated with such a chunk. Formally, we denote such an extended structure with $(\mathcal{C}, \leq, t)$, where $t : \mathcal{C} \rightarrow \mathcal{T}$ defines the mapping from chunks to trust values. Initially, for every unknown chunk C with which the user interacts for the first time, we assume $t(C)$ to be equal to the dispositional trust dt of the user, which represents the attitude to cooperate with unknown users.

Example: Figure 1, which in the following we use as running example, shows the graphical representation of a poset, where, e.g., $C_6 \leq C_4 \leq C_2 \leq C_1$, as well as $C_7 \leq C_5 \leq C_3$, while, e.g., C_6 and C_3 are not related with each other. Moreover, the figure reports also the trust associated with each known chunk at a given instant of time, by assuming the trust domain $[0, 50]$.

To emphasize the nature of the independent model of privacy release, notice that even if Alice invested chunk C_1 in a past interaction with Bob, whose reference trust structure is that depicted in Figure 1, then in the current transaction she may use chunk C_2 only, while Bob cannot infer the link between the user of the past interaction associated with C_1 and the current one. As a side effect, notice also that all the users with a crypto-id matching with C_2 actually benefit from the trust (or pay the mistrust) associated with C_2. ■

The obfuscation mechanism illustrated in the example above, which is crucial for the requirements of the independent model of privacy release, can be viewed as an additional incentive to take collaborative and honest decisions, as a high number of crypto-id chunks highly trusted contribute to increase the probability of obtaining services at a reasonable cost by preserving the desired level of privacy.

A fundamental issue for any trust system is given by the transaction feedback that induces a trust variation influencing the trust $t(C)$ towards the chunk C associated with the transaction. In our setting, it is worth observing that such a feedback should be weighted by the chunk size. More precisely, the user can decide to apply a discounting factor to the feedback result that is inversely proportional to the size of the chunk, in order to reflect that the amount of sensitive information exposed is proportional to the trustworthiness as perceived by the user.

Example: As a consequence of a positive transaction conducted through chunk C_2 and resulting in a trust variation equal to, e.g., $+5$, we would obtain $t(C_2) = 32.5$ if the discounting factor is applied, and $t(C_2) = 35$ otherwise. ■

On the other hand, it is also worth deciding whether and how the feedback related to chunk C has to be propagated to other elements of the trust structure $(\mathcal{C}, \leq, t)$. Since propagation would result in ambiguity if applied to chunks of the poset that cannot be related through $\leq$, let us examine the remaining cases. Depending on the feedback, which can be either positive or negative, and the propagation direction (towards finer or coarser chunks, or else both), every possible combination gives rise to a different propagation policy. For instance, in order to advocate a conservative policy, variations shall not be propagated to elements that refine C, because an interaction disclosing a small amount of sensitive information should not affect the trust level of chunks that expose more information. This policy contrasts also potential attacks by users preserving their identity and aiming at penalizing the trust of small chunks shared by a large number of users. On the other hand, in order to fully exploit the flexibility of the independent model of privacy release, it would be worth propagating the trust variation for C to every chunk C' in the poset that is refined by C. In this case, the trust variation for C' is discounted by a factor proportional to the difference between the size of C and the size of C'. In practice, the larger the difference between C and C' is, the slighter the impact of the trust variation of C upon C'.

Example: Consider chunk C_2 and the positive transaction of the previous example determining $t(C_2) = 32.5$. Then, by virtue of the propagation policy discussed above we have, e.g., $t(C_4) = 16.25$ and $t(C_1) = 45$. ■

As another important assumption, so far we assumed that any new chunk C that is added to the poset is initially associated with the dispositional trust of the user. Alternatively, the trust structure $(\mathcal{C}, \leq, t)$ can be employed to infer some trust information about C. Based on the same intuition behind feedback propagation, the trust values associated with known chunks that are in some relation with C can be combined. In fact, we can interpret C as an approximation of such chunks, which, however, must be pairwise unrelated by $\leq$ to avoid redundancy when counting the related trust values.

By following the conservative policy previously discussed, we initialize the trust towards C on the basis of the trust values associated with chunks that refine C.

Definition 2 (Chunk coverage): Given a trust structure $(\mathcal{C}, \leq, t)$ and a chunk $C \notin \mathcal{C}$, a coverage for C is a set $\{C_1, \ldots C_m\} \subseteq \mathcal{C}$ such that:

- $C_i \not\leq C_j$ for all $1 \leq i, j \leq m$;
- $C \leq C_i$ for all $1 \leq i \leq m$.

The initial trust associated with C by the coverage $\{C_1, \ldots C_m\}$ is $\frac{1}{m} \cdot \sum_{i=1}^{m} t(C_i)$.

Since in the poset several different coverages may exist for a chunk C, we can adopt different policies to select one of them, e.g., by choosing the coverage inducing the highest/lowest trust, or by keeping all of them and then calculating the average trust.

Example: A coverage for chunk $C_8 : 00000\underline{0}\underline{0}$ is the set $\{C_4, C_5\}$, which determines the initial trust value 20. Other candidates are $\{C_2, C_3\}$, $\{C_3, C_4\}$, and $\{C_1\}$. The average trust resulting from all the possible coverages is 30.625. ■

 ISBN: 978-1-61208-376-6

In general, from the effectiveness standpoint, the trust structure $(\mathcal{C}, \leq, t)$ is used to manage locally information (about chunk's trust) allowing the user to approximate the trust towards other users, without any knowledge about their crypto-ids and actual behaviors. As far as efficiency issues are concerned, in order to circumvent the problem of dealing with a huge trust structure, it is possible to constrain the choice of the bitmask, e.g., by fixing a priori a rule for splitting the crypto-id into a limited set of chunks.

Finally, we emphasize that the presentation of the proposed design model abstracts away from the specific trust metric adopted. Indeed, basically, our method may be integrated with any computational notion of trust and with any recommendation mechanism used in classical trust systems for distributed environments [21] [22].

III. Formal Verification

In this section, we evaluate the proposed independent model of privacy release through a comparison with an abstraction of standard approaches in which information linking is irrevocable, in the following called incremental model of privacy release. To this aim, we employ the model checker PRISM [23] [24] [5], through which it is possible to build automatically probabilistic models – like discrete-time Markov chains and Markov decision processes – from state-based formal specifications. On the semantic models deriving from formal descriptions, quantitative properties expressed in probabilistic extensions of temporal logics are verified through model checking techniques.

The comparison is conducted by assuming that the two models of privacy release are applied in the setting of a real-world cooperation system [6], in which users providing services, called *requestees*, and recipients of such services, called *requesters*, are involved in a cooperation process balancing trustworthiness of each participant with access to services and related costs. In the following, we omit the specification of the formal description given in the PRISM modeling language and we briefly introduce the original trust model and its relation with service remuneration [6]. Then, we describe our modeling assumptions and the metrics that are used to evaluate how trading privacy for trust influences access to services and related costs. We finally discuss the obtained results.

A. Trust Model

Trust is a discrete metric with values ranging in the interval $[0, 50]$, such that $null = 0$, $low = 10$, $med = 25$, and $high = 40$. The trust T_{ij} of user i towards any credential j (which can be, e.g., a crypto-id chunk or an entity identity) is modeled abstractly as follows:

$$T_{ij} = \alpha \cdot trust_{ij} + (1 - \alpha) \cdot recs_{ij} \tag{1}$$

Parameter $\alpha \in [0, 1]$ is the risk factor balancing personal experience with recommendations by third parties. The trust metric $trust_{ij}$ is the result of previous direct interactions of i with j. Initially, $trust_{ij}$ is set to the dispositional trust of i, denoted by dt_i. After each positive interaction, $trust_{ij}$ is incremented by a factor v. Parameter $recs_{ij}$ is the average of the trust metrics towards j recommended to i by other users. For each service type, the service trust threshold *st* represents the minimum trust required to negotiate the service.

B. Service Cost Model

The joint combination of trust and remuneration is implemented by making the service cost function dependent on the trust T of the requestee towards the requester credential. The other main parameters are: C_{min}, which is the minimum cost asked by the requestee regardless of trust, C_{max}, which is the maximum cost asked to serve untrusted requests, and the threshold values T' and T'', such that $T'' < T'$.

The cost function proposed in [6] expresses linear dependence between trust and cost:

$$C(T) = \begin{cases} C_{min} + \frac{C_{max} - C_{min}}{T'} \cdot (T' - T) & \text{if } T < T' \\ C_{min} & \text{otherwise} \end{cases} \tag{2}$$

In order to examine thoroughly the trust/cost tradeoff, we consider two more functions approximating the linearity of the relation between trust and cost. In particular, a simple one-step function is as follows:

$$C(T) = \begin{cases} C_{max} & \text{if } T < T' \\ C_{min} & \text{otherwise} \end{cases} \tag{3}$$

while a possible two-steps function is as follows:

$$C(T) = \begin{cases} C_{max} & \text{if } T < T'' \\ C_{max}/2 & \text{if } T'' \leq T < T' \\ C_{min} & \text{otherwise} \end{cases} \tag{4}$$

C. Modeling Assumptions

Our objective is to compare the model of incremental release of privacy (represented in the figures by the curves named inc) with the model of independent release of privacy (represented in the figures by the curves named ind). For the sake of uniformity, for both models we assume abstractly that privacy is released (through the pseudonyms mechanism [14] and through the chunk mechanism, respectively) as a percentage of the total amount of sensitive information that the user may disclose. Similarly, in every trust-based formula we consider percentages of the trust involved.

The experiments are conducted by model checking several configurations of the system against formulas expressed in quantitative extensions of Computation Tree Logic [5]. For instance, Figure 2 refers to one requester interacting with one requestee with the aim of obtaining 10 services that can be of three different types. The figure reports the results for the best strategy, if one exists, allowing the requester to get access to all the services requested by minimizing the total expected cost (reported on the vertical axis) depending on the amount of revealed sensitive information (reported on the horizontal axis). The choice of the amount of privacy to spend for each request is under the control of the requester. The choice of the service type is either governed by the requester, or it is probabilistic with uniform distribution (see the curves denoted by $prob$ in the figure). Requestee's parameters are $dt = med$ and $v = 5$, as we assume that each transaction induces a positive feedback. The three service types are characterized by $st_1 = null$ and (2), $st_2 = low$ and (3), $st_3 = med$ and (4), respectively. The service cost parameters are $C_{min} = 0$, $C_{max} = 10$, $T' = high$, and $T'' = med$.

 ISBN: 978-1-61208-376-6

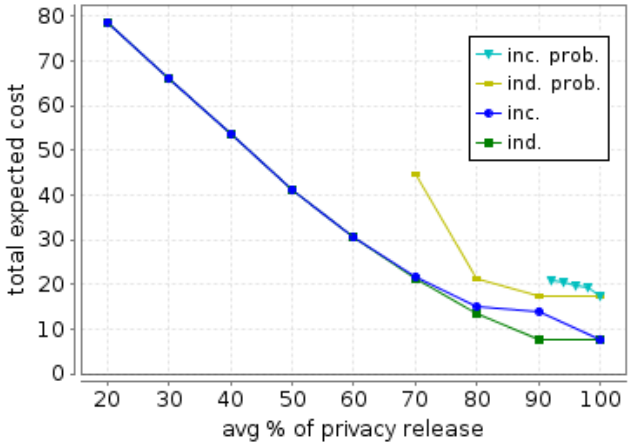


Figure 2. Trading cost for privacy.

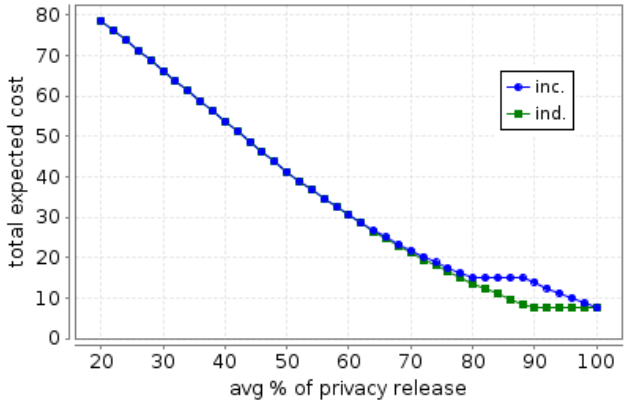
We complete the comparison with an experiment assuming one requester and two requestees, which are chosen nondeterministically by the requester. The number of issued requests is 10, while we consider only the first type of service. The analysis, reported in Figure 3, proposes the results obtained by changing the service cost function. Requestee's trust parameters are as follows: $dt = med$, $st = null$, $\alpha = 0.5$.

D. Discussion

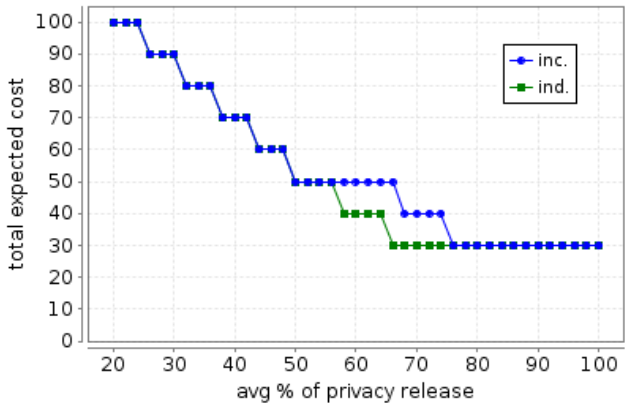
We now comment on the obtained results, by first considering Figure 2, which reveals two interesting behaviors.

Firstly, if the choice of the service is under the control of the requester, then the difference between the two models is significant only for values of the privacy release higher than 70%. In order to interpret this result, we checked the best requester's strategy, which consists of choosing always the service offering the best ratio trust/cost, i.e., the one using (2). Whenever trust is high enough to apply the minimum cost, then it turns out to be convenient to select also the other two service types. According to this strategy, if the privacy disclosure is below 70% it happens that trust does not reach the threshold T'. Therefore, as a consequence of (2), the relation between trust and cost is always linear and the two privacy models turn out to be equivalent from the economic standpoint. On the other hand, if the requester is highly trustworthy, then the cost to pay becomes constantly equal to the minimum cost, meaning that the requester could invest less privacy to obtain the same cost, thus revealing the advantages of the independent model. In practice, independently of the privacy model, it is economically convenient for the requester to disclose the information needed to obtain rapidly the best cost. Instead, for high levels of trust, it would be convenient for requester's privacy to reduce as much as possible the amount of disclosed information. Whenever identity of the requester is always fully disclosed, then the two models experience the same performance.

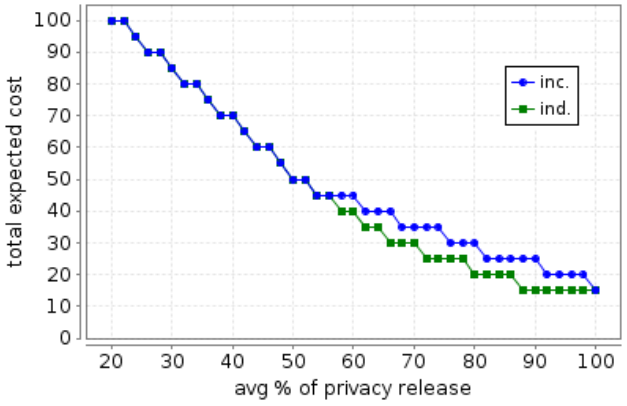
Secondly, if the choice of the service is probabilistic, thus modeling, e.g., a situation in which the requester may require every type of service independently of their cost, then it is not possible to satisfy all the requests if a minimum disclosure of



(a) Equation (2).



(b) Equation (3).



(c) Equation (4).

Figure 3. Trading cost for privacy by varying cost function.

privacy is not guaranteed. However, such a minimum value is considerably higher for the incremental model, in which case at least an average privacy release of 92% is needed. Hence, if the requester is somehow forced to require certain services, then the independent model performs better.

The role of the service cost function is emphasized by the curves of Figure 3, which show that whenever a step function is used, the independent model is able to exploit better the intervals of trust in which the service cost is constant.

 ISBN: 978-1-61208-376-6

In the previous experiments, priority is given to cost and to the average disclosure of privacy needed to optimize such a cost. However, if cost is not a fundamental issue, then the tradeoff of interest concerns trust and privacy. In order to analyze such a tradeoff, we reformulate the experiment of Figure 2 by focusing on the optimization of the average percentage of privacy release needed to obtain 10 services of a given type. In particular, we consider the second and third service types, for which the service trust threshold is *low* and *med*, respectively. Since to obtain such services the requester must be trusted by the requestee, we examine the tradeoff between such a trust and requester's privacy. For the second (resp., third) service type, the average percentage of privacy release is 38% (resp., 92%) when applying the incremental model, while it is equal to 28% (resp., 64%) in the case of the independent model. Therefore, the observed values show that through the independent model we obtain all the required services by disclosing much less privacy than through the incremental model. The related difference is directly proportional to the trust threshold needed to obtain the services.

IV. Conclusion

The attitude to cooperation is strongly affected by the tradeoff existing among privacy and trustworthiness of the involved parties and cost of the exchanged services. In order to balance the related incentive mechanisms, it is worth considering the constraints of the model of privacy release. Thanks to a mechanism based on the splitting of crypto-ids, it is possible to manage the disclosure of sensitive information in a less restrictive way with respect to classical models, even in distributed environments.

The formal evaluation has emphasized that the flexibility of the independent model ensures better performance with respect to the incremental model. This is always true if the main objective is trading privacy for trust. If services must be paid and cost depends on trust, then the adopted cost function affects the tradeoff among privacy, trust, and cost, by revealing the advantages of the independent model in the intervals of trust values in which cost is constant.

As work in progress, the integration of the proposed distributed trust system with the centralized reputation system of [4] is under development. Moreover, a successful deployment of the proposed model is strictly related to the choice of the trust policies and configuration parameters discussed in Section II, which are currently subject to sensitive analysis through formal verification.

References

[1] A. Aldini and A. Bogliolo, Eds., User-Centric Networking – Future Perspectives, ser. Lecture Notes in Social Networks. Springer, 2014.

[2] A. Jøsang, "Trust and reputation systems," in Foundations of Security Analysis and Design IV (FOSAD'07), ser. LNCS, A. Aldini and R. Gorrieri, Eds. Springer, 2007, vol. 4677, pp. 209–245.

[3] S. Greengard, "Social games, virtual goods," Communications of the ACM, vol. 54, no. 4, 2011, pp. 19–22.

[4] A. Aldini, A. Bogliolo, C. Ballester, and J.-M. Seigneur, "On the tradeoff among trust, privacy, and cost in incentive-based networks," in 8th IFIP WG 11.11 Int. Conf. on Trust Management, ser. IFIP AICT, J. Zhou et al., Eds., vol. 430. Springer, 2014, pp. 205–212.

[5] V. Forejt, M. Kwiatkowska, G. Norman, and D. Parker, "Automated verification techniques for probabilistic systems," in Formal Methods for Eternal Networked Software Systems, ser. LNCS, M. Bernardo and V. Issarny, Eds. Springer, 2011, vol. 6659, pp. 53–113.

[6] A. Bogliolo et al., "Virtual currency and reputation-based cooperation incentives in user-centric networks," in 8th Int. Wireless Communications and Mobile Computing Conf. (IWCMC'12). IEEE, 2012, pp. 895–900.

[7] Y. Zhang, L. Lin, and J. Huai, "Balancing trust and incentive in peer-to-peer collaborative system," Journal of Network Security, vol. 5, 2007, pp. 73–81.

[8] M. Yildiz, M.-A. Khan, F. Sivrikaya, and S. Albayrak, "Cooperation incentives based load balancing in UCN: a probabilistic approach," in Global Communications Conf. (GLOBECOM'12). IEEE, 2012, pp. 2746–2752.

[9] Z. Li and H. Shen, "Game-theoretic analysis of cooperation incentives strategies in mobile ad hoc networks," IEEE Transactions on Mobile Computing, vol. 11, no. 8, 2012, pp. 1287–1303.

[10] A. Aldini and A. Bogliolo, "Model checking of trust-based user-centric cooperative networks," in 4th Int. Conf. on Advances in Future Internet (AFIN2012). IARIA, 2012, pp. 32–41.

[11] A. Aldini, "Formal approach to design and automatic verification of cooperation-based networks," Journal On Advances in Internet Technology, vol. 6, 2013, pp. 42–56.

[12] M. Kwiatkowska, D. Parker, and A. Simaitis, "Strategic analysis of trust models for user-centric networks," in Int. Workshop on Strategic Reasoning (SR'13), vol. 112. EPTCS, 2013, pp. 53–60.

[13] A. Aldini and A. Bogliolo, "Modeling and verification of cooperation incentive mechanisms in user-centric wireless communications," in Security, Privacy, Trust, and Resource Management in Mobile and Wireless Communications, D. Rawat, B. Bista, and G. Yan, Eds. IGI Global, 2014, pp. 432–461.

[14] J.-M. Seigneur and C.-D. Jensen, "Trading privacy for trust," in 2nd Int. Conf. on Trust Management (iTrust'04), ser. LNCS, vol. 2995. Springer, 2004, pp. 93–107.

[15] M. Raya, R. Shokri, and J.-P. Hubaux, "On the tradeoff between trust and privacy in wireless ad hoc networks," in 3rd ACM Conf. on Wireless Network Security (WiSec'10), 2010, pp. 75–80.

[16] L. Lilien and B. Bhargava, "Privacy and trust in online interactions," in Online Consumer Protection: Theories of Human Relativism. IGI Global, 2009, pp. 85–122.

[17] W. Wagealla, M. Carbone, C. English, S. Terzis, and P. Nixon, "A formal model of trust lifecycle management," in Workshop on Formal Aspects of Security and Trust (FAST'03), 2003.

[18] S. Köpsell and S. Steinbrecher, "Modeling unlinkability," in 3rd Workshop on Privacy Enhancing Technologies, ser. LNCS, vol. 2760. Springer, 2003, pp. 32–47.

[19] I. Goldberg, "A pseudonymous communications infrastructure for the internet," Ph.D. dissertation, University of California at Berkeley, 2000.

[20] A. Kobsa and J. Schreck, "Privacy through pseudonymity in user-adaptive systems," ACM Transactions on Internet Technology, vol. 3, no. 2, 2003, pp. 149–183.

[21] S.-D. Kamvar, M.-T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in 12th Conf. on World Wide Web (WWW'03). ACM, 2003, pp. 640–651.

[22] R. Zhou and K. Hwang, "Powertrust: a robust and scalable reputation system for trusted peer-to-peer computing," IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 4, 2007, pp. 460–473.

[23] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis, "Prism-games: a model checker for stochastic multi-player games," in 19th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'13), ser. LNCS, vol. 7795. Springer, 2013, pp. 185–191.

[24] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: verification of probabilistic real-time systems," in 23rd Int. Conf. on Computer Aided Verification (CAV'11), ser. LNCS, vol. 6806. Springer, 2011, pp. 585–591.

 ISBN: 978-1-61208-376-6

Enhancing Privacy on Identity Providers

Rafael Weingärtner Carla Merkle Westphall

Department of Informatics and Statistics
Networks and Management Laboratory
Federal University of Santa Catarina
Florianópolis, Brazil
Email: {weingartner, carla}@lrg.ufsc.br

***Abstract*—Cloud computing is widely used to provide on demand services as a consequence of its benefits such as reduced costs, structure flexibility and agility on resource provisioning. However, there are still people that are not comfortable with the idea of sending their sensitive data to the cloud such as the personally identifiable information (PII) that could be used to identify someone in the real world. Moreover, there have been cases of data leaks, which resulted in huge losses both for companies and its clients. Therefore, this article addresses the security and privacy aspects of identity management. We present a model that tackles privacy issues within the PII that is stored on identity providers (IdPs). Thus, our proposal supports users and improves theirs awareness when disseminating PIIs.**

Keywords–Cloud Computing; Security; Privacy; Federation; Identity providers;

I. Introduction

Cloud computing is been largely adopted to provide services to industry. As presented in [1] and [2], the reduced costs, flexibility and agility are the main characteristics for the widespread successful of cloud computing. However, there are people that are not comfortable to send their sensitive data to the cloud [3]. Moreover, it is pointed out by the Cloud Industry Forum in [2] that when the cloud is in discussion there are huge debates not about the technology aspect per se, but rather about the commercial and governance issues that relate to data security and privacy.

Users have the right to be skeptic about the privacy and security aspects of that model. Hence, there have been recent cases of data breaches and leaks as noticed in [4] [5] [6], which resulted in identity data leaks. Therefore, as pointed by Betgé-Brezetz, Kamga, Dupont and Guesmi in [7] cloud service providers should focus on protecting sensitive data than on tight security perimeters, hence, the biggest threat may be internal.

Sánchez, Almenares, Arias, Díaz-Sánchez and Marín in [8] and De Capitani di Vimercati, Foresti and Samarati in [9] discussed that as soon as users' data is on identity providers (IdP) the control on how that data is disclosed, stored and used is lost. Moreover, data stored in the cloud may be sensitive and if linked with its owner identity may violate his/her privacy.

This paper addresses some security and privacy aspects of identity providers. In one side, we tackle the lack of control that users have over their identification data (PII) that is stored on identity providers. On the other side, it is proposed an enhancement in the dissemination process to support users with their PII data disclosure, in a way that it is lowered the risks of unaware/unintentional data dissemination.

The rest of this paper is structured as follows. Section II gives a brief overview of the concepts that are going to be used throughout this paper. Section III presents and discusses related works. Section IV describes the issue that is going to be addressed and presents our proposals. Section V closes the paper with the conclusions and future works.

II. Background

In order to provide a better understanding of the issue that is being addressed and the proposed model, this section presents a brief overview on each concept that will be used throughout the rest of this paper.

A. Privacy

Landwehr and et al. in [10] defines privacy as the control of release of personal data that users have. Furthermore, privacy is a fundamental human right as pointed out by United Nations (UN) in its universal declaration of humans rights [11]. In addition, the Human Rights Council reinforced that the same right that people have off-line must also be protected on-line [12].

Therefore, privacy is a vital characteristic that has to be considered into every system. Identity provider systems should not be an exception and have privacy added into its design.

In addition, Diaz and Gürses presented in [13] three different paradigms of privacy:

- Privacy as a control – privacy violations are often associated with disclosure of data to third parties. In this context, privacy technologies provide individuals with means to control the disclosure of their information and organizations with means to define and enforce data security policies to prevent abuse of personal information for unauthorized purposes. Thus, the main goal of this paradigm is to provide users with control and oversight over collection, processing and use of their data;
- Privacy as confidentiality – the previous paradigm relies on the assumption that organizations that collect and process users' data are completely honest. However, once data is under the control of an organization, it is hard for individuals to verify how their data is being used. This paradigm aims to prevent information disclosure, focusing on minimizing the information

 ISBN: 978-1-61208-376-6

disclosed in a way that cannot be linked to users identity;

- Privacy as practice – this paradigm views privacy in a social dimension, as users make privacy decisions often based on how their social groups make those decisions. In this context, technologies strive to make information flow more transparent through feedback and awareness, enabling a better individual and collective understanding on how information is collected, analyzed and used.

Moreover, there are plenty of legislations that aim to protect users' privacy in the Internet and communication systems. In Europe, there is the Data Protection Directive [14], in USA, we have the Health Insurance Portability and Accountability Act (HIPAA) [15], the Gramm-Leach-Bliley Act [16], the Children's Online Privacy Protection Rule [17] and in Brazil, it was recently approved the Internet Bill of Rights[18]. All of those aforementioned acts aim to protected users against unwilling data disclosure and processing.

B. *Identity management*

Identity management can be defined as the process of managing users' identity attributes [19]. Moreover, Hansen, Schwartz and Cooper in [20] stated that identity management systems are programs or frameworks that administer the collection, authentication, and use of identity and information linked to identity. Thus, it provides means to create, manage and use identities' attributes.

Bertino and Takahashi in [21] presented the roles that exist in an identity management system:

- Users – entities that want to access some kind of service or resource;
- Identity – set of attributes that can be used to represent a user, it is also called personally identifiable information (PII);
- Identity provider (IdP) – provide means to manage users' attributes. It delivers users' PIIs to service providers;
- Service provider (SP) – delivers the resource/service desired by a user. It delegates the process of authentication to IdPs and usually is responsible for the authorization process.

Therefore, identity management systems are the frameworks, which enable users to properly manage their PIIs. Thus, they enable users to access resources and services using identification data that is stored in identity providers, from which a subset of the identification attributes may be disclosed to service providers.

In this context we also have the concept of federation, which is define by Chadwick in [19] as an association of service providers and identity providers. Furthermore, Orawiwattanakul, Yamaji, Nakamura, Kataoka and Sonehara in [22] said that a federation allows users to access resources in multiple administrative domains (ADs) by initially authenticating with their home AD instead of authenticating with the accessed one.

Therefore, identity federation is a set of standards and technologies that enable the exchange of identities in a secure way between different administrative domains.

Shibboleth [23] is one of the tools that can be used to create a federation; it uses Security Assertion Markup Language (SAML) to exchange data between IdPs and SPs. In one hand, it has an IdP module that is developed in Java and can cope with distinct data repositories, such as databases, Lightweight Directory Access Protocol (LDAP) and Central Authentication Service (CAS). On the other hand, its SP module is developed in *C* as a module for the Apache Web Server and it is used to manage the access control of a protected resource.

Shibboleth [23] has a plug-in called uApprove.jp, which is presented in [22] that provides users with some means to manage PII disclosure and some feedback about the reasons of the data collection. Figure 1 presents the Shibboleth's with its plug-in uApprove.jp work flow. Each step is described as follows:

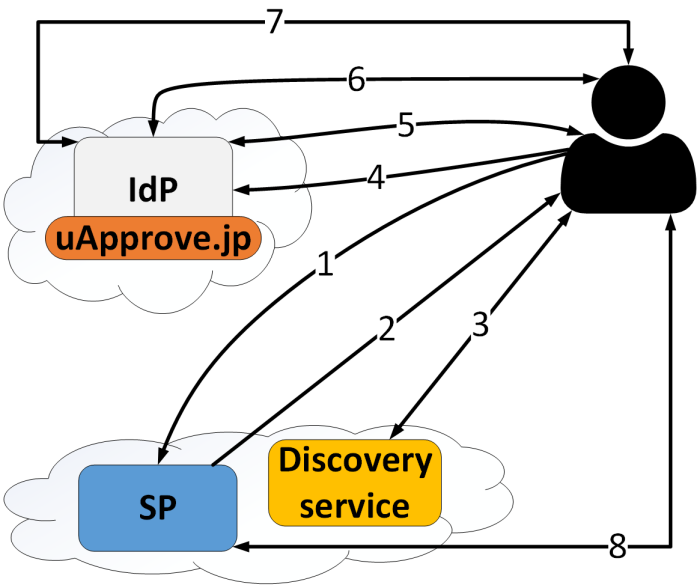


Figure 1. Shibboleth + uApprove.jp workflow

1) Unauthenticated users by means of a browser access a protected resource;
2) The service provider sends users to the discovery service (DS) in which they have to choose an IdP that has their attributes;
3) Users submit to DS the IdP they are enrolled. The DS starts the session initiators at the protected resource and sends users to the selected IdP;
4) IdP answers the request and presents users with a login page in which they have to enter their credentials;
5) Users present their credentials that are checked upon the IdP database. If the authentication process ends with success, the IdP presents users with SP's terms of usage (ToU), which users should read and accept;
6) After the ToU acceptance, users are presented with an attribute release page of uApprove.jp, which will display user's attributes from which the user can select/unselect the optional ones, accordingly to she/he will.
7) The IdP creates an assertion with the result of the authentication and user's attributes that were chosen

by the user to be disclosed, which is sent to the SP through the users' browser;

8) With the authentication confirmation and some PII data the SP can deliberate about the resource delivery.

There are other tools that can be used to create federations such as OpenAM and OpenId Connect. This paper uses Shibboleth for its widespread adoption in the academia and because it is developed and maintained by the Internet 2 foundation as a free open source framework to build federations.

III. Related work

Switch in [24] developed a plugin to Shibboleth IdPs that provides awareness of data disclosure when accessing some resource/service. However, users cannot select which data is going to be disclosed, the user has either to agree or disagree with the PII dissemination.

Orawiwattanakul, Yamaji, Nakamura, Kataoka and Sonehara in [22] tackled the lack of control on PII disclosure in cloud federations. It proposed an extension of [24] that would enable users to select among all non-mandatory attributes which ones they wish to disclose to the SP that is being accessed. This way, it guarantees that data disclosure is happening with user consent.

In a different approach to deal with privacy in cloud, Sánchez, Almenares, Arias, Díaz-Sánchez and Marín in [8] proposed a reputation protocol that weights the reputation of entities in a federation in order to support data disclosure. This way, users can check SPs reputations among the federation before they send any data to it. It is also provided a way in which users would have the ability to check what is being done with their data, and based on that they could lower or increase the provider reputation.

Betgé-Brezetz, Kamga, Guy-Bertrand, Mahmoud and Dupont in [25] addressed the cloud privacy and security issues in which users send data to cloud providers without any guarantee that it is going to be secured in a proper way. As Sánchez, Almenares, Arias, Díaz-Sánchez and Marín did in [8], it was proposed a o define if a user trusts or not a cloud provider and the level of trust. Based on how much the user trusts the cloud provider, he/she could send data in plain text, partially encrypted (encrypted with some metadata in plain text) or fully encrypted to the cloud. It was also proposed a package called PDE (Privacy Data Envelope) to carry users' data to the cloud. That package could hold the data (encrypted or not) with some policies that state how, where, by whom and when that data can be used.

Works [8] and [25] suffer from the same problem, a SP with a good reputation does not mean that it is not vulnerable to attacks, and that it is taking all the required measures to guarantee users privacy.

As an alternative to previous presented works, Chadwick and Fatema in [26] addressed the lack of means to create access policies for data stored in the cloud and the absence of standards to apply such policies defined not just by users, but also, by countries where data is stored. It was proposed a series of web services that would analyze policies that are uploaded within the data before any action is executed. Therefore, once an application receives a request to process some data, it should consult the proposed web services if it can proceed with the requested action.

TABLE I. PROPERTIES OF WORKS

Publications		Characteristics				
Reference	Year	Use of cryptography	Based on reputation	Use of Policies	Awareness of data disclosure	Disclosure support
[24]	–				X	
[22]	2010			X	X	
[8]	2012		X			
[25]	2012	X	X			
[26]	2012			X		
[7]	2013	X	X	X		
Our proposal	2014	X		X	X	X

Betgé-Brezetz, Kamga, Dupont and Guesmi in [7] combined the approached of reputation presented in [25] with policies presented in [26]. Its proposal addresses privacy issues of cloud computing in an end-to-end fashion way. It used stick policies with the PDE proposed in [25] to carry all together policies and data to the cloud. The proposal consists in adding on cloud service providers points that evaluate those policies before using the data, these points are called data protection module (DPM), which would guarantee the evaluation of defined policies before any process is made with the data. It is also defined that the PDE containing the policies and data would just be sent (processed, copied and stored) into cloud nodes that have the DPMs modules deployed.

Works [7] and [26] experience the same problem, that is the lack of guarantee that a provider is truly obeying the proposed models. Users do not have means to check if the protection modules were developed, deployed and are working properly.

Having presented the related works, we can categorize the papers that were presented into the following properties:

- Use of cryptography – use of cryptography to store data at a provider;
- Based on reputation – use of reputation to back up users' decision of which data and how it is sent to SPs;
- Use of Policies – policies that regulate how data is used/disclosed at a provider;
- Awareness of data disclosure – provide feedback to make users aware of data dissemination;
- Disclosure support – provide means to support users when they are disseminating data from an IdP to a SP.

Table I matches the properties shown above with the ones found in presented related works. Therefore, it can be noticed that our proposal combines the properties found in related works, striving to enhance the support and privacy in identity providers.

IV. Enhancing privacy on identity providers

This section discusses and presents the issues that are being addressed. Thus, it introduces our proposals to tackle those problems.

A. Privacy issues

There are legislations [14] [15] [16] [17] [18] and guidelines create by Jansen and et al. [27] and Security Alliance in [28] to address privacy issues that arise in information systems. Those laws and standards aim to guarantee users rights over their data. Furthermore, works [7] [8] [22] [24] [25] [26] tried

 ISBN: 978-1-61208-376-6

to address some of the issues that exist when data is stored out of users boundaries. However, there are still a lack of models and mechanisms:

- Lack of control over user's PII – users do not have effective means to manage their data that is stored in identity providers;
- Disclosure support – as presented in a research by Zhang and et al. in [29] people could not successfully define their personal information disclosure policies. Therefore, there should be created a way to support users when they are disseminating PII information.

The lack of control that users have over their sensitive data gets worse once they migrate to cloud services. As presented by Mather, Kumaraswamy and Latif in [30], once organizations have migrated to the cloud they lose control over their structure used to host services. Moreover, Zhang and et al. discussed in [31] that loss of control can lead to data leaks as a consequence of curious/malicious system administrators of the underlying structures.

B. *Working with privacy in identity providers*

Our proposal uses the concepts of privacy described by Diaz and Gürses [13], striving to minimize data disclosure and provide means for users to effectively control personal data disclosure. Thus, it makes the flow of data more transparent providing users awareness of data dissemination.

In addition, users are responsible for data entered into IdPs, which is then used to access some service provided by an SP. Thus, users should have means to proper control data disclosure, and that process must be improved to be more transparent for its users.

Therefore, we extended the federation framework presented earlier. In one hand, we added templates for data dissemination to support users with the PII disclosure. On the other hand, we used the cryptography approach to store PII data encrypted in IdPs.

Our model is presented in Figure 2, users would enter their PII data into IdP providers encrypted with some key, therefore, the disclosure process had to be extended to allow users to open the data they wish to disseminate. We propose that layer of protection over the PII data, in order to make it harder to access and disseminate that data without users' awareness and consent.

We also created a way in which users can send their preferences for data dissemination to IdPs in order to ease and secure the disclosure process. As pictured in Figure 2, those preferences would be created as policies written in XML, they would be drawn by entities such as security labs, privacy commissioners and security experts of the area who hold the knowledge of which data can cause more or less harm to users' privacy if disclosed.

The process of data dissemination from IdPs to SPs was extended to cope with our proposal of templates for data dissemination. The dissemination process has to use the proposed templates to support users with data disclosure.

Therefore, our proposal adds new objects into the model of identity management of the Shibboleth framework. Each object and its role is described as follows:

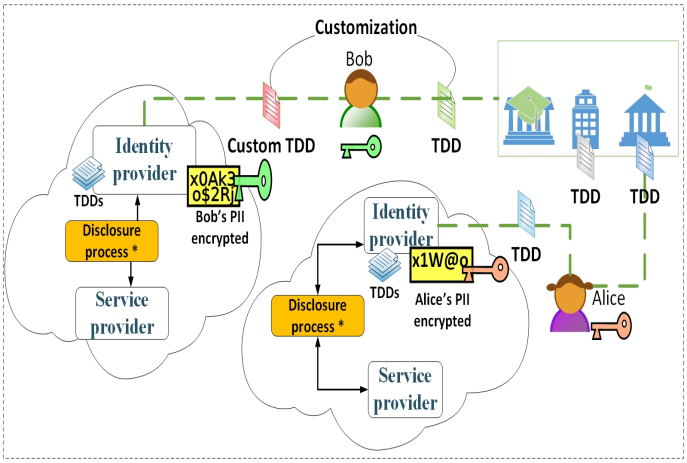


Figure 2. Enhancing privacy on identity providers.

- Template data dissemination (TDD) – it is the template which users can get from entities that the user trust, customize if needed (as Bob does in Figure 2) and enter it into IdPs to help them manage their PIIs release. It guides users throughout the disclosure process with different granular configuration to different SPs;
- Cryptography Keys – are the keys used to encrypt and decrypt users PII that is store in the IdP. Users would encrypt their PIIs before sending them to IdPs with Key I, and during a transaction when some PII data is needed users would be asked to open that data with key II in order to disseminate it to a SP.

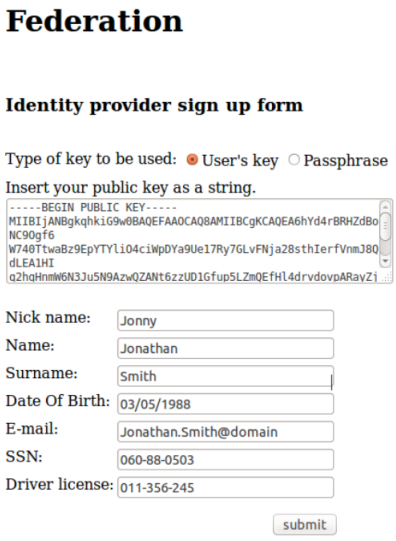
The following subsections present the extensions that we developed in order to make Shibboleth IdP and its uApprove.jp plugin cope with our proposals. We divided the work into addressing the loss of control on users PIIs and adding support to users at the disclosure process.

1) Addressing the loss of control on users PIIs: Papers [7] [31] [26] suggested that there could be curious/malicious internal entities into providers (SP and IdP) with privileges and technical means to harm users privacy. Therefore, we propose to store users' PIIs into IdPs encrypted in a way that just the user can decrypt the data and use it.

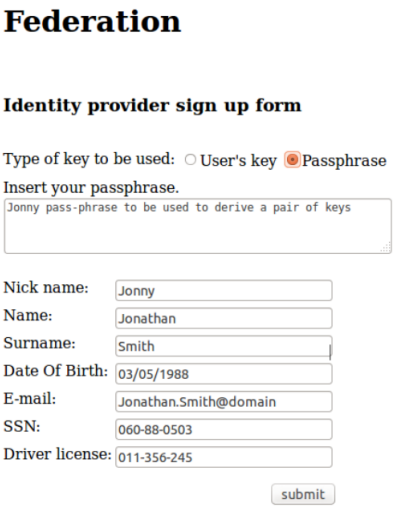
We did not propose any way to deal with this situation at the SP side at this moment. In one hand, because as argued by Chadwick in [19] if the fair principles of data collection and minimization are followed the SP will just receive a pseudonym and some data that by themselves do not give any hint about the user's identity. On the other hand, because the IdP concentrate all the sensitive information needed to link a system user to a person. Furthermore, as presented by De Capitani di Vimercati, Foresti and Samarati in [9], data per se is not sensitive, what is sensitive is its association with an identity.

We developed a tag library using Java Web technologies to be used as a basic framework to create forms in which users would enter their PII data as they usually do when creating an account in some IdP system as shown in Figure 3. However, the data that is sent to the IdP will be encrypted with some key, just the password and the login would not be encrypted, as they are needed to execute the authentication process.

 ISBN: 978-1-61208-376-6

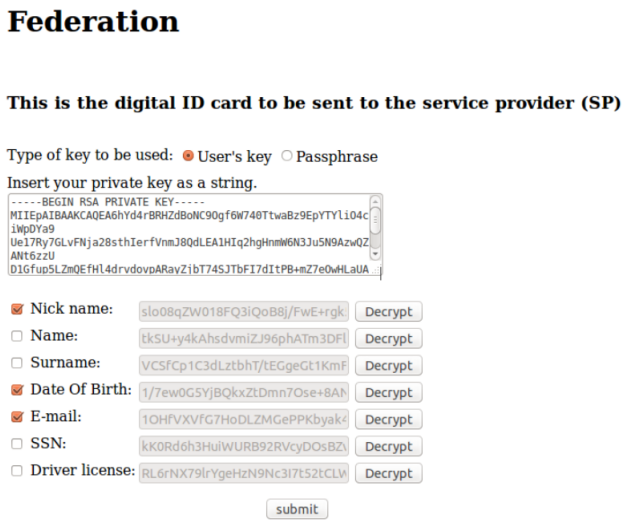


(a) User's public key.

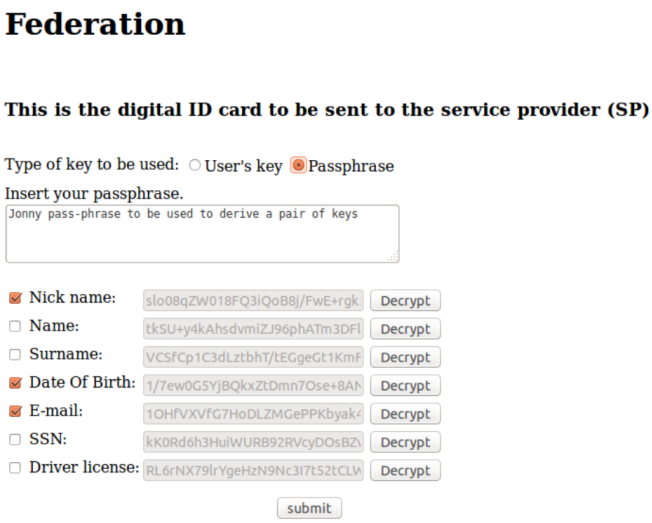


(b) Key derivation from passphrase.

Figure 3. Privacy enhanced sign up forms for IdP.



(a) User's private key.



(b) Key derivation from passphrase.

Figure 4. User decrypting data to send to SP.

The framework we developed gives the following options to users when asking for a key:

- Use a public key – the user can choose to enter a public key that she/he already has as the key to encrypt the PII data, as shown in Figure 3(a);
- Use a pass-phrase – users can enter a pass-phrase that is used to derive a pair of keys from which we take the first one and encrypted their data before sending them to the IdP, as depicted in Figure 3(b).

Both of the aforementioned approaches are performed at the client side, the user's keys are never sent to the IdP server. Thus, to encrypt the data at the client site we used the web programming language Javascript with libraries Cryptico [32] and pidCrypt [33] respectively when users desire to use a passphrase or a public key to encrypt her/his data. Thus, both libraries are based on the Javascript cryptography library developed by Wu [34].

Our proposal inserts data into the IdP encrypted. Thereby, we had to change the flow of message presented in Figure 1, hence the IdP would not have users' PII in clear text anymore. It was needed an extension to enable users to decrypt the data that is going to be sent to SPs as pictured in Figure 4.

If the user selected to send data encrypted with a pass-phrase or a public key, there will be some difference when we decrypt the PII needed to send to the SP.

In one side, if users selected to encrypt data with a public key, when the decryption is required we ask them for a private key as depicted in Figure 4(a). On the other side, if they chose to encrypt data with a key derived from a pass-phrase, we then ask for the pass-phrase to derive the keys, from which we use the second key generated to decrypt the data as pictured in Figure 4(b).

2) Adding support to users at the disclosure process: Birrell and Schneider discussed in [35] that the control of PII dissemination can be inconvenient forcing users to decide which data can be sent to which SP every time they access a new service. Furthermore, Zhang and et al. in [29]

 ISBN: 978-1-61208-376-6

demonstrated that users usually fail to successfully define their data disclosure policies. Thus, Hansen, Schwartz and Cooper in [20] argued that one single default setting would not suit properly every user needs. Therefore, we proposed the use of TDDs based on different user types, this way, we could have different TDDs, enabling users to customize data disclosure in a granular way. The TDDs developed in XML look like the document presented in Figure 5.

```
<?xmlversion="1.0"encoding="UTF?8"?>
<templateDataDissemination
xmlns="http://privacy.lrg.ufsc.br/tdd"
xmlns:xsi=
"http://www.w3.org/2001/XMLSchema?instance" xsi:schemaLocation="
http://privacy.lrg.ufsc.br/tdd
http://privacy.lrg.ufsc.br/tdd-1.0.xsd">
<spDomain>sp.domain.com</spDomain>
<spAttributesBehaviours>
       <attributeBehaviour>
              <attributeName>name</attributeName>
              <selectedByDefault>true</selectedByDefault>
       </attributeBehaviour>
       <attributeBehaviour>
              <attributeName>lastName</attributeName>
              <selectedByDefault>true</selectedByDefault>
       </attributeBehaviour>
       <attributeBehaviour>
              <attributeName>email</attributeName>
              <selectedByDefault>false</selectedByDefault>
       </attributeBehaviour>
       <attributeBehaviour>
              <attributeName>SSN</attributeName>
              <selectedByDefault>false</selectedByDefault>
       </attributeBehaviour>
       <attributeBehaviour>
       <!-- Any other we use false -->
              <attributeName>*</attributeName>
              <selectedByDefault>false</selectedByDefault>
       </attributeBehaviour>
</spAttributesBehaviours>
</templateDataDissemination>
```

Figure 5. Example of TDD

Thereby, we extended the Shibboleth IdP to use the TDDs shown above. This way, when users reach the process of PII disclosure, they will be presented with a page in which the attributes to be disclosed will already be selected/deselected.

V. Conclusion

While papers [8] [25] and [7] [26] tried to manage privacy in the cloud respectively by assessing cloud service providers reputation and creating sticky policies within data, our proposal tackles the lack of control of users' PIIs into IdPs and the lack of support when disclosing PIIs to SPs, respectively by encrypting PIIs into IdPs and using templates for data dissemination to support users when disclosing data.

Our proposal avoids curious and malicious system administrators to gather users' PII data without permission in IdPs. If an administrator accesses the data repository she/he will not be able to retrieve any relevant data about a user identity, hence, that sensitive information will be encrypted.

Furthermore, our proposal is a lightweight extension on top of Shibboleth identity provider and its uApprove.jp plugin, which works transparently to SPs, thence, all of the extensions were developed at the IdP. Moreover, once the proposal is deployed it can prevent PII data leaks that cause identity theft and the correlation of big data processing with a specific user's identity without her/his consent.

In addition, this paper focused on tackling some privacy issues in identity providers, there are still issues to be dealt with at the service provider side, such as means to control attributes that were released from an IdP to a SP. Our proposal of personas to manage the granular release of users PIIs has the goal to lower the risks that arise with the dissemination of certain combination of attributes. It does not protect privacy by itself; users are still vulnerable to malicious SPs that may collude to profile a user identity in a federated environment. Therefore, as a future works we intend to investigate means to enforce users privacy in service providers.

As a next step to be taken in our research we will extend the OpenId Connect federation protocol, in order to add our proposals. The OpenId Connect protocol uses JSON instead of SAML (XML), which makes it easier to use in mobile environments in which XML processing can become a problem. We also intend to investigate the possibility to use web semantic into our proposals, to ease the adaptation of systems already developed and to decouple identity management models and protocols from the technology aspect.

Acknowledgment

The research is funded by the Brazilian Funding Authority for Studies and Projects (FINEP) under the Brazilian National Research Network in Security and Cryptography project (RENASIC) and conducted at the virtual laboratory of secure implementations (LATIM) at the Federal University of Santa Catarina (UFSC) in the Networks and Management laboratory (LRG).

References

[1] P. Hall, "Opportunities for csps in enterprise-grade public cloud computing," OVUM, May, 2012.

[2] C. I. Forum, "Uk cloud adoption and trends for 2013," Tech. Rep., 2013. [Online]. Available: http://cloudindustryforum.org/downloads/whitepapers/cif-white-paper-8-2012-uk-cloud-adoption-and-2013-trends.pdf

[3] S. Srinivasamurthy and D. Liu, "Survey on cloud computing security," in Proc. Conf. on Cloud Computing, CloudCom, vol. 10, 2010.

[4] M. Helft, "After breach, companies warn of e-mail fraud," The New York Times, Abril 2011, retrieved: February, 2014. [Online]. Available: http://www.nytimes.com/2011/04/05/business/05hack.html

[5] D. Kocieniewski, "Adobe announces security breach," The New York Times, Outubro 2013, retrieved: February, 2014. [Online]. Available: http://www.nytimes.com/2013/10/04/technology/adobe-announces-security-breach.html

[6] C. Sang-Hun, "Theft of data fuels worries in south korea," The New York Times, Janeiro 2014, retrieved: February, 2014. [Online]. Available: http://www.nytimes.com/2014/01/21/business/international/theft-of-data-fuels-worries-in-south-korea.html

[7] S. Betgé-Brezetz, G.-B. Kamga, M.-P. Dupont, and A. Guesmi, "End-to-end privacy policy enforcement in cloud infrastructure," in Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on. IEEE, 2013, pp. 25–32.

[8] R. Sánchez, F. Almenares, P. Arias, D. Díaz-Sánchez, and A. Marín, "Enhancing privacy and dynamic federation in idm for consumer cloud computing," Consumer Electronics, IEEE Transactions on, vol. 58, no. 1, 2012, pp. 95–103.

[9] S. De Capitani di Vimercati, S. Foresti, and P. Samarati, "Managing and accessing data in the cloud: Privacy risks and approaches," in Risk and Security of Internet and Systems (CRiSIS), 2012 7th International Conference on. IEEE, 2012, pp. 1–9.

[10] C. Landwehr, D. Boneh, J. C. Mitchell, S. M. Bellovin, S. Landau, and M. E. Lesk, "Privacy and cybersecurity: The next 100 years," Proceedings of the IEEE, vol. 100, no. Special Centennial Issue, 2012, pp. 1659–1673.

 ISBN: 978-1-61208-376-6

[11] H. Lauterpacht, "Universal declaration of human rights, the," Brit. YB Int'l L., vol. 25, 1948, p. 354.

[12] H. R. Council, "The promotion, protection and enjoyment of human rights on the internet (a/hrc/20/l.13)," 2012, retrieved: February, 2014. [Online]. Available: http://ap.ohchr.org/documents/alldocs.aspx?doc_id=20280

[13] C. Diaz and S. Gürses, "Understanding the landscape of privacy technologies," Extended abstract of invited talk in proceedings of the Information Security Summit, 2012, pp. 58–63.

[14] E. Directive, "95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data," Official Journal of the EC, vol. 23, no. 6, 1995, retrieved: February, 2014. [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML

[15] U. S. Congress, "Health insurance portability and accountability act of 1996," 1996, retrieved: February, 2014. [Online]. Available: http://www.gpo.gov/fdsys/pkg/PLAW-104publ191/html/PLAW-104publ191.htm

[16] U. Congress, "Gramm-leach-bliley act," 1999, retrieved: February, 2014. [Online]. Available: http://www.gpo.gov/fdsys/pkg/PLAW-106publ102/html/PLAW-106publ102.htm

[17] U. S. F. T. Commission, "Children's online privacy protection rule," 2013, retrieved: February, 2014. [Online]. Available: http://www.gpo.gov/fdsys/pkg/FR-2013-12-20/html/2013-30293.htm

[18] C. Civil, "Lei nº12.965, de 23 abril de 2014," 2014, retrieved: July, 2014. [Online]. Available: http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm

[19] D. W. Chadwick, "Federated identity management," in Foundations of Security Analysis and Design V. Springer, 2009, pp. 96–120.

[20] M. Hansen, A. Schwartz, and A. Cooper, "Privacy and identity management," Security & Privacy, IEEE, vol. 6, no. 2, 2008, pp. 38–45.

[21] E. Bertino and K. Takahashi, Identity Management: Concepts, Technologies, and Systems. Artech House, 2011.

[22] T. Orawiwattanakul, K. Yamaji, M. Nakamura, T. Kataoka, and N. Sonehara, "User-controlled privacy protection with attribute-filter mechanism for a federated sso environment using shibboleth," in P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 International Conference on. IEEE, 2010, pp. 243–249.

[23] Shibboleth, "What's shibboleth?" retrieved: July, 2014. [Online]. Available: https://shibboleth.net/about/

[24] SWITCH, "uapprove - user consent module for shibboleth identity providers," retrieved: June, 2014. [Online]. Available: https://www.switch.ch/aai/support/tools/uApprove.html

[25] S. Betgé-Brezetz, G.-B. Kamga, M. Ghorbel, and M.-P. Dupont, "Privacy control in the cloud based on multilevel policy enforcement," in Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on. IEEE, 2012, pp. 167–169.

[26] D. W. Chadwick and K. Fatema, "A privacy preserving authorisation system for the cloud," Journal of Computer and System Sciences, vol. 78, no. 5, 2012, pp. 1359–1373.

[27] W. Jansen, T. Grance et al., "Guidelines on security and privacy in public cloud computing," NIST special publication, vol. 800, 2011, p. 144.

[28] C. Alliance, "Security guidance for critical areas of focus in cloud computing v3. 0," Cloud Security Alliance, 2011.

[29] Q. Zhang, Y. Qi, J. Zhao, D. Hou, T. Zhao, and L. Liu, "A study on context-aware privacy protection for personal information," in Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on. IEEE, 2007, pp. 1351–1358.

[30] T. Mather, S. Kumaraswamy, and S. Latif, Cloud security and privacy: an enterprise perspective on risks and compliance. " O'Reilly Media, Inc.", 2009.

[31] W. Han-zhang and H. Liu-sheng, "An improved trusted cloud computing platform model based on daa and privacy ca scheme," in Computer Application and System Modeling (ICCASM), 2010 International Conference on, vol. 13, Oct 2010, pp. V13–33–V13–39.

[32] R. Terrell, "An easy-to-use encryption system utilizing rsa and aes for javascript." 2012, retrieved: May, 2014. [Online]. Available: https://github.com/wwwtyro/cryptico

[33] Pidder, "pidcrypt – a javascript crypto library." retrieved: May, 2014. [Online]. Available: https://www.pidder.de/pidcrypt/

[34] T. Wu, "Rsa and ecc in javascript." 2009, retrieved: May, 2014. [Online]. Available: http://www-cs-students.stanford.edu/~tjw/jsbn/

[35] E. Birrell and F. B. Schneider, "Federated identity management systems: A privacy-based characterization," IEEE security & privacy, vol. 11, no. 5, 2013, pp. 36–48.

 ISBN: 978-1-61208-376-6

Enforcing Security Policies on Choreographed Services using Rewriting Techniques

Karim Dahmani
karim.dahmani@fst.rnu.tn

Mahjoub Langar
mahjoub.langar@ift.ulaval.ca

LIP2 Research Laboratory
Faculté des Sciences de Tunis
Tunis, Tunisia

***Abstract*—This paper presents an automated formal approach for enforcing security policies on a choreography of Web Services. We take as input a formal description of a choreography of web services and a security property represented by a process, then we define some rewriting rules and rewrite the two processes in order to make them synchronize on each communication action. This approach produces as output a secure version of the concerned web service which behaves like the original one but does not violate the security property.**

***Keywords*-Web Service Composition Security; Instrumentation; Choreography; Formal Verification; End-Point Calculus.**

I. INTRODUCTION

Web Services (WS) are distributed and modular applications that communicate by message passing in order to complete specific activities. Composition of WS consists in combining different WS to provide value-added services. WS composition rules deal with how different services are composed into a coherent global service. In particular, they specify the order in which services are invoked, and the conditions under which a certain service may or may not be invoked. Among the approaches investigated in service composition, we distinguish orchestration and choreography. The orchestration composes available services and adds a central coordinator (the orchestrater) which is responsible for invoking and composing the single sub-activities. However the second one, referred to as WS choreography, does not assume the exploitation of a central coordinator but rather defines complex tasks via the definition of the conversation that should be undertaken by each participant. Several proposals exist for orchestration and choreography languages such as Business Process Execution Language (BPEL) [1] for orchestration and Web Service Choreography Description Language (WS-CDL) [2] for choreography. Since the orchestration technique uses a central coordinator that composes the available services, it seems trivial to enforce security policies. So the technique that will be used in this paper for composing WS is the choreography. One of the main challenges for researchers in this domain is the formalization of these composition languages. Although, several contributions have been developed in the last decade that formalize WS-CDL such as the Global Calculus (GC) and the End-Point Calculus (EPC) proposed by Carbone et al. [3], the Choreography Language proposed by N. Busi et al. [4], The Choreography Description Language proposed by H. Yang et al. [12] and timed automata proposed by G. Diaz et al. [5]. The formal specification language used in this paper is the End-Point Calculus that has been introduced by Carbone et al. [3]. One of the reasons behind this choice is that the end-point calculus is a modified version of the pi-calculus, so its syntax is more familiar and its expressivity is stronger.

The need of secure WS composition has led to a great interest from researchers in the last decade. In this paper, we propose an automated formal approach for the enforcement of security policies on choreographed services. We take as input a formal description of the behavior of a participant in a choreography and a security property. We define rewriting rules for adding some special actions to processes and security properties in order to ensure synchronization and consequently control the evolution of the behavior of a participant.

This paper is structured as follows: in Section II, we introduce the choreography specification language used in this topic. In Section III we present the security property specification language. Section IV deals with the enforcement approach. The proof of this approach is given in Section V. Related work is in Section VI and conclusion and future work are in Section VII.

II. CHOREOGRAPHY SPECIFICATION LANGUAGE

Carbone et al. [3] have proposed a formal language for specifying a choreography of WS. This language is the GC. It describes behaviors of WS from a global viewpoint. GC is distilled from WS-CDL. Carbone et al. [3] have also proposed a second formal language: the EPC, which specifies behaviors of WS from a local viewpoint. Finally, a projection under some assumptions from GC to EPC have been proposed by Carbone et al. [3], which is called the End-Point Projection (EPP). The language adapted in this paper for formally specifying processes and security properties is EPC.

A. Syntax of the End-Point Calculus

EPC describes the behavior of each participant in the choreography from its end-point view. EPC is a variant of the pi-calculus augmented with the notion of participants and their local states. We present hereafter the syntax and formal semantics of EPC where P,Q range over processes, ch range over service channels, s range over session channels, op_i range over operator names, x range over

 ISBN: 978-1-61208-376-6

variables, *e* range over expressions and *X* range over term variables.

$$P ::= !ch(\tilde{s}).P \mid \overline{ch}(\nu\,\tilde{s}).P \mid s \triangleright \Sigma_i op_i(x_i).P_i \mid \overline{s} \triangleleft op(e).P \mid x := e.P \mid P \oplus Q \mid P|Q \mid if\ e\ then\ P\ else\ Q \mid (\nu\, s)P \mid rec\ X.P \mid 0$$

- $!ch(\tilde{s}).P$ and $\overline{ch}(\nu\,\tilde{s}).P$ represent session initiation. $!ch(\tilde{s}).P$ is used for input and $\overline{ch}(\nu\,\tilde{s}).P$ for output. $!ch(\tilde{s}).P$ says that the service channel *ch*, which is available to public, is ready to receive an unbounded number of invocations, offering a communication via its freshly generated session channels $s \in \tilde{s}$. $\overline{ch}(\nu\,\tilde{s}).P$ is an invocation of a service located at the service channel *ch* and an initiation of a communication session that will occur through session channels $s \in \tilde{s}$. After a session has been initiated between two participants and freshly generated session channels have been shared between them, they can communicate via these channels using the communication constructs.
- $s \triangleright \Sigma_i op_i(x_i).P_i$ is an offer of one of operators op_i and a reception of an expression *e* through the session channel *s* that will be evaluated and stored in the local variable *x*. A participant having this behavior will receive an invocation of one of its operator names op_i and an expression *e*. The value of *e* is saved in its local variable *x*.
  For instance, a seller service receives a confirmation or a cancellation for a purchase :
  $s \triangleright confirmPurchase(x_1).P + s \triangleright cancelPurchase(x_2).0$
- $\overline{s} \triangleleft op(e).P$ sends the expression *e* and invokes operator *op* through the session channel *s*. Indeed, a buyer service requests a quote of a chosen product from a seller through the channel *s*:
  $\overline{s} \triangleleft quoteRequest(e_{product}).P$
  In addition, operator names $op_1, op_2, \ldots$ are invoked by a message sender or offered by a message receiver. Operator names in in-session communications are analogous to methods in objects [3].
- $x := e.P$ is the assignment operator. It is a local operation. It assigns the result of the evaluation of the expression *e* to the variable *x*. For example, the buyer assigns to its variable *x* the value of the quote received from the seller : $x_{quote} := e_{quote}.P$
- *if e then P else Q* is a choice based on the evaluation of the boolean expression *e*. For example, the buyer accepts the quote if it is under 1000
  $if\ e_{quote} < 1000\ then\ \overline{s} \triangleleft accept(e_{quote}).P\ else\ \overline{s} \triangleleft reject(e_{quote}).0$
- $P \oplus Q$ is the non deterministic choice. When the choice of the buyer is arbitrary, it would be written as: $\overline{s} \triangleleft accept(e_{quote}).P \oplus \overline{s} \triangleleft reject(e_{quote}).0$
- $P|Q$ is the parallel composition of processes. For example, a seller that offers his service to buyers should have his service running in parallel for new requesters $!ch_{seller}(s).P|P'|P''|...$ where *P',P'',...* are processes dealing with different buyers.
- $(\nu\, s)P$ expresses the fact that the session channel *s* is local to *P*. It is used to restrict a session channel to be used between only two participants that communicate through it.
- *rec X.P* is the recursion operator used to express repetitive behaviors. For example, a participant having the following behavior will always request quotes until he receives an acceptation $rec\ X.\overline{s} \triangleleft quoteRequest(e). s \triangleright accept.P \oplus s \triangleright reject.X$.
- Finally, 0 is the inaction.

Processes are located within participants. Participants and their composition are called Networks (written *N,M,...*), whose grammar is given by:

$$N ::= A[P]_\sigma \mid N|M \mid (\nu\, s)N \mid \epsilon \quad .$$

For more details about the syntax of EPC, the reader can refer to [3].

B. Semantics of the End-Point Calculus

In order to minimize the number of reduction rules, we define $\equiv$ as the least congruence generated from:

$$P|0 \equiv P$$
$$P|Q \equiv Q|P$$
$$(P|Q)|R \equiv P|(Q|R)$$
$$P \oplus P \equiv P$$
$$P \oplus Q \equiv Q \oplus P$$
$$(P \oplus Q) \oplus R \equiv P \oplus (Q \oplus R)$$
$$(\nu\, s)0 \equiv 0$$
$$(\nu\, s_1)(\nu\, s_2)P \equiv (\nu\, s_2)(\nu\, s_1)P$$
$$((\nu\, s)P)|Q \equiv (\nu\, s)(P|Q) \quad (s \notin fn(Q))$$

$$A[P]_\sigma \equiv A[Q]_\sigma \quad (P \equiv Q)$$
$$A[(\nu\, s)P]_\sigma \equiv (\nu\, s)(A[P]_\sigma)$$
$$M|\epsilon \equiv M$$
$$M|N \equiv N|M$$
$$(M|N)|L \equiv M|(N|L)$$
$$(\nu\, s)\epsilon \equiv \epsilon$$
$$(\nu\, s_1)(\nu\, s_2)M \equiv (\nu\, s_2)(\nu\, s_1)M$$
$$((\nu\, s)M)|N \equiv (\nu\, s)(M|N) \quad (s \notin fn(N))$$

The operational semantics of EPC are given in Figure 1.

- Init shows how two participants initiate a session by sharing new freshly generated session channels $\tilde{s}$. These session channels are restricted to participants *A* and *B* using the binding operator ν .

- Comm explains how a communication is established between two participants: when *B* invokes the operator op_j, which is offered by *A*, and sends the expression e_j, which will be evaluated to value *v* at *A*, then *A* receives it and assigns *v* to its local variable x_j.
- Assign is a local operation. Assignment rule evaluates an expression *e* and assigns the result of this evaluation to variable *x* in *A*, then *A* behaves as *P*.
- IfThenElse evaluates the boolean expression *e* and following the result of this evaluation, it behaves either as P_1 or P_2.
- Par shows the behavior of two concurrent processes.
- Sum shows the alternative choice behavior.
- Rec says that if the process *P*, within which we replace each occurrence of *X* by *rec X.P*, behaves as *P'* then *rec X.P* will behave as *P'*.
- Res restricts the use of session channels $\tilde{s}$ to the process *P* in *A*.

Finally, the following rule says we take the reductions up to the structural rule:

$$\frac{M \equiv M' \quad M' \to N' \quad N' \equiv N}{M \to N} \; Struct - NW$$

C. Example

We consider a simplified version of a travel reservation system. The scenario consists of three participants: a traveler, a travel agent and an airline reservation system. The traveler is planning for taking a trip. Once the traveler selects a trip, he submits his choice to the travel agent. The travel agent checks for seats availability within the airline reservation system and sends back either a trip cancel or a validation. The traveler wants to reserve tickets for this trip by sending payment details to the travel agent. The travel agent now must verify one more time availability of seats. If the seats are still available then the airline reservation system accepts the payment details and sends back to the travel agent tickets of the trip. The travel agent responds to the traveler either by tickets of the trip or by canceling the reservation.

The behavior of the traveler is given in EPC by:

$$\overline{ch_{TA}}(\nu s).s \triangleright ack.\bar{s} \triangleleft orderTrip(e_1).$$
$$s \triangleright cancel.0 \oplus s \triangleright available(x_1).\bar{s} \triangleleft book(e_2).$$
$$s \triangleright cancelBook.0 \oplus s \triangleright tickets(x_2).0$$

The traveler starts by opening a session with the travel agent through the public service channel ch_{TA} and initiates a communication channel *s* through which the communication between the traveler and the travel agent will occur. Then, the traveler receives through *s* an acknowledgment message $s \triangleright ack$. After that, the traveler sends an order trip $\bar{s} \triangleleft orderTrip(e_1)$ with expression e_1 which contains details about the chosen trip. At this point, there are two scenarios: the traveler either receives a cancel request $s \triangleright cancel$ when there are no available seats, or an available message $s \triangleright available(x_1)$ containing details of the trip. In this case, the traveler may book the flight $\bar{s} \triangleleft book(e_2)$. Finally, traveler receives *tickets* message $s \triangleright tickets(x_2)$ if the transaction has succeed, otherwise he will receive a *cancelBook* message .

The behavior of the travel agent is given by:

$$!ch_{TA}(s).\bar{s} \triangleleft ack.s \triangleright orderTrip(x_1).$$
$$\overline{ch_A}(\nu s').s' \triangleright ack.\overline{s'} \triangleleft checkSeat(e_1).$$
$$s' \triangleright noSeats.\bar{s} \triangleleft cancel.0 \oplus$$
$$s' \triangleright seatsOK(x_2).\bar{s} \triangleleft available(e_2).s \triangleright book(x_3).$$
$$\overline{s'} \triangleleft reserve(e_3).$$
$$s' \triangleright reserved(x_4).\bar{s} \triangleleft tickets(e_4).0 \oplus$$
$$s' \triangleright notReserved(x_5).\bar{s} \triangleleft cancelBook.0$$

The travel agent offers his service through ch_{TA} by providing a communication channel s. Once his service is invoked, he sends an acknowledgment message, receives an order trip, then contacts the airline service through its public service channel ch_A. The communication between the airline service and the travel agent occurs through the session channel s'. The travel agent looks for available seats $\overline{s'} \triangleleft checkSeat(e_1)$. If there are no available seats then he receives *noSeats* message $s' \triangleright noSeats$ and he sends a *cancel* message $\bar{s} \triangleleft cancel$ to the traveler. Otherwise he receives a *seatsOK* message $s' \triangleright seatsOK(x_2)$ and sends an *available* message $\bar{s} \triangleleft available(e_2)$ to the traveler. After that, the travel agent receives a *book* message from the traveler $s \triangleright book(x_3)$ and proceeds to flight reservation $\overline{s'} \triangleleft reserve(e_3)$. Depending on seats availability, he receives either a confirmation message $s' \triangleright reserved(x_4)$ or a *notReserved* message $s' \triangleright notReserved(x_5)$. In the first case, he sends tickets $\bar{s} \triangleleft tickets(e_4)$ to the traveler elsewhere he sends a book cancellation $\bar{s} \triangleleft cancelBook.$

The behavior of the airline is given by:

$$!ch_A(s').\overline{s'} \triangleleft ack.s' \triangleright checkSeat(x_1).$$
$$if \; available(x_1) \; then \; \overline{s'} \triangleleft seatsOk(e_1).$$
$$s' \triangleright reserve(x_2).if \; available(x_2) \; then$$
$$\overline{s'} \triangleleft reserved(e_2).0 \; else$$
$$\overline{s'} \triangleleft notReserved(e_3).0$$
$$else \; \overline{s'} \triangleleft noSeats.0$$

The airline service offers his service through the service channel ch_A. Once his service is invoked, he sends an acknowledgment message. Then, he receives a *checkSeat* request. Subject to availability, he responds with a *seatsOK* or *noSeats* message. In the first case, he may receive a seat's booking request $s' \triangleright reserve(x_2)$. At this stage, the airline service checks another time seats availability before finalizing the process by either sending a *reserved* $s' \triangleleft reserved(e_2)$ or $\overline{s'} \triangleleft notReserved(e_3)$ as *notReserved* message.

III. Security Policy Specification Language

In this Section, we introduce the Security Policy Calculus (SPC), a formalism used for describing security policies. SPC is considered as a subset of EPC in the sense that it uses only some operators of EPC. Indeed the operators that are

used in SPC are communication actions, recursion, indeterministic choice and no action. Security policies will be represented by processes that will monitor the execution of an another process from EPC.

security properties that we verify in this paper are safety properties and liveness properties without infinite behavior. The reason behind this choice is that some liveness properties can only be verified statically.

B. Shortcuts

$$\frac{}{A[!ch(\tilde{s}).P \mid P']_{\sigma_A} \mid B[\overline{ch}(\nu\tilde{s}).Q \mid Q']_{\sigma_B} \rightarrow (\nu\tilde{s})(A[!ch(\tilde{s}).P|P|P']_{\sigma_A} \mid B[Q|Q']_{\sigma_B})}\ \text{Init}$$

$$\frac{\sigma_A \vdash e \Downarrow v}{A[s \triangleright \Sigma_i op_i(x_i).P_i|P']_{\sigma_A}|B[\overline{s} \triangleleft op_j\langle e\rangle.Q|Q']_{\sigma_B} \rightarrow A[P_j|P']_{\sigma_A[x_j \mapsto v]}|B[Q|Q']_{\sigma_B}}\ \text{Comm}$$

$$\frac{\sigma_A \vdash e \Downarrow v}{A[x := e.P|P']_{\sigma_A} \rightarrow A[P|P']_{\sigma_A[x \mapsto v]}}\ \text{Assign}$$

$$\frac{\sigma_A \vdash e \Downarrow tt}{A[\text{if } e \text{ then } P_1 \text{ else } P_2|P']_{\sigma_A} \rightarrow A[P_1|P']_{\sigma_A}}\ \text{IfTrue}$$

$$\frac{\sigma_A \vdash e \Downarrow ff}{A[\text{if } e \text{ then } P_1 \text{ else } P_2|P']_{\sigma_A} \rightarrow A[P_2|P']_{\sigma_A}}\ \text{IfFalse}$$

$$\frac{A[P_1|P']_{\sigma_A} \rightarrow A[P_1'|P']_{\sigma_A'}}{A[P_1|P_2|P']_{\sigma_A} \rightarrow A[P_1'|P_2|P']_{\sigma_A'}}\ \text{Par} \qquad \frac{A[P_1|P']_{\sigma_A} \rightarrow A[P_1'|P']_{\sigma_A'}}{A[P_1 \oplus P_2|P']_{\sigma_A} \rightarrow A[P_1'|P']_{\sigma_A'}}\ \text{Sum}$$

$$\frac{A[P[\text{rec } X.P/X]]_{\sigma_A} \rightarrow A[P']_{\sigma_A'}}{A[\text{rec } X.P]_{\sigma_A} \rightarrow A[P']_{\sigma_A'}}\ \text{Rec} \qquad \frac{A[P]_{\sigma_A} \rightarrow A[P']_{\sigma_A'}}{A[(\nu s)P]_{\sigma_A} \rightarrow A[(\nu s)P']_{\sigma_A'}}\ \text{Res}$$

Figure 1

A. Syntax

The syntax of SPC is given by:

$\varphi ::= \overline{s} \triangleleft \oplus op_i.\varphi_i \mid s \triangleright \Sigma op_i.\varphi_i \mid \varphi_1 \oplus \varphi_2 \mid rec\ X.\varphi \mid 0$

- The construct $\overline{s} \triangleleft \oplus op_i.\varphi_i$ expresses the fact that invoking one of operators op_i through session channel *s* is permitted by φ.
- Next we have $s \triangleright \Sigma op_i.\varphi_i$, which allows reception of operators op_i through *s*.
- The indeterministic branching is given by $\varphi_1 \oplus \varphi_2$.
- For representing repeated behaviors, we use the recursion operator and
- finally, 0 denotes the lack of actions.

For describing security properties, we need usually to express the prohibition of executing some actions. In our case, when we want for example to interdict sending operation op_1 through *s* we would write this security property: $\varphi = \overline{s} \triangleleft \underset{i \neq 1}{\oplus} op_i.0$, which says that we can invoke anyone of the operators op_i unless op_1. If we want this behavior to be repeatedly verified we would write $\varphi = rec\ X.\overline{s} \triangleleft \underset{i \neq 1}{\oplus} op_i.X$. The semantics are the same as for EPC since SPC is a subset of EPC.

Usually, we use temporal logics for describing security properties but when using the security policy calculus we also reach our goal of expressing any security property that we want to enforce on the behavior of a WS. Since this approach introduces a dynamic verification of WS, so the

For shortness, we will denote by ϕ_s and $\phi_{\overline{s}}$ the portions of a security property that respectively allows all input (output) interactions through *s*. So $\phi_s = s \triangleright \Sigma op_i(x_i)$ and $\phi_{\overline{s}} = \overline{s} \triangleleft \oplus op_i(e_i)$.

C. Example

In the airline reservation system, it is assumed the travel agent wants to be sure that his service does not send tickets before the reception of payment details. The travel agent receives payment details within the book message $s \triangleright book(x_3)$ and sends tickets within the tickets message $\overline{s} \triangleleft tickets(e_4)$. So we want to ensure that $\overline{s} \triangleleft tickets(e_4)$ does not occur before $s \triangleright book(x_3)$. The security property will be written as follows:

$$rec\ X.\overline{s} \triangleleft \underset{op_i \neq tickets}{\oplus} op_i(e_i).X \oplus s \triangleright \underset{op_i \neq book}{\Sigma} op_i(x_i).X \oplus \phi_{s'}.X \oplus \phi_{\overline{s'}}.X \oplus s \triangleright book(x).(rec\ Y.\phi_s.Y \oplus \phi_s.Y \oplus \phi_{\overline{s'}}.Y \oplus \phi_{s'}.Y)$$

The security property is written using a recursion. The idea is to put after each action different from *book* and *tickets* the recursion variable *X*. Thus, the property will remain invariant when executing these actions. It will evolve only by executing the book message. In this recursion, one of these 4 blocks will be executed:

- $\overline{s} \triangleleft \underset{op_i \neq tickets}{\oplus} op_i(e_i).X$: it prohibits invoking *tickets* operator through *s*, which is shared between the traveler and the travel agent. Any message different from *tickets* can be sent.

- $s \triangleright \sum_{op_i \neq book} op_i(x_i).X$: this block prohibits the reception of *book* operator invocation through the session channel *s*. Any message different from *book* can be received.
- $\phi_{s'}.X$ and $\phi_{\overline{s'}}.X$: all actions are permitted between the travel agent and the airline reservation service. The channel *s'* is a session channel shared between the travel agent and the airline reservation system.
- $s \triangleright book(x).rec\ Y.\phi_s.Y \oplus \phi_s.Y \oplus \phi_{\overline{s'}}.Y \oplus \phi_{s'}.Y$ this block intercepts the invocation of the *book* operator within the travel agent and then we take off the control on *tickets* operator by allowing all actions between the traveler and the traveler agent through *s* and the travel agent and the airline reservation system through *s'* to be executed.

IV. Enforcement Approach

In this Section, we will introduce our enforcement approach using rewriting techniques. This approach consists in adding some special actions to processes representing the behavior of a WS and a security property in order to make them synchronize on each interaction that will occur.

A. Communication Actions of EPC

Communication actions are used in this context to designate interactions of EPC. Interactions of EPC are given by these two constructs: $\overline{s} \triangleleft op(e)$ and $s \triangleright \Sigma op_i(x_i)$. They are distinguished by three criteria:

- session channel (*s*),
- operator name (op_i),
- and direction ($\overline{s} \triangleleft$ or $s \triangleright$).

Indeed, each interaction in EPC occurs through a session channel that have been freshly generated and shared between two participants. Within each interaction an operator is either invoked or offered depending on the direction of the interaction. For instance, $\overline{s} \triangleleft op_1(e)$ is an invocation of the operator op_1 through the session channel *s*, $s' \triangleright op_2(x)$ is a reception of an invocation of the operator op_2 on the session channel *s'*. The goal of this approach is to monitor the execution of interactions inside a choreography. This goal will be achieved by controlling the execution of communication actions of EPC.

B. Synchronization Actions

Synchronization actions are special actions that we add to the process and to the security property enforced on this process in order to ensure the interception of each communication action by the monitor. The idea of using synchronization actions to intercept actions is inspired from [6]. So, given a communication action $\overline{s} \triangleleft op(e)$ (respectively $s \triangleright \Sigma op_i(x_i)$), the corresponding synchronization action is $\overline{\overline{s} \triangleleft op(e)}$ (respectively $\overline{s \triangleright \Sigma op_i(x_i)}$).

C. Rewriting Processes

In order to achieve our goal that consists on enforcing a security property on the behavior of a participant *A* in a choreography, we need to rewrite its process by adding synchronization actions. Informally, we will add before each communication action its corresponding synchronization action. Formal rules for rewriting a process *P* of EPC are:

$$\langle !ch(\tilde{s}).P \rangle := !ch(\tilde{s}).\langle P \rangle$$
$$\langle \overline{ch}(\nu \tilde{s}).P \rangle := \overline{ch}(\nu \tilde{s}).\langle P \rangle$$
$$\langle x := e.P \rangle := x := e.\langle P \rangle$$
$$\langle P \oplus Q \rangle := \langle P \rangle \oplus \langle Q \rangle$$
$$\langle P | Q \rangle := \langle P \rangle | \langle Q \rangle$$
$$\langle if\ e\ then\ P\ else\ Q \rangle := if\ e\ then \langle P \rangle\ else \langle Q \rangle$$
$$\langle rec\ X.P \rangle := rec\ X.\langle P \rangle$$
$$\langle \overline{s} \triangleleft op(e).P \rangle := \overline{\overline{s} \triangleleft op(e)}.\overline{s} \triangleleft op(e).\langle P \rangle$$
$$\langle s \triangleright op(x).P \rangle := \overline{s \triangleright op(x)}.s \triangleright op(x).\langle P \rangle$$

D. Rewriting Security Properties

Rewriting the security property consists on replacing each communication action by its synchronization action. Formal rules for rewriting security properties are:

$$\langle \overline{s} \triangleleft \oplus op_i.\varphi_i \rangle := \overline{\overline{s} \triangleleft \oplus op_i}.\langle \varphi_i \rangle$$
$$\langle s \triangleright \Sigma\, op_i.\varphi_i \rangle := \overline{s \triangleright \Sigma\, op_i}.\langle \varphi_i \rangle$$
$$\langle \varphi_1 \oplus \varphi_2 \rangle := \langle \varphi_1 \rangle \oplus \langle \varphi_2 \rangle$$
$$\langle rec\ X.\varphi \rangle := rec\ X.\langle \varphi \rangle$$

E. Restriction Operator

In order to make the rewritten security property φ and the rewritten process *P* synchronize, we will define an operator of EPC that we call the restriction operator and we denote by $P \backslash \varphi$. The role of this operator is to let the process evolve normally when no communication actions is willing to occur. Before a communication action will occur $P \backslash \varphi$ will intercept its synchronization action and verify if the security property can evolve by executing this synchronization action. If it is the case then *P* and φ execute this synchronization action. Else *P* will block and will not execute any other actions. An another role of this enforcement operator is that it hides synchronizations of *P* and φ for the rest of the choreography. Thus, executions of synchronization actions in EPC will be marked by τ as silent actions. Thus our restriction operator does not affect the evolution of *P* when no synchronization action is willing to occur. $P \backslash \varphi$ must ensure the synchronization of *P* and φ on only synchronization actions.

F. Normal Form of a Process

Every process representing the local behavior of a participant in a WS can be written as an internal sum of processes, which we call the normal form of a process: $P = \bigoplus_{i \in I} a_i.P_i$ where a_i range over atomic actions, *I* is a finite subset of natural numbers, and P_i range over processes.

Atomic actions of EPC are: session initiation request

($\overline{ch}(\nu\,\tilde{s})$), session initiation offer ($!ch(\tilde{s})$), communication input ($s \triangleright op(x)$), communication output ($\overline{s} \triangleleft op(e)$), assignment ($x := e$) and synchronization actions ($\overline{s \triangleright op(x)}$, $\overline{\overline{s} \triangleleft op(e)}$).

G. Simulation

We say that a process P can execute an action a and becomes P' and we write $P \xrightarrow{a} P'$ if, when we write P in its normal form ($P = \oplus_{i \in I} a_i . P_i$), there exists $j \in I$ such that $a_j = a$ and $P_j \equiv P'$ where $\equiv$ is the structural equality defined in the semantics of EPC.

H. Semantics

Reduction rules for making $P \backslash \varphi$ progress when executing synchronization actions are given by:

$$\frac{P \xrightarrow{\overline{\overline{s} \triangleleft op(e)}} P' \quad \varphi \xrightarrow{\overline{\overline{s} \triangleleft op(e)}} \varphi'}{A[P \backslash \varphi]_\sigma \xrightarrow{\tau} A[P' \backslash \varphi']_\sigma} \qquad \frac{P \xrightarrow{\overline{s \triangleright op(x)}} P' \quad \varphi \xrightarrow{\overline{s \triangleright op(x)}} \varphi'}{A[P \backslash \varphi]_\sigma \xrightarrow{\tau} A[P' \backslash \varphi']_\sigma}$$

These rules say that each synchronization action of P will be intercepted by φ and it cannot be executed if φ prohibits it. If the synchronization action can be executed by φ then P becomes silently P' and φ becomes φ' .

I. Example

Consider the airline reservation system case study. We will enforce the security property φ defined in the precedent example on the behavior P of the travel agent defined in the first example of this paper. The rewritten process and security property are:

$$\begin{aligned}
&P = !ch_{TA}(s).\overline{\overline{s} \triangleleft ack}.\overline{s} \triangleleft ack.\overline{s \triangleright orderTrip(x_1)}.\\
&s \triangleright orderTrip(x_1).\overline{ch_A}(\nu\, s').\overline{s' \triangleright ack}.s' \triangleright ack.\\
&\overline{\overline{s'} \triangleleft checkSeat(e_1)}.\overline{s'} \triangleleft checkSeat(e_1).\\
&(\overline{s' \triangleright noSeat}.s' \triangleright noSeat.\overline{\overline{s} \triangleleft cancel}.\overline{s} \triangleleft cancel.0)\\
&\oplus \overline{s' \triangleright seatOk(x_2)}.s' \triangleright seatOk(x_2).\overline{\overline{s} \triangleleft available(e_2)}.\\
&\overline{s} \triangleleft available(e_2).\overline{s \triangleright book(x_3)}.s \triangleright book(x_3).\\
&\overline{\overline{s'} \triangleleft reserve(e_3)}.\overline{s'} \triangleleft reserve(e_3).\overline{s' \triangleright reserved(x_4)}.\\
&s' \triangleright reserved(x_4).\overline{\overline{s} \triangleleft ticket(e_4)}.\overline{s} \triangleleft ticket(e_4).0\\
&\oplus \overline{s' \triangleright notReserved(x_5)}.s' \triangleright notReserved(x_5).\\
&\overline{\overline{s} \triangleleft cancelBook}.\overline{s} \triangleleft cancelBook.0
\end{aligned}$$

$$\varphi = rec\ X.\overline{\overline{s} \triangleleft \oplus_{op_i \neq ticket} op_i(e_i)}.X \oplus \overline{s \triangleright \sum_{op_i \neq book} op_i(x_i)}.X \oplus \langle \phi_{s'} \rangle.X \oplus \langle \phi_{\overline{s'}} \rangle.X \oplus \overline{s \triangleright book(x)}.rec\ Y.\langle \phi_s \rangle.Y \oplus \langle \phi_s \rangle.Y \oplus \langle \phi_{\overline{s'}} \rangle.Y \oplus \langle \phi_{s'} \rangle.Y$$

where $\langle \phi_s \rangle = \overline{s \triangleright \sum_i op(x)}$, $\langle \phi_{\overline{s}} \rangle = \overline{\overline{s} \triangleleft \oplus_i op_i(e_i)}$ and similarly for $\langle \phi_{s'} \rangle$ and $\langle \phi_{\overline{s'}} \rangle$.

$TravelAgent[P \backslash \varphi]$ will use first Init reduction rule to open a parallel session then for each communication action, it will synchronize with φ using the reduction rules of $P \backslash \varphi$ and then communicates using communication reduction rules. We can see easily that this process P satisfies the security property φ .

V. Proof of the Approach

In this Section, we prove the correctness of our theory by defining first a partial order over processes and the satisfaction notion.

A. Definition (Subprocess)

Let P, Q be two processes. We say that P is a subprocess of Q and we write $P \subseteq Q$ if the following condition hold : $P \xrightarrow{a} P' \Rightarrow Q \xrightarrow{a} Q'$ and $P' \subseteq Q'$.

B. Definition (Safe Action, Safe Trace)

A trace ξ of EPC is a sequence of atomic actions executed by a process. An atomic action is said to be *safe* if it is not a synchronization action. A trace is said to be safe if it contains only safe actions.

C. Definition (Progression of P)

We say that a process P can progress by executing some safe actions and a synchronization action a and become Q, and we write $P \xrightarrow{a} \!\!| \, Q$ if it exists a safe trace ξ and a process P' such that $P \xrightarrow{\xi} P'$ and $P' \xrightarrow{a} Q$.

D. Definition (Satisfaction Notion)

We say that a process P satisfies a security property φ and we write $P \approx \varphi$ if for all synchronization action a such that $P \xrightarrow{a} \!\!| \, P'$ we have $\varphi \xrightarrow{a} \varphi'$ and $P' \approx \varphi'$.

E. Theorem

Let P be a process and φ a security property. The following properties hold :

- $P \backslash \varphi \subseteq P$,
- $P \backslash \varphi \approx \varphi$,
- $\forall P' \approx \varphi,\ P' \subseteq P \Rightarrow P' \subseteq P \backslash \varphi$.

F. Proof

- The proof is obtained directly from the reduction rules of our enforcement operator and from the definition of $\subseteq$. Indeed $P \backslash \varphi$ is defined so that it cannot execute any actions that P does not execute it.
- Let a be a synchronization action such that $P \backslash \varphi \xrightarrow{a} \!\!| \, P' \backslash \varphi'$. It exists a safe trace ξ such that $P \backslash \varphi \xrightarrow{\xi} P'' \backslash \varphi$ and $P'' \backslash \varphi \xrightarrow{a} P' \backslash \varphi'$. But executions of synchronization actions by $P'' \backslash \varphi$ are given by In-Sync and Out-Sync rules. Then we have necessarily $\varphi \xrightarrow{a} \varphi'$.
- Let P' be a process satisfying a security property φ such that $P' \subseteq P$. Suppose $P' \xrightarrow{a} P''$. As $P' \subseteq P$ then $P \xrightarrow{a} Q$. If a is a synchronization action then from the hypothesis

$P' \approx \varphi$ we conclude that $\varphi \xrightarrow{a} \varphi'$ and then $P \setminus \varphi \xrightarrow{a} Q \setminus \varphi'$. If a is not a synchronization action then $P \setminus \varphi \xrightarrow{a} Q \setminus \varphi$.

VI. Related Work

Several works have studied the correctness and conformance of composition of WS to security requirements.

A. Baouab et al. [7] show a run-time event-based approach to deal with the problem of monitoring conformance of interaction sequences. When a violation is detected, the program shows errors in dashboards. So the program does not stop before the violation occurred. J. Simmonds et al. [8] formalize sequence diagrams to express WS conversations and security requirements and then translate them to nondeterministic finite automata and generate monitors from NFA. Their WS conversation is extracted from the definition of simple services and so they did not consider the great number of WS conversations that will be provided with the composition of WS. D. Dranidis et al. [9] introduced an approach to verify the conformance of a WS implementation against a behavioral specification, through the application of testing. The Stream X-machines are used as an intuitive modeling formalism for constructing the behavioral specification of a stateful WS and a method for deriving test cases from that specification in an automated way. The test generation method produces complete sets of test cases that, under certain assumptions,

are guaranteed to reveal all non-conformance faults in a service implementation under test. However, this approach only returns nonconformance faults and does not react dynamically against these errors. Furhtermore, L. Ardissono et al. [10] propose a monitoring framework of a choreographed service which supports the early detection of faults and decide whether it is still possible to continue the service.

R. Gay et al. [11] have proposed service automata as a framework for enforcing security policies in distributed systems. They encapsulate the program in a service automaton composed of the monitored program, an interceptor, an enforcer, a coordinator and a local policy.

The interceptor intercepts critical actions and passes them to the coordinator that determines whether the action complies the security policy or not and decides upon possible countermeasures then the enforcer implements these decisions. However the authors do not precise how to detect critical actions. W. She et al. [13] have developped an innovative security-aware service composition protocol with composition-time information flow control, which can reduce the execution-time failure rate of the composed composite services due to information flow control violations. This approach only guarantees that there are no access control violations at execution time but do not guarantee that there are not access control violations at runtime. Jose A. Martìn et al. [14] developed a framework based on the partial model checking technique for statically verifying whether a composition of WS satisfies cryptographic properties such as secrecy and authenticity.

VII. Conclusion and Future Work

The goal of this research is to introduce an automated formal approach for enforcing dynamically security policies on a choreography of WS using the rewriting technique. We used a formal language to express conversations of different participants and to express also security requirements. Then, we have shown how to restrict the progression of participant's behavior in order to satisfy security policies. Future work is concentrated on the optimization of this approach by reducing the number of synchronization actions that have been added to processes.

References

[1] I. Corporation, “Business process execution language for web services bpel-4ws,” http://www.ibm.com/developerworks/library/ws-bpel/, 2002.

[2] N. Kavantzas, D. Burdett, G. Ritzinger, T. Fletcher, and Y. Lafon, “Web services choreography description language version 1.0,” W3C Working Draft, December 2004.

[3] M. Carbone, K. Honda, and N. Yoshida, “Theoretical aspects of communication-centred programming,” Electr. Notes Theor. Comput. Sci., vol. 209, 2008., pp. 125–133.

[4] N. Busi, R. Gorrieri, C. Guidi, R. Lucchi, and G. Zavattaro, “Towards a formal framework for choreography,” in WETICE, 2005, pp. 107–112.

[5] G. D´ıaz, J. J. Pardo, M.-E. Cambronero, V. Valero, and F. Cuartero, “Automatic translation of ws-cdl choreographies to timed automata,” in EPEW/WS-FM, pp. 230–242, 2005.

[6] M. Langar, M. Mejri, and K. Adi, “Formal enforcement of security policies on concurrent systems,” J. Symb. Comput., vol. 46, no. 9, pp. 997–1016, 2011.

[7] A. Baouab, O. Perrin, and C. Godart, “An optimized derivation of event queries to monitor choreography violations,” in ICSOC, 2012, pp. 222– 236.

[8] J. Simmonds et al., “Runtime monitoring of web service conversations,” IEEE T. Services Computing, vol. 2, no. 3, 2009, pp. 223–244.

[9] D. Dranidis, E. Ramollari, and D. Kourtesis, “Run-time verification of behavioural conformance for conversational web services,” in ECOWS, 2009, pp. 139–147.

[10] L. Ardissono, R. Furnari, A. Goy, G. Petrone, and M. Segnan, “Monitoring choreographed services,” in Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, 2007, pp. 283– 288.

[11] R. Gay, H. Mantel, and B. Sprick, “Service automata,” in Formal Aspects in Security and Trust, 2011, pp. 148–163.

[12] H. Yang, X. Zhao, Z. Qiu, G. Pu, and S. Wang, “A formal model for web service choreography description language (WS-CDL),” in 2006 IEEE International Conference on Web Services (ICWS 2006), September 2006, 18-22. Chicago, Illinois, USA, 2006, pp. 893–894.

[13] W. She, I. Yen, B. M. Thuraisingham, and E. Bertino, “Security-aware service composition with fine-grained information flow control,” IEEE T. Services Computing, vol. 6, no. 3, pp. 330–343, 2013.

[14] J. A. Mart´ın, F. Martinelli, I. Matteucci, E. Pimentel, and M. Turuani, “On the synthesis of secure services composition,” in Engineering Secure Future Internet Services and Systems - Current Research, 2014, pp. 140– 159.

 ISBN: 978-1-61208-376-6

Obtaining Strong Identifiers Through Attribute Aggregation

Walter Priesnitz Filho, Carlos Ribeiro

Inesc-id, Instituto Superior Técnico, Universidade de Lisboa
Lisboa, Portugal
Email: {walter.filho, carlos.ribeiro}@tecnico.ulisboa.pt

***Abstract*—The development of services and the demand for resource sharing among users from different organizations with some level of affinity motivate the creation of identity management systems. An identifier can be a single name or a number that uniquely identifies a person, although this is often just a representation of a facet of the person. In a federation, services may require user facets comprised of attributes managed by different identity systems which may then be perceived as two facets of two distinct users and not as belonging to the same user. This problem can be handled by adding a new entity type to the traditional architecture thereby creating links between users from different Identity Providers (IdPs), or by using ontologies in order to establish relations between user attributes from several IdPs. In this paper, we propose a solution consisting of obtaining strong identifiers by combining user attributes within IdPs using direct attribute matching and ontologies. Our application context is the Stork 2.0 Project, an eGovernment Large Scale Project (LSP).**

***Keywords*—Privacy; Identity Management Systems; Attribute Aggregation.**

I. Introduction

The development of services and the demand for resource sharing among users from different organizations with some level of affinity motivate the creation of identity federations. An identity federation features a set of common attributes, information exchange policies and sharing services, allowing for cooperation and transactions between the Federation's members [1].

Although there is no definitive architecture, an identity federation is frequently described as being comprised by: an Identity Provider (IdP), a Relying Party (RP), and a Service Provider (SP) [2]. An IdP is responsible for establishing, maintaining, and securing the digital identity associated with a subject, it may also verify the identity and sign up of that subject. A RP makes transaction decisions based upon receipt, validation, and acceptance of a subject's authenticated credentials and attributes within the Identity System. Relying parties select and trust the identity and attribute providers of their choice, based on risk and functional requirements. Finally, the SP controls the access to the services and resources relying on authorities [3].

An identity is composed by a set of attributes, of which at least one identifies its owner. Although an identifier is often seen as a single name or number that uniquely identifies a person, this is often just a representation of a person's facet, characterizing the person as authorized to access a service (e.g., employer of, member of). Within a federation, this kind of identities are not relevant for authorization, given that different services require different user facets. Therefore, within a federation each person is characterized by a number of attributes that may be combined to create several facets, which are released whenever necessary to SPs. IdPs manage these attributes, releasing them to SPs according to a security policy, often when required by the authenticated user.

Inside a federation there might be services requiring user facets comprised by attributes managed by different identity systems, which is a problem because often those facets are perceived as belonging to different users rather than the same user. Facets composed by attributes managed by different IdPs may be required for functionality reasons (e.g., checking the curriculum vitae of a person with degrees in several different universities) or it might be required just to increase the strength of the identity.

According to [4], a strong identifier is capable of uniquely identifying a subject in a population by minimizing multiplicity (i.e., the size of the subset of subjects that match that identifier) within a group of subjects, thereby improving the quality of the identification attributes. When considering the overall strength of the identifier, in addition to the multiplicity of the identifier, the Assurance Level of an attribute must also be considered. Assurance Levels (ALs) [5] are the levels of trust associated with a credential and depend of several factors, namely associated technology, processes and policy and practice statements controlling the operational environment.

In some cases, in order to build a strong identifier to satisfy a service's requirement it may be necessary to use a larger set of attributes than the ones present in any IdP. However, incorrect merging of attributes could result in credentials of different persons being attributed to a single user if, for instance, they share the same name and birthday or have other matching attributes.

In this paper, we propose a solution to build strong identifiers by combining the users' attributes within IdPs using direct attribute matching and ontologies in order to find correspondences in users' attributes distributed on IdPs.

This paper is structured as follows: Section II describes some recent proposals on attribute merging. Section III describes open issues on building strong identifiers, while Sec-

 ISBN: 978-1-61208-376-6

tion IV presents particular considerations and possibilities for solving the problem. Finally, Section V considers future research and remaining issues, and Section VI concludes the paper.

II. Related Work

The integration of diverse sources of attributes has been the subject of research by several authors [4], [6]–[9]. Several approaches have been proposed to overcome the challenges discussed above. Some of the proposed solutions include: Aggregation Entities, Aggregation with Focus on Privacy, and Ontologies.

A. Aggregation Entities

Aggregation entities are specifically designed and run to aggregate attributes from several sources.

The Linking Service (LS) is a special kind of aggregation entity proposed in [6] and [7]. The LS acts as an intermediary between the IdP and SP creating links, through user interaction, so that attributes that are present in more than one IdP can be linked and used to identify the user of a particular service. The solution proposed by the authors allows the users to safely establish links between their accounts on several IdPs. The LS connects the different identities and also manages the authentication of different IdPs, so that the user is not required to authenticate separately on each IdP.

The work proposed by [8] enriches the Linking Service concept with some privacy properties identified in Federated Identity Management Systems (FIMS). Firstly, an IdP should not be able to profile the users' actions, therefore, direct links between IdPs and SPs are not allowed and direct interaction between IdPs and SPs is prevented by specific services pseudonyms. Secondly, the disclosure of personal information is controlled by multiple parties, preventing that any single entity from compromising user privacy. SPs cannot obtain the users' personal information from IdPs without prior consent of the users.

B. Aggregation with Focus on Privacy

When working with various sources of integrated data one should take into account the mechanisms in use to which control attributes and data sources should have its access released or denied. This briefly very describes a pertinent issue namely the privacy of the users involved in these processes of data source integration and attribute aggregation.

In [10], authors present lookup tables, dictionaries, and ontologies to map vocabularies and customers. They use aggregated zero knowledge proofs of knowledge (AgZKPK) to allow users to prove ownership of multiple attributes of their identity, without disclosing anything else. The proposal features an authentication service architecture (User-SP-IdP) with Registrars (Rs) entities, which store and manage information regarding reliability/strength of the identifying attributes used in their approach.

Another proposal focused on privacy [11] uses an extension to the Oblivious Commitment Based Envelope (OCBE) protocol. The proposed extension is a version of OCBE protocol for equality predicates (Agg-EQ-OCBE) that analyses multiple functions simultaneously without a significant increase in computational cost. The proposed extension also uses less bandwidth compared to the EQ-OCBE.

C. Ontologies

The use of ontologies allows a higher degree of automation in the process of attribute merging/aggregation. Through its application, it is possible to deal with heterogeneity, which is one of the problems related to aggregating data from different sources.

In [12], authors define four classes of heterogeneity: heterogeneity of the system, that occurs due to technical differences between platforms; syntactic heterogeneity, which is related to representation and formats of data; structural heterogeneity, which results from differences in schemas; and semantic heterogeneity, which refers to differences in meaning generated by different vocabularies and terminologies used.

Ontologies are used in order to share and reuse knowledge [13]. In this context, an ontology is a specification used to create ontological commitments, which are agreements to use a certain vocabulary so that it is consistent with the theory specified in that ontology.

In [14], the authors analysed the requirements of the Pan European e-Services and other features related with integration. The analysis applies basic concepts from a generic model of public service of Governance Enterprise Architecture (GEA) and the Web Service Modelling Ontology (WSMO) to the semantic description of e-Services.

In spite of findings related to defining ways of achieving reliable attribute aggregation processes, to solutions providing privacy, and on ontology mapping, a solution that integrates all these properties has yet to be found.

III. Open Issues

Approaches used for attribute aggregation are beneficial with regards to obtaining data from various sources, as they are intended to be. However, there are issues that could be improved in these approaches such as the availability of users data aggregators or the use of a single aggregation point for instance.

According to [10] attributes of strong/reliable identification are those capable of uniquely identifying a subject in a population (low multiplicity and high quality), and weak identification attributes are those that may correspond to several subjects (high multiplicity and low quality). Although, two strong identification attributes may separately be able to uniquely identify a subject, their intersection may form a weak identification attribute set, which is not enough to uniquely identify a subject, and therefore is not enough for merging

 ISBN: 978-1-61208-376-6

two identities. Procedures using different IdPs, weak links, etc. could decrease the confidence level of merged identities.

The solutions presented in [10] relate to the treatment of name heterogeneity, mainly with regards to variations in wording, and restrict the language to English. In more heterogeneous environments using the Lookup Tables, as the authors propose, would not be feasible.

The use of ontologies is an interesting resource that we can use to aggregate users' attributes, but when considering the works mentioned above there is one aspect that must be taken into account: in the solutions presented the use of ontologies was only applied to a small number of databases.

The solutions proposed have not yet been tested in a heavy environment. Thus the proposals present no data showing how they perform in a production environment with multiple IdPs and a large number of users.

IV. Proposed Solution

A. Application Context

Our application context is the Stork Project, which is one of five eGovernment Large Scale Projects (LSP). The LSPs eCodex, epSOS, PEPPOL and SPOCS carry information regarding justice, health care, procurement and generic business processes, respectively, from one Member State (MS) service to the other. These services communicate with each other through a network of gateways. Stork aims to provide a fundamental building block of any application or service: Authentication. From this perspective, the Stork Project may share four different building blocks with the other LSPs: Authentication, Authorization, Electronic Signatures (long term authentication), and Document Credentials (long term authorization).

B. Solution

As mentioned earlier, previous solutions make no attempt to build strong identifiers by merging identities. Or even try to increase the assurance level of the identification process by joining attributes from several IdPs.

The Stork Project aims to be a basic building block for eGovernment services, providing services such as: Authentication, Authorization, etc. Our proposed mechanism will act in the Citizen "Pan European Proxy Service" (C-PEPS), the Stork gateway. C-PEPS takes on the task verifying citizen credentials and obtaining additional data, e.g., from the represented person and mandates. This role also entails three business processes: Authentication on behalf of, Powers (digital signature), and Business Attributes. Each PEPS includes functionalities specific to its Member State, which are typically the interfaces with the local ID providers, national and business attribute providers.

We propose a mechanism, named User Identification Strengthen (UsIdS), which performs an open search through users IdPs finding correspondences in the users' attributes (UAs) in order to improve user identification strength. Through an iterative process with the user, he/she will specify which of the IdP(s) that can be used to authenticate him/her in SP does he/she want to use.

A search performed in all pointed IdPs can find matches that are able to certify, with a greater level of assurance, that a user is, in fact, who he/she claims to be. The greater the number of matches found, the greater the strength of the identifier, both because the number of attributes comprising the identifier becomes bigger, but also because some attributes with low assurance levels are repeated by several IdPs. An overview of our mechanism can be observed in Fig. 1.

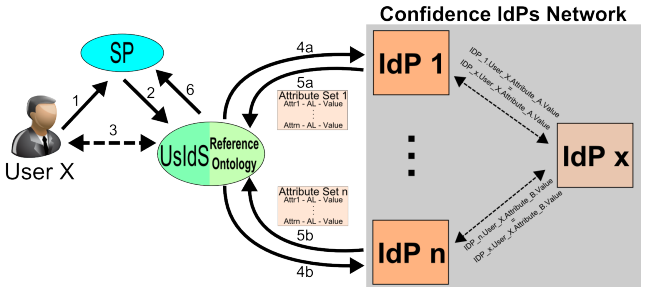


Fig. 1. Proposed search mechanism to find correspondences in user identification attributes

The mechanism assumes, as start point, that user provides a list of IdPs where UAs can be found. As can be observed in Fig. 1, the user sends a service request (Fig. 1 - step 1), indicating the UsIdS as IdP. The SP redirects those instructions to UsIdS (Fig. 1 - step 2). Then, the user sends authentication attributes/authenticates in UsIdS (Fig. 1 - step 3), and attribute set requests are sent to all of the user's IdPs (Fig. 1 - steps 4a, and 4b), the IdPs will then send responses to the UsIdS (Fig. 1 - steps 5a, and 5b) with user attributes sets, each set containing at least the attribute name, Assurance Level (AL), and value. The purpose is to find direct attribute matches, intersections, in attribute sets that can confirm and strengthen the user's identity. The answers received can be handled in two ways depending on the result of the UsIdS analysis of the IdP response.

If an attribute name match is found, the next step is to verify if the attribute values correspond. Otherwise, when attribute names do not match, the next step is to verify on the reference ontology if there is any Ontological Relation (OR) which may established between IdPs involved. If that is the case, attribute values are verified for correspondence. When attribute names do not match, and no ontological relations can be established, UsIdS tries to establish a trusted IdPs network.

To find IdPs, the UsIdS proceeds to search stored ontological relations looking for previously used IdPs. Then a request for user information is sent to that IdPs, and attribute sets are returned in response. UsIdS looks for attribute relations (ARs) between each of the two first IdPs (e.g., IdP_1 and IdP_n) and the new one (e.g., IdP_x). Once an AR is found between i.e., IdP_1 and IdP_x (e.g., $\text{IdP}_1.\text{Attr}_X = \text{IdP}_x.\text{Attr}_X$), the existence

 ISBN: 978-1-61208-376-6

of AR between IdP_n and IdP_x is then is verified. This is repeated until an AR be can found among three, or more, IdPs. When this occurs, the IdP_x attribute set search for presence of any attribute that may be used to improve the strength of the aggregated identity.

When a match is found, the AL of each attribute is verified and the UsIdS sends, as the AL of aggregation, the lowest value within the aggregated value pairs.

In a more schematic way, the process can be seen as follows:

1) Structural Level Verification
   a) With naming conflicts: verifies whether or not similar values, from different user attributes sets, have the same attribute identification.
      i) Reference ontology-based strategy: ontological relations must be established/verified to solve naming conflicts and help find attribute value correspondences.
   b) Without naming conflicts: when there are correspondences in attributes identification names.
2) Verification Matches
   a) Direct matches: a search is performed in the attribute sets that looks for matches in attribute values. i.e.: $\text{Set}_1.\text{Attr}_1.\text{Value}=\text{Set}_2.\text{Attr}_1.\text{Value}$?
   b) Ontological Relation matches: once a UsIdS finds ontological relations (step 1(a)i) it performs a search through those association sets looking for correspondences in attribute values. i.e.: $\text{Set}_1.\text{OR}_1.\text{Value}=\text{Set}_2.\text{OR}_2.\text{Value}$?
   c) Through trusted IdPs network establishment: it is necessary to obtain user' IdPs in order to create such a network. Then, the process restarts from step 1.

Once UsIdS has performed its searches if correspondences were found an indicator of trust on the User Identity is provided to SP (Fig. 1 - step 6).

As previously described, these matches can be through direct attribute matching or obtained from ontological resources. These ontological resources use an ontology-reference based strategy due to the reduced mapping requirements.

As a start point, the ontologies are used to solve Schema-Level conflicts. According to [14], this kind of conflicts involve differences at the structural level of domain models that need to be mapped. The conflicts can be divided into following categories: naming conflicts, entity identifier conflicts, schema-isomorphism conflicts, generalization conflicts, and aggregation conflicts. We will keep our focus on naming conflicts. This type of conflicts arise when similar concepts are labelled in a different way, or when different concepts are labelled in a similar way.

All established ontological relations are stored in C-PEPS, to improve matching performance in the following searches involving the same users and IdPs, although no private data is kept.

When no matches can be found, the UsIdS tries to establish a trusted IdP network path. The purpose of this network is to find data associations with a third or fourth IdP that can be used to establish a relation among the others IdPs. However, finding the necessary IdPs may be a problem. One possible solution is to search in established ontological relations previously stored. It is also possible to ask the user to indicate where the UsIdS may find more attributes that can lead to matches.

C. Privacy

In order to keep users' privacy, we define a protocol considering the model where partners are "honest but curious", or "semi-honest" [15]. This protocol will be used in communications between IdPs and UsIdS to prevent disclosing of user information in the process of trying to find matches; the entities involved should not gain more information than the one authorized by the user. For instance, if for creating a link between two sets of attributes it is necessary to use another attribute that both IdPs know, this linking attribute should only be revealed, to each attribute source, if the attributes match (i.e., the link is possible), otherwise the attribute source would become aware of user private information for which it was not authorized.

The protocol is defined as follows:

Let p and q be two large prime numbers such that q divides $p-1$, G_q be the unique subgroup of $\mathbb{Z}_p^*$ or order q, and g and h be generators of G_q.

Let x, y, and c be random numbers in $\mathbb{Z}_q$.

Let id be the identifier/attribute that both IdPs know but don't want to share.

The identifiers id_1 and id_2 are private within the attributes. The protocol must prove that these two identifiers were generated from the same id but it should not be possible to know the exact id value.

Assuming that both the IdP_1 and the IdP_n follow the protocol:

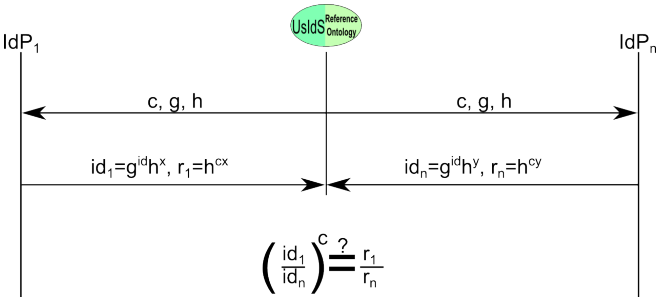


Fig. 2. Proposed privacy preserving protocol

UsIdS sends a challenge c, and generators g and h to both IdP_1 and IdP_n. They reply with a Security Assertion Markup Language (SAML) [16] assertion containing an identifier inside, $id_1 = g^{id}h^x$ and $id_n = g^{id}h^y$, respectively, but these

identifiers are not equal $id_1 \neq id_n$. They also include in response $r_1 = h^{cx}$ and $r_n = h^{cy}$. Finally, the UsIdS must verify that the following equation holds: $(\frac{id_1}{id_n})^c = \frac{r_1}{r_2}$.

Following this privacy protocol, it is possible to verify if the attribute values correspond without disclosing them.

V. Remaining Problems and Future Research

There is still, room for further research on how to apply ontologies in UsIdS i.e., an evaluation of how accurate are ontology mappings. A formal definition of collusion resistance must be specified (UsIdS x IdPs, and SPs x IdPs). Some accuracy validations need to be performed on aggregations in order to verify how efficient the UsIdS is. Once there are prototypes it will be possible evaluate and validate the proposed ideas.

VI. Conclusions

We have proposed a solution to increase the strength of user identifiers by combining facets (i.e., sets of attributes) from several IdPs. The strength of the identifiers results both from an increase in the assurance level of attributes repeated in both sets and an increase of the number of attributes that comprise the combined facet.

Ontologies solve the problem of "Naming Conflicts" that occur when combining sets of attributes. Our chosen Reference Ontology fits our application context (STORK Project), in which there are several languages being used and user data definition on IdPs also has different designations.

Our decision to store ontological relations is due to the fact that the process of establishing these relations could be computationally heavy. So storing the results can improve future searches and can be used to discover IdPs to use in IdPs networks.

The communication process between UsIdS and IdPs uses a privacy protocol in order to assure that user attribute values are not disclosed when IdPs network establishment is being perform. Furthermore, no user attribute values are stored in UsIdS, it just acts as an User Identity Aggregator by relaying IdPs attributes and establishing relations among IdPs and Users.

Acknowledgment

This work was partially supported by CAPES Proc. Num. BEX 9096/13-2 and EU project Stork 2.0 CIP-ICT-PSP-2011-5-297263.

References

[1] S. Carmody, M. Erdos, K. Hazelton, W. Hoehn, B. Morgan, T. Scavo, and D. Wasley, "Incommon technical requirements and information," 2005.

[2] T. W. House. (2009, February) National strategy for trusted identities in cyberspace: Enhancing online choice, efficiency, security, and privacy. URL: http://www.whitehouse.gov/sites/default/files/rss_viewer/NSTICstrategy_041511.pdf [accessed: 2014-09-02].

[3] M. Ates, J. Fayolle, C. Gravier, and J. Lardon, "Complex federation architectures: Stakes, tricks & issues," in *Proceedings of the 5th International Conference on Soft Computing As Transdisciplinary Science and Technology*, ser. CSTST '08. New York, NY, USA: ACM, 2008, pp. 152–157, ISBN: 978-1-60558-046-3, URL: http://doi.acm.org/10.1145/1456223.1456258 [accessed: 2014-09-02].

[4] E. Bertino, F. Paci, and N. Shang, "Keynote 2: Digital identity protection - concepts and issues," in *Availability, Reliability and Security, 2009. ARES '09. International Conference on*, March 2009, pp. lxix–lxxviii, ISBN: 978-1-4244-3572-2, URL: http://dx.doi.org/10.1109/ARES.2009.176 [accessed: 2014-09-02].

[5] "Identity assurance framework: Assurance levels," 2010.

[6] D. Chadwick and G. Inman, "Attribute aggregation in federated identity management," *Computer*, vol. 42, no. 5, pp. 33–40, May 2009, ISSN: 0018-9162, URL: http://dx.doi.org/10.1109/MC.2009.143 [accessed: 2014-09-02].

[7] D. W. Chadwick, G. Inman, and N. Klingenstein, "A conceptual model for attribute aggregation," *Future Gener. Comput. Syst.*, vol. 26, no. 7, pp. 1043–1052, Jul. 2010, ISSN: 0167-739X, URL: http://dx.doi.org/10.1016/j.future.2009.12.004 [accessed: 2014-09-02].

[8] J. Vossaert, J. Lapon, B. Decker, and V. Naessens, "User-centric identity management using trusted modules," in *Public Key Infrastructures, Services and Applications*, ser. Lecture Notes in Computer Science, J. Camenisch and C. Lambrinoudakis, Eds. Springer Berlin Heidelberg, 2011, vol. 6711, pp. 155–170, ISBN: 978-3-642-22632-8, URL: http://dx.doi.org/10.1007/978-3-642-22633-5_11 [accessed: 2014-09-02].

[9] M. Barisch, E. Garcia, M. Lischka, R. Marques, R. Marx, A. Matos, A. Mendez, and D. Scheuermann, "Security and privacy enablers for future identity management systems," in *Future Network and Mobile Summit, 2010*, June 2010, pp. 1–10, ISBN: 978-1-905824-18-2.

[10] F. Paci, R. Ferrini, A. Musci, K. Steuer, and E. Bertino, "An interoperable approach to multifactor identity verification," *Computer*, vol. 42, no. 5, pp. 50–57, May 2009, ISSN: 0018-9162, URL: http://dx.doi.org/10.1109/MC.2009.142 [accessed: 2014-09-02].

[11] N. Shang, F. Paci, and E. Bertino, "Efficient and privacy-preserving enforcement of attribute-based access control," in *Proceedings of the 9th Symposium on Identity and Trust on the Internet*, ser. IDTRUST '10. New York, NY, USA: ACM, 2010, pp. 63–68, ISBN: 978-1-60558-895-7, URL: http://doi.acm.org/10.1145/1750389.1750398 [accessed: 2014-09-02].

[12] V. Kashyap and A. P. Sheth, *Information Brokering Across Heterogeneous Digital Data: A Metadata-based Approach (Advances in Database Systems)*. Springer, 2000, ISBN: 0792378830.

[13] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, Jun. 1993, ISSN: 1042-8143, URL: http://dx.doi.org/10.1006/knac.1993.1008 [accessed: 2014-09-02].

[14] A. Mocan, F. M. Facca, N. Loutas, V. Peristeras, S. K. Goudos, and K. A. Tarabanis, "Solving semantic interoperability conflicts in cross-border e-government services," *International Journal on Semantic Web & Information Systems*, vol. 5, no. 1, pp. 1–47, 2009, DOI: 10.4018/jswis.2009010101, URL: http://dx.doi.org/10.4018/jswis.2009010101 [accessed: 2014-09-02].

[15] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 59–98, 2009, URL: http://repository.cmu.edu/cgi/viewcontent.cgi?article=1004&context=jpc [accessed: 2014-09-02].

[16] "Saml specifications," 2013, URL: http://saml.xml.org/saml-specifications/ [accessed: 2014-09-08].

 ISBN: 978-1-61208-376-6

Wi-Fi Intruder Detection

An experimental approach

Rui Fernandes[1], João N. Matos[1], Tiago Varum[1]
[1]Instituto de Telecomunicações Aveiro
Aveiro, Portugal
ruifelix@ua.pt, matos@ua.pt, tiago.varum@ua.pt

Pedro Pinho[1,2]
[2]Inst. Sup. Eng. Lisboa – ISEL
Lisboa, Portugal
ppinho@deetc.isel.pt

***Abstract* - In a society where monitoring and security are one of the most important concerns, this system represents a convenient and interesting low-cost solution in terms of intruder detection. Using widely spread infrastructure such as Wi-Fi routers and laptops, we proposed an innovative alternative, capable of detecting intruders by sensing the different electromagnetic interference caused. These perturbations are sensed by the system through the changes in the acquired Wi-Fi Received Signal Strength Indicator (RSSI), in the presence of obstacles/targets between the transmitter and receiver.**

Keywords-*Wi-Fi; RF Signature; Wavelet Transform; Intruder Detection; RSSI, Security; Wireless.*

I. INTRODUCTION

A wide number of solutions for intrude detection are available nowadays. From the simple and low cost infrared and Passive Infrared (PIR) sensors [1][2] that detect the heat radiated from the human body, up to the high-end RADAR security [3] systems, a large variety of effective solutions are available to fulfil the various needs of different scenarios.

Among all the mentioned solutions are the requirement to introduce or install extra components in the medium under surveillance. The goal of this work is to propose an innovative and pertinent alternative suitable for modern scenarios.

With a sense of practicality in mind, our system reutilizes the widely spread Wi-Fi infrastructures, taking leverage of easy implementation, turning suitable for both domestic and industrial environments. Utilizing only a standard Wi-Fi router, connected wirelessly to a laptop with dedicated software, this security system can be a simple solution for the actual intruder detection problem. This work mostly tries to show the concept of Wi-Fi intruder detection with results in a controlled scenario. Despite that, some considerations and challenges to adapt this prototype to a real scenario are addressed.

This paper is divided in six sections. The first two sections are dedicated to provide an overview of the develop work and the state of the art applications. The third section, unveils the system operation modules and provides a brief explanation of the concepts of Wavelet Transform and the RSSI in the scope of the designed prototype. The fourth and fifth sections address the experimental set up and the results obtained, with an additional presentation of considerations regarding the set up used and its consequent analysis. Finally, the last section draws some conclusions and indicates the future work proposed by the group.

II. RELATED WORK

In the last decades, with the proliferation of mobile phones and Wi-Fi Access Points (AP), a set of ground breaking applications were developed to demonstrate the large capacity of wireless networks.

An example of this trend is presented by the concept of Wi-Fi localization [3][4][5]. This concept exploits RSSI data from different AP's to reassemble innovative and accurate localization systems, providing an attractive solutions and complement to the Global Positioning System (GPS). So parallel to the development of these applications, studies were conducted focusing on the Received Signal Strength Indicator (RSSI) characteristics and practical concerns [3][4].

Recently, in the monitoring scope, WiSee [6] and WiVi [7] displayed the large tracking detail that can be obtained from Wi-Fi signals when proper signal processing tools are applied.

The WiSee showed the capacity of Wi-Fi based systems to recognize human gestures by extracting the signals frequency Doppler shifts [6]. The WiVi using a Multi Input Multi Output (MIMO) interference nulling [7], detects human movement through walls by the elimination of the static objects reflections.

More in the context of this work, a detection system based on the RSSI was presented with the goal of monitoring pedestrian and automobile traffic [6]. The differentiation of the targets was obtained through the different RSSI changes triggered by the cars and the humans. To achieve this objective a moving mean and variance technique was adopted to analyze the data.

We proposed a less complex system inspired in the previously mentioned works that through the RSSI, senses the alterations of intruders on a static environment. To refine the detection and to possibly avoid false alarms, the Wavelet Transform is applied to the RSSI data. This signal processing technique is characterized to have a time and frequency multiresolution being utilized in diverse image and video processing procedures [9][10][11][12][13].

III. SYSTEM

This section is dedicated to the system characterization. As mentioned before, the RSSI and the Wavelet Transform are the core of the operation principle, so due to their

 ISBN: 978-1-61208-376-6

importance an introduction of these concepts is presented in the following subsections.

A. RSSI

The RSSI is a Radio Frequency (RF) measure, which indicates in dBm, a reference value of the received signal power in the receiver antenna.

Nevertheless, the RSSI is a precise indicator of the received signal strength and quality of the connection in real time, it was proven that the RSSI used on its own needs to pass through a calibration process to overcome the environment factors that influence the signal quality [4][14][15].

The RSSI was addressed in this paper as a measure that indicates the effects on the received signal of the presence of intruders or other targets.

B. Wavelet Transform

The Wavelet Transform is a multiresolution signal processing method capable of adjusting the window length to get a desirable precision in different signal regions, allowing long time windows where low frequency information is needed and short windows for high frequency.

According to Misiti et al. [10], "A wavelet is a waveform of effectively limited duration that has an average value of zero.". However in contrast with the sinusoids, basis of the Fourier analysis, the wavelets tend to have irregularities and "unpredictable" shape.

The Wavelet Transform uses shifted and scaled versions of the main wavelet to separate the signal under analysis. So, the choice of an adequate wavelet is an important step in the analyzing process.

The Wavelet Transform is represented in mathematical terms by:

$$C(a,p) = \int_{-\infty}^{+\infty} f(t)\Psi(a,p,t)dt \qquad (1)$$

where a represents the scale factor and p the shifted position. The Continuous Wavelet Transform (CWT) is a sum over time of the multiplication of the signal with a scaled and shifted version of the main wavelet. Each coefficient evaluates the comparison between the original signal and the wavelet, where the higher the value of the coefficient the more similarities exist between the signal and the wavelet.

In the proposed system, the interference generated by intruders is analyzed applying the Wavelet Transform over the RSSI stream of data. The core of the analysis process comes from the correct choice of a wavelet and scale function, giving great importance and detail to find the best match between wavelet and the pattern to be detected.

The wavelet chosen was the Haar wavelet (step function) with a scale factor of 30. This decision was made taking in consideration the similarities of the step function with the human interference and the better results obtained after several wavelets tested, the ones presented in Figure 1.

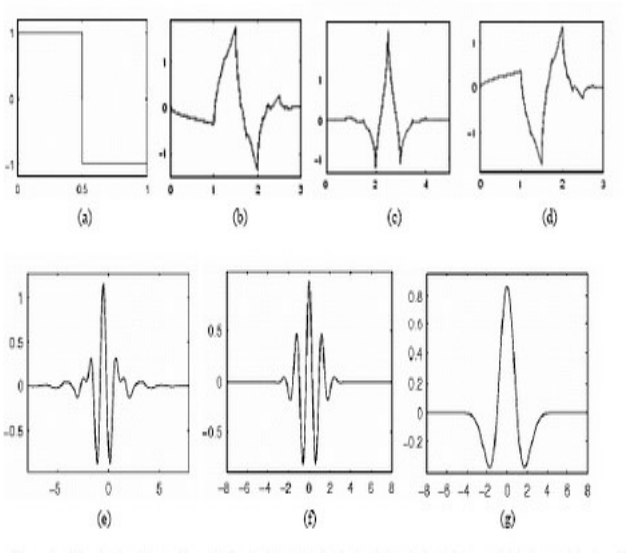


Fig. 1. Example of Wavelet families. (a)Haar, (b) Daubechie4, (c) Coiflet1, (d)Symlet2, (e) Meyer, (f) Moelet, (g) Mexican Hat [8]

C. System Architecture

The system architecture is divided in three interconnected modules: radio, data and processing module.

a) Radio module

The radio module is responsible for emitting and receiving the data using the Wi-Fi protocol. It is constituted by a router in the transmitter side and a laptop with a receiver antenna connected to a network card in the receiver side.

The designed prototype utilizes a Samsung laptop model NP350V5C, an Asus LAN Wireless Router model WL-500n and an external network card from the manufacturer TPLINK, model TL-WN722N (see Figure 4).

This hardware module is responsible for both generating and receiving electromagnetic signals in the 2.4 GHz operation band.

Our system is influenced by the inherent characteristics of wireless communication protocols, being the most relevant the multipath path fading, packets collision and the natural interference from other AP's.

b) Data Module

The data module is both software and hardware based and has the role to be the communication bridge between the radio and processing module. Connected with the PHY layer through the network card, the data module selects the packets applying network filters, discarding packets from undesirable network address. When this filtering process is completed, the stream of RSSI data is sent to the processing module (Figure 2).

The software used was the Microsoft Network Monitor 3.4, responsible for gathering all the RSSI values and selecting the correct network address.

 ISBN: 978-1-61208-376-6

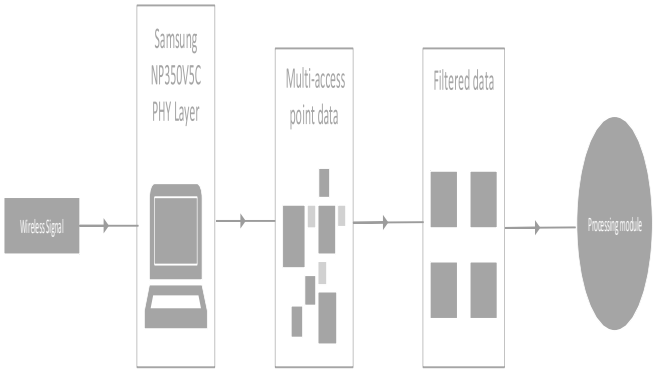


Fig. 2. Simplification of the Data Module operation principle

c) Processing Module

The processing module is completely software based and is implemented in a MATLAB platform (Figure 3). This module has the important task of filtering the noise from the received signals and applying signal processing methods to detect and distinguish the different targets.

The filtering process is simply used to eliminate noise due to multi path and collision components inherent in Wi-Fi connections. This noise appears in the received signals, in the form of notches of one sample duration, with 20 to 30 dB of attenuation, in comparison with the trend of the signal. To filter these undesirable samples, was adopted a simple scheme that detects and discards packets, having always in mind the concern of maintaining the original signal response. Then, the Wavelet coefficients are computed and the human presence is analyzed. The application of the Wavelet Transform also guaranties an additional filtering process, because turns the system insensitive to the variance of quality of the received signal in a wireless channel, having the coefficients values oscillating around zero. This last feature can be very interesting for example to create/design an automatic target identification method. This would be based on the thresholds of the coefficient values generated from different targets, which is very difficult to be implemented directly from the RSSI data.

Connecting all the modules, the system works in the following manner: the radio module generates the signals in the emitter side; the signals propagate in the medium, affected by the intruder presence. When received, the signals are handled in the receiver side of the radio module. Then, the data module gathers from the PHY layer the RSSI data and selects the correct network address, sending posteriorly the data to the processing module. Here, the data is filtered and then the Wavelet Transform is applied. With Wavelet coefficients computed, the data is analyzed with the goal to see if an intruded is detected.

IV. EXPERIMENTAL SET UP

The experiments were elaborated in a domestic indoor scenario. The line of sight between the transmitter and receiver was intentionally clear in a radius of approximately 3 meters. The directional antennas used, were set at a height of 1 meter with 1.8 meters distance to the ceiling (Figure 4). The receiver and transmitter were separated by 3 meters being the targets inserted in half distance, i.e., 1.5 meters (Figure 4).

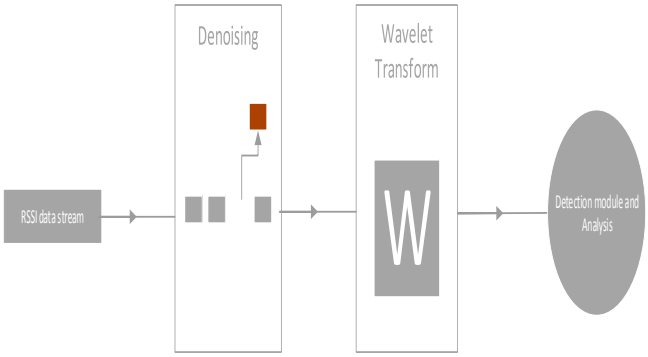


Fig. 3. Simplification of the Data Module operation principle

The half distance was adopted after the testing of several set up's with the targets more close to the Tx or to the Rx. These asymmetrical arrangements proved that the results are dominated by the smaller distance between the target and the Tx or Rx. Also, when the targets are inserted close to the transmitter or receiver end of the system the signals are highly attenuated. The minimum acceptable distance between the target and the Tx or Rx obtained for our system was of 20 cm. In distances below that threshold, the received signal is very low and with large power fluctuations.

The system was also evaluated for more Tx and Rx distances between the 0.2 to 10 meters interval. With this study it was concluded that the distance does not affect much the performance until the 6 meters distance mark, in this controlled environment experiments, generating only a decreased of the mean value of the received signals in order of 1 to 6 dB. After this distance, the presence of the human is more difficult to identify.

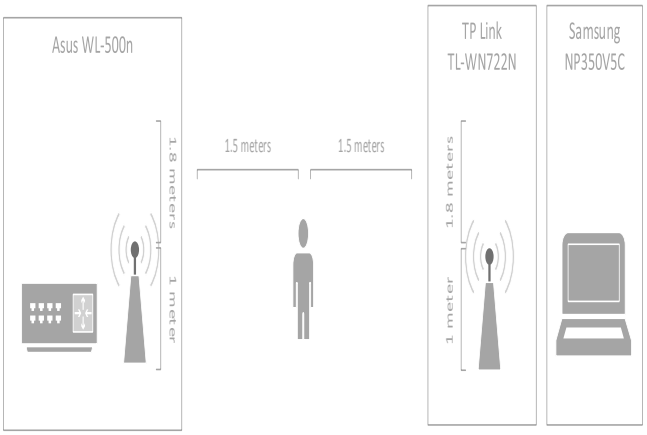


Fig. 4. Schematic of the experimental set up.

Is also worth to mention that the system works with the presence of obstacles in the nearby, with only the special attention to the line of sight Tx and Rx. With the experiments elaborated by the group, it can be preliminary concluded that the presence of objects with similar size to the human body blocking the line of sight influence the system performance, especially metallic objects.

 ISBN: 978-1-61208-376-6

a) Detection Experiments

The system detection performance was tested in two different scenarios: presence of one and two humans.

To avoid false alarms, the response of the system with the presence of domestic animals, in particular cats and dogs, was evaluated. The dimensions of the different targets are presented in Table 1.

TABLE I. TARGET DIMENSIONS

	Human 1	Human 2	Dog	Cat
Height (m)	1.70	1.68	0.68	0.3
Width (m)	0.45	0.40	0.45	0.57

Regarding the testing procedure, the experiments consisted on taking samples of the environment in a silent scenario, for approximately 20 seconds, inserting then the targets during the same interval. In the animal detection, the accuracy of the sampling intervals dropped due to the unpredictable animal reaction.

The angle of detection in the 3 meters experiments was approximately ± 30° in the longitudinal plane.

V. RESULTS AND ANALYSIS

The results are presented in two subsections. The first one evaluates the human detection and the second one the domestic animal response. In both sections, the first graphic shows the RSSI data and the second one the Wavelet Coefficients plot.

A. Human detection

The human detection results are presented in Figures 5 and 6. In both plots, the moments of intruder presence are easily distinguished. Specifically, is visible the attenuation of approximately 10 dB of the received signal in the RSSI data and the increase values of Wavelet coefficients with a consequently oscillation of the pattern.

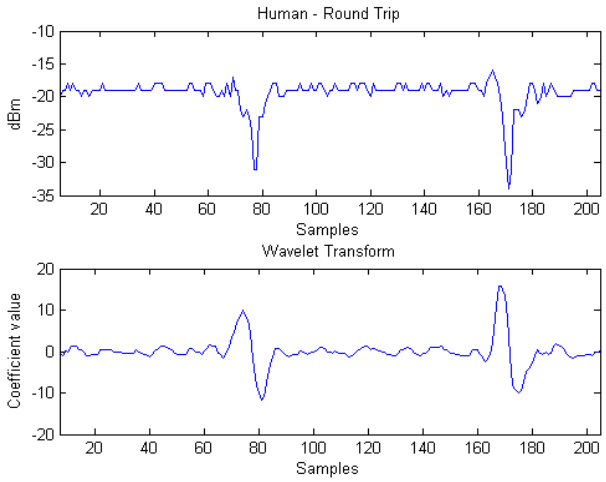


Fig. 5. Human walking; Up) RSSI data; Down) Wavelet coefficients

The experiment with the presence of two humans side by side shown a similar interference in comparison to the single human experiment, presenting only a wider attenuation interval (Figure 6).

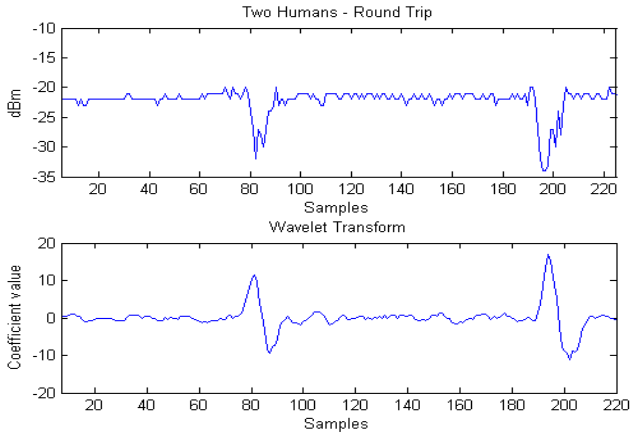


Fig. 6. Two humans walking side by side, Up) RSSI data; Down) Wavelet Coefficients

B. Animal detection

The results from animal detection are presented in Figure 7 and 8. The interference of a dog presented to be smaller in comparison with the human's. Both signal attenuation and Wavelet coefficient alterations are reduced but perceptible in both patterns.

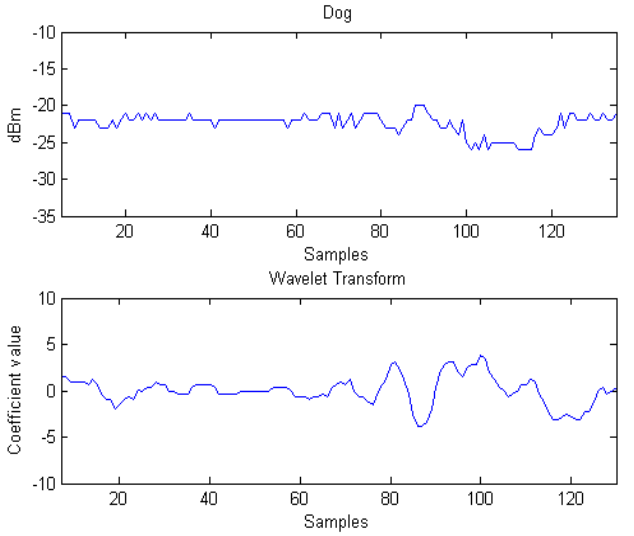


Fig. 7. Dog, Up) RSSI data; Down) Wavelet coefficient

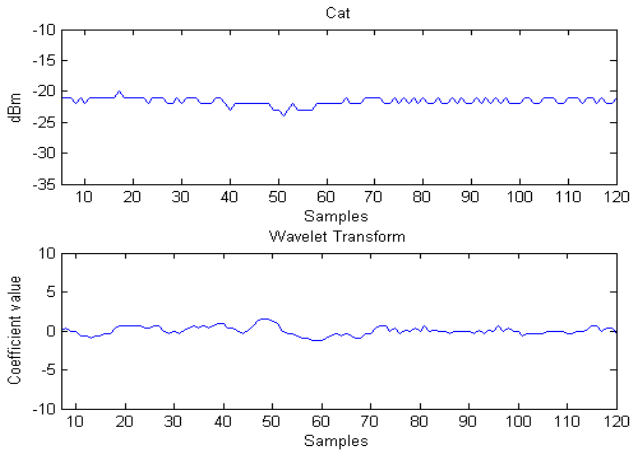


Fig. 8. Cat, Up) RSSI data; Down) Wavelet coefficient

 ISBN: 978-1-61208-376-6

The cat detection results are shown in Figure 8. Due to the smaller dimensions of the cat, principally in height, the signal attenuation is only around 1 to 3 dB which presents to be out of the system detection range.

VI. CONCLUSION AND FUTURE WORK

This work presented an innovative security system able of detecting intruders based on the RF interference generated in a static environment. The proposed Wavelet Transform based technique exhibited a good detection capability and enhanced the target identification performance of the system.

The Wavelet Transform coefficient analysis shows to be a good complement to the RSSI data, with suitable characteristic to improve the system to autonomously identify the targets (recalling the processing module subsection) and possibly avoiding false alarms like neglecting a car or a dog detection.

The Wavelet Transform improvements are also noticeable in the detection of moving targets with significant speed, e.g., running human, where the interference triggered can be mistaken with noise/oscillations in the RSSI raw data. In contrast, the patterns obtained with the Wavelet coefficients are similar to those presented in this paper, where the detections moments are easily seen. Additionally, to enhance the system detection capacity, the adjustment of the scale factor of the Wavelet Transform can be used to detect/neglect smaller RSSI signal interference/patterns. These two last topics are not addressed in detail in this paper because are currently under study, with only preliminary results.

To support the system, an evaluation experiment exposed the different effects on the received signals of the domestic animals presence. The domestic animals proved to have a reduced influence in the system performance, except when the emitter and receivers are very close to the animal (less than 0.75 meters).

The results proved the feasibility and performance of this interesting low-cost solution, achieving in a total of 500 experiments, a 95% human detection in a domestic scenario ratio comparable to other RSSI based systems [5][8]. In [5][8], the authors claim to achieve 100% human detection ratio in similar conditions, i.e., line of sight.

Under study are methods to distinguish different targets more efficiently, the adaptation of the system to perform a real-time detection, the introduction of additional antennas to improve the system coverage area and the use of dual frequency mode available in the 802.11 standard. In terms of propagation, the influence of linear and circular polarized antennas in the system performance are also under analysis.

In the experimental set up it is also under study the influence of the presence of obstacles, the intruder detection outside the line of sight and to conclude, test our system in a real environment to further prove our concept and to isolate the improvements needed.

REFERENCES

[1] T. Yokoishi, J. Mitsugi, O. Nakamura, and J. Murai, "Room occupancy determination with particle filtering of networked pyroelectric infrared (PIR) sensor data," Sensors, 2012 IEEE, October, 2012, pp. 1-6.

[2] Y.W. Bai, Z.H. Li, and Z.L. Xie, "Enhancement of the complement of an embedded surveillance system with PIR sensors and ultrasonic sensors," Consumer Electronics (ISCE), 2010 IEEE 14th International Symposium on, June, 2010, pp. 1-6.

[3] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Buidling RF-based User Location and Tracking System," IEEE INFOCOM, March, 2000, vol.2, pp.775- 784.

[4] M. Saxena, P. Gupta, and B. N. Jain, "Experimental Analysis of RSSI-based Location Estimation in Wireless Sensor Networks," Communication Systems Software and Middleware and Workshops, January, 2008, pp. 503-510.

[5] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, and G. Wang, "I Am the Antenna: Accurate Outdoor AP Location using Smartphones," MobiCom '11, August, 2011, pp. 109-120.

[6] F. Adib and D. Katabi, "See through walls with WiFi!," ACM SIGCOMM Computer Communication, August, 2013,Volume 43 Issue 4, pp.75-86.

[7] Q. Pu, S. G. S. Gollakota and S. Patel, "Whole-home gesture using wireless signals," MobiCom '13, October, 2013, pp. 27-38.

[8] A. Al-Husseiny and M. Youssef, "RF-based Traffic Detection and Identification," Vehicular Technology Conference (VTC Fall), IEEE, September, 2013, pp. 1-5.

[9] A. N. Akansua, W. A. Serdijn, and I. W. Selesnick, "Emerging applications of wavelets: A review," Physical Communication 3, Elsivier, March, 2010, pp.1-8.

[10] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, Wavelet Toolbox™ 4,User's Guide, The MathWorks, Inc., 2009.

[11] S. Arivazhagan and R. N. Shebiah, "Object Recognition Using Wavelet Based Salient Points," The Open Signal Processing Journal 2, December, 2009, pp. 14-20.

[12] Y. Jin, E. Angelini, and A. Laine, "Wavelets in Medical Image Processing: Denoising, Segmentation, and Registration ", International Topics in Biomedical Engineering,Springer US, January, 2005, pp. 305-358.

[13] J. N. Bradley, C. M. Brislawn, and T. Hopper, "FBI wavelet/scalar quantization standard for gray-scale fingerprint image compression," Visual Information Processing II, June, 1993, p. 293.

[14] X. Li, J. Teng, D. X. Qiang Zhai, Junda Zhuy, and Y. F. Zhengy, "EV-Human: Human Localization via Visual Estimation of Body Electronic Interference" Proceedings of INFOCOM 2013, April, 2013, pp. 500-504.

[15] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo, "Self-Mapping in 802.11 Location Systems," Intel Research Seattle, Seattle, 2005, pp. 87-104.

 ISBN: 978-1-61208-376-6

Adding Secure Deletion to an Encrypted File System on Android Smartphones

Alexandre Melo Braga, Alfredo H. Gallinucci Colito
Centro de Pesquisa e Desenvolvimento em Telecomunicações (Fundação CPqD)
Campinas, São Paulo, Brazil
{ambraga,acolito}@cpqd.com.br

***Abstract*—Nowadays, mobile devices are powerful enough to accomplish most of the tasks previously accomplished only by personal computers; that includes file management. However, on many devices the file deletion operation misleads the user into thinking that the file has been permanently removed, when that is usually not the case. Also, with the increasing use of encryption, attackers have been directed to weaker targets. One of them is the recovery of supposedly deleted data from flash memories. This paper describes a way to integrate secure deletion technologies in an encrypted file system in Android smartphones.**

Keywords-secure delete; secure storage; encrypted file system; flash memory; mobile devices; Android.

I. INTRODUCTION

Nowadays, many users keep their sensitive data on mobile devices. However, mobile devices are vulnerable to data leakage. As the amount of digital data grows, so does the theft of sensitive data through loss of device, exploitation of vulnerabilities or misplaced security controls. Sensitive data may also be leaked accidentally due to improper disposal or resale of devices.

With the increasing use of encryption systems, an attacker wishing to gain access to sensitive data is directed to weaker targets. One possible attack is the recovery of supposedly erased data from internal storage, possibly a flash memory card. To protect the secrecy of data during its entire lifetime, encrypted file systems must provide not only ways to securely store, but also reliably delete data, in such a way that recovering them from physical medium is almost impossible.

The new generations of mobile devices are powerful enough to accomplish most of the tasks previously accomplished only by personal computers. That includes file management operations (e.g., create, read, update, and delete). Also, today's devices possess operating systems that are hardware-agnostic by design and abstract from ordinary users all hardware details, such as writing procedures for flash memory cards.

Additionally, it is a real threat the misuse by intelligence agencies of data destruction standards as well as embedded technologies, which can suffer from backdoors or inaccurate implementations, in an attempt to facilitate unauthorized access to supposedly deleted data. In fact, there is a need for practical security technologies that work at the operating system level, under the control of the user. This technology has to be easy to use in everyday activities and easily integrated into mobile devices with minimal maintenance and installation costs.

This paper describes a way to integrate secure deletion technologies to an encrypted file system in Android smartphones. This work is part of an effort to build security technologies into an integrated framework for mobile device security [1][2].

The remaining parts of the text are organized as follows. Section II offers background information. Section III discusses related work. Section IV details the proposed integration of encrypted file systems and secure deletion functions. Section V presents a performance evaluation for the secure deletion function. Section VI discusses improvements on the proposed approach. Section VII concludes this text.

II. BACKGROUND

Traditionally, the importance of secure deletion is well understood by almost everyone and several real-world examples can be given on the subject: sensitive mail is shredded; published government information is selectively redacted; access to top secret documents ensures all copies can be destroyed; and blackboards at meeting rooms are erased after sensitive appointments.

In mobile devices, that metaphor is not easily implemented. All modern file systems allow users to "delete" their files. However, on many devices the remove-file command misleads the user into thinking that her file has been permanently removed, when that is not the case. File deletion is usually implemented by unlinking files, which only changes file system metadata to indicate that the file is "deleted"; while the file's full contents remain available in physical medium. This simple procedure is called logical or ordinary deletion.

Unfortunately, despite the fact that deleted data are not actually destroyed in the device, logical deletion has the additional drawback that ordinary users are generally unable to completely remove her files. On the other hand, advanced users or adversaries can easily recover logically deleted files.

Deleting a file from a storage medium serves two purposes: (i) it reclaims storage to operating system and (ii) ensures that any sensitive information contained in the file becomes inaccessible. The second purpose requires that files are securely deleted.

Secure data deletion can be defined as the task of deleting data from a physical medium so that the data is

 ISBN: 978-1-61208-376-6

irrecoverable. That means its content does not persist on the storage medium after the secure deletion operation.

Secure deletion enables users to protect the confidentiality of their data if their device is logically compromised (e.g., hacked) or stolen. Until recently, the only user-level deletion solution available for mobile devices was the factory reset, which deletes all user data on the device by returning it to its initial state. However, the assurance or security of such a deletion cannot be taken for granted, as it is highly dependent on device's manufacturer. Also, it is inappropriate for users who wish to selectively delete data, such as some files, but still retain their address books, emails and installed applications.

Older technologies [14] claim to securely delete files by overwriting them with random data. However, due the nature of log-structured file systems used by most flash cards, this solution is no more effective than logically deleting the file, since the new copy invalidates the old one but does not physically overwrite it. Old secure deletion approaches that work at the granularity of a file are inadequate for mobile devices with flash memory cards.

Today, secure deletion is not only useful before discarding a device. On modern mobile devices, sensitive data can be compromised at unexpected times by adversaries capable of obtaining unauthorized access to it. Therefore, sensitive data should be securely deleted in a timely fashion.

Secure deletion approaches that target sensitive files, in the few cases where it is appropriate, must also address usability concerns. A user should be able to reliably mark their data as sensitive and subject to secure deletion. That is exactly the case when a file is securely removed from an encrypted file system.

On the other hand, approaches that securely delete all logically deleted data, while less efficient, suffer no false negatives. That is the case for purging techniques.

III. RELATED WORK

This section briefly describes related work on the subjects of secure deletion and encrypted file systems on mobile devices, particularly Android.

The use of cryptography as a mechanism to securely delete files was first discussed by Boneh and Lipton [6]. Their paper presented a system which enables a user to remove a file from both file system and backup tapes on which the file is stored, just by forgetting the key used to encrypt the file.

Gutman [14] covered methods available to recover erased data and presented actual solutions to make the recovery from magnetic media significantly more difficult by an adversary. In fact, the paper covered only magnetic media and, to a lesser extent, RAM. Flash memory barely existed at the time it was written, so it was not considered by him.

Kyoungmoon et al. [12] proposed an efficient secure deletion scheme for flash memory storage. This solution resides inside the operating system and close to the memory card controller.

Diesburg and Wang [16] presented a survey summarizing and comparing existing methods of providing confidential storage and deletion of data in personal computing environments, including flash memory issues.

Wang et al. [19] present a FUSE (File-system in USErspace) encryption file system to protect both removable and persistent storage on devices running the Android platform. They concluded that the encryption engine was easily portable to any Android device and the overhead due to encryption is an acceptable trade-off for achieving the confidentiality requirement.

Reardon et al. [7]-[10] have shown plenty of results concerning both encrypted file system and secure deletion. First, Reardon et al. [11] proposes the Data Node Encrypted File System (DNEFS), which uses on-the-fly encryption and decryption of file system data nodes to efficiently and securely delete data on flash memory systems. DNEFS is a generic modification of existing flash file systems or controllers that enables secure data deletion. Their implementation extended a Linux implementation and was integrated in Android operating system, running on a Google Nexus One smartphone.

Reardon et al. [7] also propose user-level solutions for secure deletion in log-structured file systems: purging, which provides guaranteed time-bounded deletion of all data previously marked to be deleted, and ballooning, which continuously reduces the expected time that any piece of deleted data remains on the medium. The solutions empower users to ensure the secure deletion of their data without relying on the manufacturer to provide this functionality. These solutions were implemented on an Android smartphone (Nexus One) and experiments have shown that they neither prohibitively reduce the longevity of flash memory nor noticeably reduce device's battery lifetime.

In two recent papers, Reardon et al. [8][9] study the issue of secure deletion in details. First [9], they identify ways to classify different approaches to securely deleting data. They also describe adversaries that differ in their capabilities, show how secure deletion approaches can be integrated into systems at different interface layers. Second [8], they survey the related work in detail and organize existing approaches in terms of their interfaces to physical media. They further present taxonomy of adversaries differing in their capabilities as well as systematization for the characteristics of secure deletion approaches.

More recently, Reardon et al. [10] presented a general approach to the design and analysis of secure deletion for persistent storage that relies on encryption and key wrapping.

Finally, Skillen and Mannan [4] designed and implemented a system called Mobiflage that enables plausibly deniable encryption (PDE) on mobile devices by hiding encrypted volumes within random data on a device's external storage. They also provide [3] two different implementations for the Android OS to assess the feasibility

 ISBN: 978-1-61208-376-6

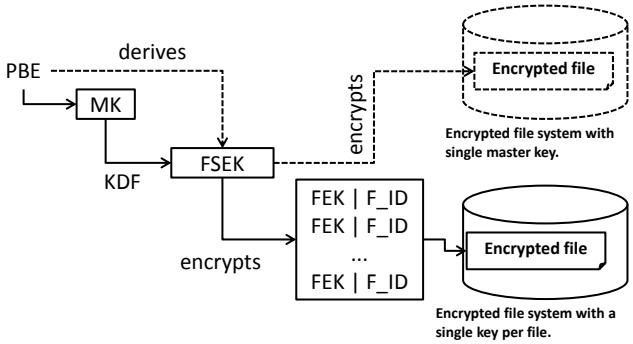


Figure 1. Extending an encrypted file system for secure deletion.

and performance of Mobiflage: One for removable SD cards and other for internal partition for both apps and user accessible data.

IV. DESCRIPTION OF PROPOSED SOLUTION

The rationale behind the proposed solution is the actual possibility of performing secure deletion of files from ordinary Android applications, in user mode, without administrative privileges or operating system customization. The solution handles two cases according to the place where the file already deleted or about to be deleted is stored:

1) The file is already kept by encrypted file system;
2) A file or bunch of files was logically deleted by the operating system and their locations are unknown.

A. Secure Deletion of Encrypted Files

The simplest way to fulfill the task of securely delete a file from an encrypted file system is to simply lose the encryption key of that file and then logically remove the file. This method does not need memory cleaning (purging) and is very fast. A prototype was built upon an Android port for the EncFS encrypted file system [18][19]. To accomplish this task, the way EncFS manages cryptographic keys had to be modified. EncFS encrypts all files with a single master key derived from a password based encryption (PBE) function. It is seams quite obvious that it is not feasible to change a master key and encrypt the whole file system every time a single file is deleted. On the other hand, if each file were encrypted with its own key, then that key could be easily thrown away, turning the file irrecoverable. The modification to EncFS consists in the following steps:

a) Use PBE to derive a master key MK;
b) Use a key derivation function (KDF) to derive a file system encryption key FSEK from MK;
c) Use an ordinary key generation function (e.g., PRNG) to generate a file encryption key FEK;
d) Encrypt files along with their names using FEK and encrypts FEK with FSEK and random IV.
e) Keep a mapping mechanism from FEK and IV to encrypted file (FEK||IV → file).

A simple way to keep that mapping is to have a table file stored in user space as application's data. Care must be taken to avoid accidentally or purposely remove that file when cleaning device's user space. In Android devices, this can be done by rewriting the default activity responsible for deleting application's data. An application-specific delete activity would provide a selective deletion of application's data or deny any deletion at all. The removal from table of the FEK and IV makes a file irrecoverable. The ordinary delete operation then return storage space of that file to operating system. Figure 1 depicts the solution.

Another way to keep track of keys and files is to store the pair {FEK,IV} inside the encrypted name of the encrypted file. In this situation, a file has to be renamed before removed from the encrypted file system. The rename operation destroys the FEK and makes file irrecoverable. The ordinary delete operation then return storage space to operating system.

It is interesting to note that the proposed solution contributes to solve some known security issues of EncFS [13][17]. By using distinct keys for every file, a Chosen Ciphertext Attack (CCA) against the master key is inhibited. Also, it reduces the impact of IV reuse across encrypted files. Finally, it eliminates the watermarking vulnerability, because a single file imported twice to EncFS will be encrypted with two distinct keys and IVs.

Finally, the key derivation function is based upon PBKDF2 standard [5], keys and IVs are both 256 bits, and the table for mapping the pair {key,IVs} to files is kept by an SQLite scheme accessible only by the application.

B. Secure Deletion of Ordinary Files

In this context, a bunch of files were logically deleted by the operating system for the benefit of the user, but they left sensitive garbage in the memory. Traditional solutions of purging memory cells occupied by those files are innocuous, because there is no way to know, from user's point of view, where purging data will be written.

An instance of this situation occurs when a temporary file is left behind by an application and manually deleted. This temporary file may be a decrypted copy of an encrypted file kept by the encrypted file system. Temporary unencrypted copies of files are necessary in order to allow other applications handle specific file types, e.g., images, documents, and spreadsheets.

Whether temporary files will or will not be imported back to the encrypted file system, they have to be securely removed anyway. A premise is that the files to be removed are not in use by any application. The secure deletion occurs in three steps:

1) Logically remove targeted files with ordinary deletion;
2) Write a temporary file of randomized content that occupies all memory's free space;
3) When there is no free space anymore, logically deletes that random file. That action purges all free memory in a way that no sensitive data is left behind.

The final result of this procedure is a flash storage free of sensitive garbage. Steps two and three can be encapsulated as a single function, called memory purging, and performed by an autonomous application. That application would be activated by the user whenever she needs to clean memory from sensitive garbage. The proposed solution adopted this implementation.

 ISBN: 978-1-61208-376-6

Unfortunately, this procedure has two drawbacks. First, it takes time proportional to the size of the free space to be cleaned and the speed of memory writes. Second, this procedure, in the long term, if used with high frequency, have the potential to shorten the lifetime of flash memories.

In order to minimize the negative impact over memory life and avoid excessive delays during operation, steps two and three from above should not be carried out for every single file deleted from the system.

C. Limitations of the solution

The protection of cryptographic keys is of major importance. In spite of being stored encrypted, decrypted just before being used, and then released, the protection of cryptographic keys relies on Android security and the application confinement provided by that operating system.

The proposed solution for memory purging is supposed to work in user-mode, as an ordinary mobile app, without administrative access, with no need for operating system modification, and using COTS devices. These decisions have consequences for security.

First of all, the solution is highly dependent on the way flash-based file systems and controllers behave. Briefly speaking, when the flash storage is updated, the file system writes a new copy of the changed data to a fresh memory block, remaps file pointers, and then erases the old memory blocks, if possible, but not certainly. This constrained design actually enables the alternatives discussed in Section VI.

A second issue is that the solution is not specifically concerned about the type of physical memory (e.g., internal, external SD, NAND, and NOR) as long as it behaves like a flash-based file system. The consequence is that only software-based attacks are considered and physical attacks are out of scope.

Finally, the use of random files is not supposed to have any effect on the purging assurance, but provides a kind of low-cost camouflage for cryptographic material (e.g., keys or parameters) accidentally stored on persistent media. An entropy analysis would not be able to easily distinguish specific random data as potential security material, because huge amounts of space would look random. Of course, this software-based camouflage cannot be the only way to prevent such attacks, but it adds to a defense in depth approach to security at almost no cost.

V. Performance Evaluation of Secure Deletion

Table I shows performance measurements for the secure deletion of ordinary files by purging. The measurements were taken on two smartphones: (i) LG Prada p940h, with 4 GB of internal storage available and Android 2.3.7; and (ii) Motorola Atrix with 16GB (only 11 GB available to final user) of internal storage and Android 2.3.6. File recovery was performed by PhotoRec recovery tool [15]. Random files created for purging had size of at most 2 GB.

Tests were performed over internal memory in three conditions: memory almost free (few files), memory half occupied (many files), and memory free (no files at all). The test procedure consisted of the following steps: (a) creation of ordinary content; (b) logical deletion of that content; (c) execution of secure deletion procedure; and (d) attempting of content recovery. Tests have shown that secure deletion time is proportional to memory size and quite similar to recovery time, as was expected. LG Prada was cleaned at a rate of one Gigabyte per minute (1 GB/min). Motorola Atrix was cleaned at a rate of half Gigabyte per minute (0.5 GB/min). Additionally, a test over a class C SD card of 4 GB was carried out at 0.25 GB/min. In all cases, PhotoRec was unable to recover secure deleted files.

TABLE I. TESTING SECURE DELETION.

LG Prada p940h	Few files	Many files	No files
Free before purging	~3.9 GB	2.21 GB	3.98 GB
Purging time	4min19s	2min37s	4min24s

Motorola Atrix	Few files	Many files	No files
Free before purging	~10 GB	5,2 GB	10,59 GB
Purging time	18min51s	10min53s	19min22s

VI. Improvements under Development

The solution for memory purging is the simplest implementation of a general policy for purging flash memories. In fact, a general solution has to offer different trade-offs among security requirements, memory life, and system responsiveness. The authors have identified three points for customization:

1. The period of execution for the purging procedure;
2. The size and quantity of random files;
3. The frequency of files creation/deletion.

By the time of writing, different trade-offs among the three customization points previously identified were being implemented and evaluated. In all of them, the random file created in order to clean memory space is called bubble, after the metaphor of soap cleaning bubbles over a dirty surface. These alternatives are discussed in next paragraphs.

A. Static single bubble

The solution described in this text implements the idea of a single static bubble that increases in size until it reaches the limit of free space, and then bursts. This solution is adequate for the cases when memory has to be cleaned in the shortest period of time, with no interruption. A disadvantage is that other concurrent application can starve out of memory. This solution is adequate when nothing else is happening, but the purging.

B. Moving or sliding (single) bubble

In this alternative, a single bubble periodically moves itself or slides from one place to another. The moving bubble has size of a fraction of free space. For example, if bubble size is 1/n of free space, the moving bubble covers all free memory after n moves, considering the amount of free space does not change. A move is simply the rewriting of the bubble file, since flash memories will perform a rewrite in a different place.

 ISBN: 978-1-61208-376-6

In a period of time equals to T*(n/2), where T is the time between moves, the chance of finding sensitive garbage in memory is 50%. This solution is adequate when memory has a low to moderate usage by concurrent applications. This solution preserves system responsiveness (usability) but diminishes security.

C. Moving or sliding (multiple) bubbles

This alternative uses more than one bubble instead of a single one. The size and amount of bubbles are fixed. For instance, if bubble size is 1/n of free space, two moving bubble covers all free memory after n/2 moves each. The advantage of this method is to potentially accelerate memory coverage, reducing opportunity for memory compromising.

In the example, two bubbles of size 1/n each can move at every T/2 period, and then concluding in T*n. Alternatively, they can move at period T and terminate in 2*T*n, and so on. This solution is adequate when memory has a moderate usage by concurrent applications. This solution is probabilistic in the sense that as smaller the duration of T and greater the size of bubbles, greater the chance of successfully clean all memory.

D. Sparkling bubbles

This solution varies the size and amount of bubbles. The idea is to create a bunch of mini bubbles that are sparkled over free memory. Bubbles are created and instantly removed at period T, which can be constant or random between zero and T. The sparking of bubbles stops when the sum of sizes for all created bubbles surpasses free space. Bubble size can be small enough to not affect other applications.

This solution is adequate when memory has a moderate to high usage by concurrent applications. This solution is probabilistic in the sense that as smaller the duration of T, greater the chance of successfully clean the whole memory.

VII. CONCLUDING REMARKS

This paper discussed the implementation of two user-level approaches to perform secure deletion of files. One works on secure deletion of encrypted files and the other handles de deletion assurance of ordinary (unencrypted) files. Secure deletion of encrypted files was fully integrated to an encrypted file system and is transparent to the user. Secure deletion of ordinary files was fulfilled by an autonomous application activated under the discretion of the user. Preliminary performance measurements have shown that the approach is feasible and offers a trade-off between time and deletion assurance. Further tests have to be performed to fine-tune the solution in order to preserve system responsiveness. Also, a deep security assessment has to be performed in order to give the actual extend of the security provided by the proposed solution.

ACKNOWLEDGMENT

The authors acknowledge the financial support given to this work, under the project "Security Technologies for Mobile Environments – TSAM", granted by the Fund for Technological Development of Telecommunications – FUNTTEL – of the Brazilian Ministry of Communications, through Agreement Nr. 01.11. 0028.00 with the Financier of Studies and Projects - FINEP / MCTI.

REFERENCES

[1] A. M. Braga, E. Nascimento, and L. Palma, “Presenting the Brazilian Project TSAM – Security Technologies for Mobile Environments”, in proceeding of the 4th International Conference in Security and Privacy in Mobile Information and Communication Systems (MobiSec 2012), LNICST volume 107, 2012, pp. 53-54.

[2] A. M. Braga, “Integrated Technologies for Communication Security on Mobile Devices”, The Third International Conference on Mobile Services, Resources, and Users (Mobility’13), 2013, pp. 47-51.

[3] A. Skillen and M. Mannan, “Mobiflage: Deniable Storage Encryption for Mobile Devices”, IEEE Transactions on Dependable and Secure Computing, vol.11, no.3, May-June 2014, pp.224,237.

[4] A. Skillen and M. Mannan, “On Implementing Deniable Storage Encryption for Mobile Devices”, in 20th Annual Network & Distributed System Security Symposium, February 2013, pp. 24-27.

[5] B. Kaliski, RFC 2898, PKCS #5: Password-Based Cryptography Specification Version 2.0. Retrieved [July 2014] from http://tools.ietf.org/html/rfc2898.

[6] D. Boneh and R. J. Lipton, “A Revocable Backup System”, in USENIX Security, 1996, pp. 91-96.

[7] J. Reardon, C. Marforio, S. Capkun, and D. Basin, “User-level secure deletion on log-structured file systems”, in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, 2012, pp. 63-64.

[8] J. Reardon, D. Basin, and S. Capkun, “Sok: Secure data deletion”, in IEEE Symposium on Security and Privacy, 2013, pp. 301-315.

[9] J. Reardon, D. Basin, and S. Capkun, “On Secure Data Deletion,” Security & Privacy, IEEE , vol.12, no.3, May-June 2014, pp.37-44.

[10] J. Reardon, H. Ritzdorf, D. Basin, and S. Capkun, “Secure data deletion from persistent media”, in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (CCS '13). ACM, New York, NY, USA, 2013, pp. 271-284.

[11] J. Reardon, S. Capkun, and D. Basin, “Data node encrypted file system: Efficient secure deletion for flash memory”, in USENIX Security Symposium, 2012, pp. 333-348.

[12] K. Sun, J. Choi, D. Lee, and S.H. Noh, "Models and Design of an Adaptive Hybrid Scheme for Secure Deletion of Data in Consumer Electronics," IEEE Transactions on Consumer Electronics, vol.54, no.1, Feb. 2008, pp.100-104.

[13] M. Riser, “Multiple Vulnerabilities in EncFS”, 2010. Retrieve [july 2014] from: http://archives.neohapsis.com/archives/fulldisclosure/2010-08/0316.html.

[14] P. Gutmann, "Secure deletion of data from magnetic and solid-state memory," proceedings of the Sixth USENIX Security Symposium, San Jose, CA, vol. 14, 1996.

[15] PhotoRec, Digital Picture and File Recovery. Available [July 2014] from: http://www.cgsecurity.org/wiki/PhotoRec.

[16] S. M. Diesburg and A. I. A. Wang, “A survey of confidential data storage and deletion methods”, ACM Computing Surveys (CSUR), v. 43, n.1, p.2, 2010.

[17] T. Hornby, “EncFS Security Audit”. Retrived [July 2014] from: https://defuse.ca/audits/encfs.htm.

[18] V. Gough, “EncFS Encrypted Filesystem”, stable release 1.7.4 (2010). Available [July 2014] from: http://www.arg0.net/encfs.

[19] Z. Wang, R. Murmuria, and A. Stavrou, “Implementing and optimizing an encryption filesystem on android”. In IEEE 13th International Conference on Mobile Data Management (MDM), 2012, pp. 52-62.

 ISBN: 978-1-61208-376-6

Performance Impacts in Database Privacy-Preserving Biometric Authentication

Jana Dittmann, Veit Köppen
Christian Krätzer, Martin Leuckert, Gunter Saake

Faculty of Computer Science
Otto-von-Guericke University
Email: [jana.dittmann|vkoeppen|
kraetzer|gunter.saake]@ovgu.de

Claus Vielhauer

Department of Informatics and Media
Brandenburg University of Applied Sciences
Email: vielhauer@fh-brandenburg.de

***Abstract*—Nowadays, biometric data are more and more used within authentication processes. These data are often stored in databases. However, these data underlie inherent privacy concerns. Therefore, special attention should be paid for handling of these data. We propose an extension of a similarity verification system with the help of the Paillier cryptosystem. In this paper, we use this system for signal processing in the encrypted domain for privacy-preserving biometric authentication. We adapt a biometric authentication system for enhancing privacy. We focus on performance issues with respect to database response time for our authentication process. Although encryption implicates computational effort, we show that only small computational overhead is required. Furthermore, we evaluate our implementation with respect to performance. However, the concept of verification of encrypted biometric data comes at the cost of increased computational effort in contrast to already available biometric systems. Nevertheless, currently available systems lack privacy enhancing technologies. Our findings emphasize that a focus on privacy in the context of user authentication is available. This solution leads to user-centric applications regarding authentication. As an additional benefit, results using data mining are more difficult to be obtained in the domain of user tracking.**

***Index Terms*—Database Security, Homomorphic Encryption, Privacy, Multi-Computer Scenarios, Database Performance**

I. MOTIVATION

Biometric data are more and more used in daily life. However, these data underlie privacy concerns by design, because these data are directly related to individuals. As a result, this may potentially be misused, e.g., by means of replay attacks, once accessible by malicious parties. Therefore, biometric data require protection mechanisms to take advantage of positive aspects of an authentication scheme. So, privacy-preserving biometric authentication is a requirement that comes into focus of databases, which form the core of any biometric system. In this paper, we present a new approach for user authentication based on the assumption that encrypted data have to be stored and at the same time there is no logging information available.

Although data might be deleted from a database, it can be possible to restore the information partly or even complete. Grebhahn at al. [1] present an approach for deleting data in a database whereas at the same time information could be completely recovered. Although new approaches exist to cover this information or even to improve the system for secure deletion [2], an overall security of traditional database management systems with respect to such information leakage cannot be guaranteed.

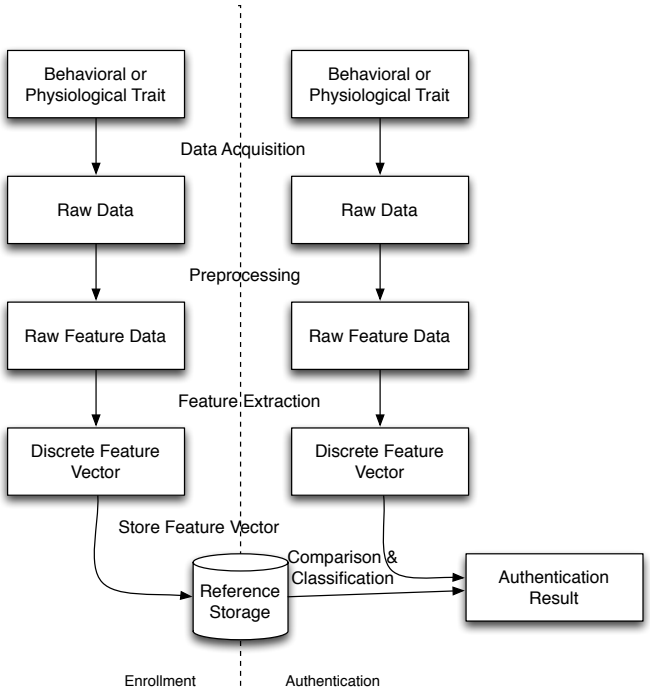


Figure 1. Enrollment and Authentication Pipeline

In a biometric authentication system, two phases are differentiated [3]. Firstly, a user has to create a specific biometric template. In practice, these templates are typically stored in a database. For achieving to store only required information, the data acquisition (e.g., by using sensors) is followed by a data preprocessing to filter out noise and non-related information of the raw data. Note that required information is often depicted in a feature space. Secondly, a feature extraction is applied which is followed by a discretion of the feature values. Finally, the feature vector is stored. This phase is called **enrollment**. We show the basic steps in Figure 1 on the left side.

The second phase is called **authentication**, where a clas-

 ISBN: 978-1-61208-376-6

sification is required to declare an identity of the biometric features. We depict this pipeline on the right side of Figure 1. The first steps from data acquisition to the discrete feature vector should be applied in the same manners as in the enrollment phase. Otherwise, it cannot be guaranteed that the same properties are compared. However, the data for authentication are not stored. In the comparison step, if a one-to-one matching is performed, we call the authentication **verification** [3]. Another classification schema is **identification**, where a biometric discrete feature vector is compared to a set of templates from the database. In both schemes, usually a threshold is used to decide on the success of the authentication.

In case the threshold does not influence the comparison of templates, the result set of an identification can be the closest match, all, k-nearest, or ϵ-distance-neighbors. With these result-sets, further analyzes are possible, e.g., data mining or forensic investigations. Due to complexity, there are several optimization approaches possible. For instance, it is possible to use index structures within the database system for an enhanced data access. However, such index structures need to be carefully optimized for a multi-dimensional feature space, see for further details [4]. Another approach is to preserve privacy in the context of deletion in database index structures as described in [2].

Data mining enables users to detect patterns that are hidden in complex data. With the use of computational techniques, it is also possible to observe and identify relations in the context of privacy preserving scenarios, see for instance [5], [6], or [7].

The work presented in this paper is based on a master thesis [8] and summarizes the main results. We present a methodology based on the Paillier cryptosystem [9] to improve user preferences with respect to authentication systems. We present a cross-evaluation of the impact of homomorphic encryption for biometric authentication using a database within our evaluation section. The Paillier system is an asymmetric cryptographic scheme with additive homomorphic properties. With our new approach, both unique identifiers need to be decrypted for every message. A disclosure of either the key is more unlikely, user-tracing becomes less likely, and the pad do not immediately reveal user content data.

The remainder of this paper is structured as follows: In Section II, we briefly describe the current state of the art regarding our new approach. In Section III, we present the architectural requirements. Our extension of the secure similarity verification is given in Section IV. The evaluation of our approach regarding performance is part in Section V, where we show that response times are accompanied with a small computational effort for privacy preserving aspects. These findings are in line with theoretical considerations and assumptions. Finally, we conclude our results and give a short outlook in Section VI.

II. Background and Related Work

In this section, we present related work for preserving privacy in a biometric authentication context. As important factors, we concentrate on homomorphic encryption as well as deletion in database systems.

The general security requirements for a biometric authentication system are summarized in [10]. Here, it is shown that all security aspects [11] become relevant for all enrollment and verification/identification related components as well as all data transitions between these. Privacy issues are mainly related to confidentiality, but require integrity, authenticity, availability, and non-repudiation of privacy related data. For each security aspect, a security level can also be introduced, e.g., ranging from non, low, up to high.

Security plays a vital role due to different scenarios, in which an attack of personal data is imaginable. A differentiation of attacks can be made on a first level regarding passive or active attacks. The data stream between sender and recipient is not influenced in passive attacks. Therefore, only the reading of data is target for such attacks. Besides just reading data, a specialization is frequency analysis, where for instance for a substitution cipher an analysis of letter frequency is used to identify a mapping. Different extensions are applicable, e.g., frequency attacks or domain attacks [12].

Protection mechanisms for such Biometric reference systems exist since more than a decade; prominent examples are BioHashes [3], Fuzzy Commitment Scheme [13], and Fuzzy Vault [14]. For an overview on challenges for biometric template protection and further current protection schemes see [15]. All these established protection schemes require data to be compared in an unencrypted form, which leads to the threat of information leakage as discussed in Section I. Therefore, these mechanisms are not relevant for the work presented in this paper.

A. Homomorphic Encryption

Homomorphic encryption is used to perform data operations on the cipher text which have a corresponding operation on plain text data. In homomorphic encryption, operations op^* can be performed on encrypted data that are adequate to operations op on the plain text. This means that the following formula holds:

$$op(x) = decryption\left(op^*(encryption\left(x\right)\right). \qquad (1)$$

In such a case, the mapping is structure preserving. The operations op and op^* depend on the cryptosystem. There exist additive and multiplicative homomorphic cryptosystems. Gentry [16] proves the existence of a fully homomorphic encryption scheme. So, it is possible to perform operations on data without possessing a decryption key. However, such systems require high computational effort. In this paper, we make use of homomorphic encryption to perform operations for authentication in an encrypted domain.

B. Verification of Homomorphic Encrypted Signals

Rane et al. [17] [18] developed an authentication scheme with adjustable fault tolerance. This is especially important for noisy sensor data. Due to error correction and similarity

 ISBN: 978-1-61208-376-6

verification, Rane's method can be applied for a wide range of biometric traits.

In their application, three participants are involved for a multi-computer scenario. Whereas the first user provides the biometric signals, the second involved user acts as the central storage server for all biometric templates. The third user is responsible for verification. However, this user is seen as vulnerable and therefore, she is not allowed to query the database system (DBS).

C. Secure Deletion in Databases

Databases can often reveal more information than intended. If an entry is deleted from the data collection, it is a mandatory step to avoid the data reconstruction afterward. Stahlberg et al. [19] and Grebhahn et al. [1] explain how data can be reconstructed from metadata or system copies. Furthermore, DBS specific data, such as index structures, can also be used for reconstruction of deleted data. This means, even if no data are left, the system inherent data structure can be used to gain information from fully deleted data tuples. Therefore, privacy awareness for database tunings, as described in [2], is required for biometric DBS to guarantee data privacy, which is especially challenging for multi-dimensional data [20].

Apart from a possible reconstruction of previously erased data, saved data can reveal additional information. For instance, the amount of queries for a data tuple can give an idea about who that tuple belongs to. This kind of vulnerabilities of the confidentiality needs to be addressed early at the stage of the database layout. Not all security risks can be solved at this stage of the design, but a good database layout can indeed be the foundation of a secure system.

III. Architecture for Privacy-Preserving Authentication

In a general authentication setup, there are two instances that have to share information with each other. There is a participant using a sensor to authenticate a claimed identity on the one side. On the other side, there is a reference DBS containing all enrolled data of all registered users. The DBS is considered to be semi-trustworthy, which means the data in this system shall never be available to the database holder without any kind of restriction or encryption. For that reason, a system allowing database authentication without revealing any information to the database holder needs to be applied. Furthermore, it has to be impossible to decrypt data without having the secret key. The solution used in this paper to address this issue is the use of homomorphic encryption.

Here, we use the Paillier crypto system as described in [9]. We slightly extend this scheme with the inclusion of user-definable key lengths for the purpose of the performance evaluations presented in Section V.

In Figure 2, we present a simplified pipeline of a verification process. Note, in this scenario, a compromised DBS administrator could keep track of the order of enrolled employees and therefore, a sequential ID has to be avoided. This is also conceivable for timestamps and other metadata. So, it is inevitable to disable any logging of enrollment steps.

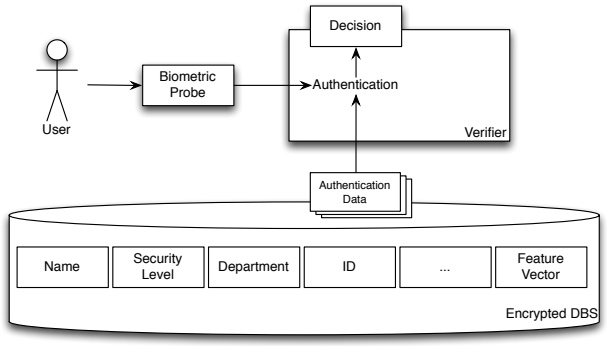


Figure 2. Authentication Process with Encrypted Database, adapted from [8]

IV. Extending Secure Similarity Verification

There exist many biometric authentication systems, which use quite different biometric modalities. Another aspect in this domain is the quality of systems with respect to accuracy and security. To some extent, both properties rely on the trait itself. So, a system that uses only a small set of features with low quality is expected to have overlapping features for different users. Due to the fact that systems often have more than one server and are using different key pairs, user tracking is not possible. Additionally, the order of users can be mixed within different systems. We introduce the padded biometrics approach, which allows user authentications in a multiple participant scenario with respect to privacy-preservation. Additionally, we present performance impacts and a brief security impact discussion.

A. The Padded Biometrics Approach

In Figure 3, we depict a scenario for user tracking with two database systems (DBS). We assume, an attacker has read access to both databases. The differences between both DBS are key pairs and user IDs. Assume, with some knowledge, the attacker identifies in DBS_1 User 1. The DBS uses an unsalted asymmetric encryption which results for a given key and plain text value always in the same cipher value. Within DBS_1, the attacker finds the exact same value for another user (User 5). With the help of this knowledge, both users can be identified in DBS_2. Due to the fact that the feature vectors are not shuffled, the attacker needs to identify a match between two users in DBS_2 with an overlap of the same two features.

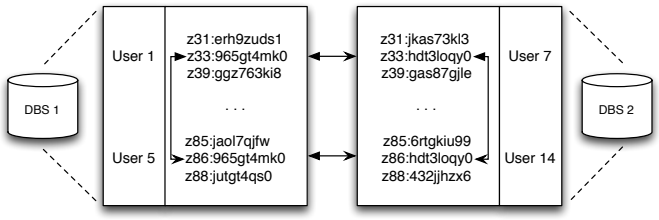


Figure 3. User Tracking in a Multiple DBS Setup, adapted from [8]

In practice, for a proper biometric trait with an appropriate resolution this scenario is implausible. As an example, we

take the iris codes with 2,048 bit representation for the iris features; there exist theoretically more than 10^{74} different codes. However, the Euclidean vector space is very sparsely populated due to cluster of iris codes. Such clustering occurs in many biometric modalities. Therefore, our example, given in Figure 3, is a result from exact matches for different feature vectors. Correlations of biometric features are the main reason for such clusters. Daugman [21] identifies the iris phase code to be 0 or 1. This results in a Hamming distance with a very small variance. Daugman uses 249 different features and obtains $\mu = 0.499$ and $\sigma = 0.0317$. There exist several other analogous examples, e.g., in face recognition for the distribution of eyes, nose, and mouth that are quite similar for every person. We conclude that it is very likely that the data in the feature space are not equally distributed.

With these insights or domain knowledge, it is possible to link users or even track users as in our example in Figure 3. An inclusion of the metadata of the database also enables further possibilities for an information gain, e.g., in the case that an index structure relates similar values, as the R-tree [22] or the Pyramid technique [23].

We propose a padding approach. This is comparable to salting. In Figure 4, we show the idea. Every user receives a specific ID (UID). This ID is encrypted together with the template, e.g., by concatenating ID and biometric feature. This approach also allows including the feature index (FID) in the pad which avoids intra-user overlapping.

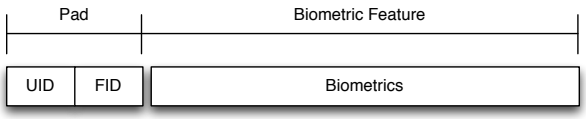


Figure 4. Lead-Pad for Biometric Features, adapted from [8]

The resulting value of a pad and a biometric feature has to be encrypted. A leading pad avoids any inter- and intra-user redundancies. At the same time, the possibility of the above described attack is close to zero. The padding, seen as a security layer, can be either maintained by the user or operated by an additional participant who has paddings and IDs. This proposal comes at the cost that identification is expected to be more difficult. The pad shifts features semantically away from others. Therefore, the Euclidean measurements for similarity cannot be used, but the complete set of pads for each person has to be processed. We concentrate on performance of our proposed approach in the following.

B. Performance of the Pad Approach in the DBS

Index methods are widely used in DBMS to increase performance [24]. In relational databases, the B-tree [25] [26] and variants, such as the B+-tree. are used to achieve a logarithmic lookup performance. A similarity search using B+-trees results on average in a linear performance overhead additionally. Including a verifier, as proposed in an encrypted data domain, influences the processing time due to transportation effort. We discuss pros and cons in the following.

TABLE I. PERFORMANCE IN A DATABASE SYSTEM AND COMPARED TO THE PADDING APPROACH

Query Type	DBS with B+-tree	Padding DBS
Exact Match	$O(log(n))$	$O(n)$
Similarity Search	$O(n)$	$O(n)$

Sorting and the use of metadata, which can improve query response times, should be avoided for security reasons. This requirement is in contrast to typically used index structures in relational data management systems. Therefore, the identification within the authentication process requires linear computational effort. Depending on the size and the application scenario, different metadata, such as gender, can be utilized to limit this effort. Note, if small subsets can be created from this metadata, it is necessary to separate these from biometrics. Alternatively, the padding approach can be applied to non-biometrics, too. In Table I, we summarize the computational efforts for a relational database and also for a database with encryption using our padding approach. Due to several other possible performance impacts, such as database size, feature size, thresholds, or key bit-length, we present in Section V a short evaluation study.

1) Implementation Issues: We propose to use a distance result from the verifier instead of a binary decision of acceptance or decline of an authentication attempt. Besides a reasonable attack scenario, where learning from accepted authentications and repeated authentication queries is possible in the later scenario, this risk can be reduced by disabling repeated authentication. In our approach, the quality of the similarity can be computed in an evaluation step. We apply the following formula:

$$d\left(X,Y\right) = \frac{\sum_{i=1}^{dim} |x_i - y_i|^a}{\tau^a \cdot dim} \tag{2}$$

with threshold τ, $a \geq 1$ as degree of freedom, and dim as dimensionality of the feature vector. These parameters are important for adjusting quality regarding sensor accuracy, error rates, and the biometric trait. The better the quality, the lower can be τ and the larger a.

We use a dictionary to maintain all pads for all enrolled users. The pads are delivered via a secure channel for each authentication process. The pads are concatenated before encryption. Due to the non-existence of relations to personal data, the pads can be generated randomly. The necessary step before enrollment or authentication is adding the pad. Note, it is not necessary to add the pad before the signal. Within an identification process, it is necessary to lookup the dictionary for the pad of a user. If outsourcing the dictionary to an external server, a processing time increase has to be respected.

In the following, we consider the three participant approach, for other system architectures from Section III. We measure the influence of computation time regarding all three involved participants. Note, if participants are embedded, as described in Section III, special security requirements have to be met.

The three participants consist of a user, a verifier, and the

 ISBN: 978-1-61208-376-6

DBS, which maintains the encrypted templates. Biometrics are taken by a sensor at user side. The verifier is responsible for authentication. Note, communication channels can be realized in different ways, such as insecure or with encryption. In the case, that only the user stores all pads to corresponding IDs, verifier and DBS do not need to be fully trusted. Hill-climbing should be avoided and therefore, a repeated authentication from single users has to be disabled. As a result, we can sum up that applying our approach to this scenario, only the user and partly the server gain information on the claimed identity. The ability to learn from the results can only be realized on user or verifier side. There is no plain information, due to encryption at user side within the complete process.

C. Security Impacts

In our experiments, we do not focus on crypto-analyzes. Since data are kept in the DBS, this is a promising entry point to gain information. A careful design and a proper security concept are mandatory. Implementation can cause vulnerabilities to the protocol that can lead to information leakage. There are some attacks, which do not immediately address the protocol. For instance, there are attacks on availability and the endpoint should be carefully considered.

An attacker can try to take advantage of vulnerability that originated from poor system design. For example, a system designer decides to embed the verifier at user side, but does not meet all steps to guarantee confidentiality. If an unauthorized user is able to listen to the verifier, an information leakage occurs.

In the case our padding approach is implemented inappropriate, e.g., without secure separation from unauthorized users, and an attacker gains access to the pads, the confidentiality is at risk. With access to the pads and the encrypted signal, known-plain-text attacks [27] are possible.

The asymmetric Paillier cryptosystem is not information-theoretically secure. Thus, there are threats leading to leakage of the biometric templates in the DBS. We introduce a padding approach to avoid opportunity of such attacks. Note, a secure dictionary is mandatory. The implementation of a system can enable various security vulnerabilities. These enable an attacker to gain trusted information. It is mandatory to implement a proper pseudonymization approach in combination with a secure dictionary.

The configuration of a system is presumably the most promising path for an attacker. The DBS amount and type of meta-information can be a threat to security. System designers have to carefully consider meta-information. Additionally, backups play an important role. With access to both, DB and backup, an attacker subtracts users from backup and current state for user tracking.

Acceptance threshold and quality classes influence *false acceptance* and *false rejection rates*. The threshold decides on size of error patterns. There are many additional factors: level of information confidentiality, quality of the signals, access frequency, expectations regarding response times, and combined biometrics.

If the authentication protocol uses web communication, a denial of service attack (DoS) can disturb the protocol from functioning and harms availability. Even without using the web, there are other possible attacks that are not only taking advantage of communication. For instance, using malware to prevent participants from following the protocol is an imaginable attack on availability. Assuming that a biometric authentication scheme applies the Four Participants scenario, a DoS attack on the disguise would prevent the system's functioning. It is possible to reduce the threat, but impossible to prevent it completely.

Endpoint security is crucial to provide confidentiality, especially if users have access to secret keys. Assuming the secret key is not as easily accessible, an attacker can try to read parts of communications. This includes plain and encrypted data such as pads. Assessing these data, follow-up attacks like known-plain or known cipher text attacks [27] are possible. For a restriction, basic security steps, including anti-virus software and firewalls, should be implemented.

V. Evaluation

In this section, we present evaluation results on performance for our approach. We evaluate processing time as performance metric. For our evaluation, we present experiments regarding different influence factors, such as enrolled users, key length, feature vector dimension, and thresholds. Firstly, we explain the evaluation setting. Secondly, we show results of our performance evaluation with respect to enrolled users, key length, feature dimensions, and threshold by studying with and without-padding approaches and encrypted versus non-encrypted scenarios.

A. Experimental Layout

For our evaluation, we use a MySQL database, version 5.5.27. We restrict our evaluation to a two table layout with index structures as follows:

- *Person*(Name, Security level, Department, ID)
- *Biometrics*(Feature, ID, BID).

Every enrolled person in the system has some attributes, i.e. a name, a security level, and a department. These attributes can be exchanged or extended by any property. In addition, every person has an ID to find a data tuple unambiguously. All properties like name, security level and department are encrypted with the public key. Biometrics are divided by the count of dimensions of the Euclidean vector. Every feature is identified by a biometric ID (BID), while biometrics are assigned to the corresponding person by an ID.

We make the following assumptions: The DBS is designed that it can be used for most common discrete biometric features. The resolution or the quality of the feature has no influence on the operative readiness of the biometric system itself. How accurate the resolution has to be is a question of acceptable error rates and needs to be adjusted by the corresponding use case. Features are saved in feature vectors and have a minimum of at least one dimension and can have as many dimensions as needed. Everything that depends on

 ISBN: 978-1-61208-376-6

the dimension of the feature vector grows corresponding to its size. For example, the codebooks are depending on the size of the feature vector.

B. Performance Evaluation

We perform all experiments on an AMD Phenom II X6 1055T Processor, an SSD, and 8GB RAM. In our evaluation, we focus on response time as crucial performance factor. We apply 10 replications per evaluation run for validity. We use artificial data that we $i.i.d.$ generated from Gaussian distribution.

TABLE II. PERFORMANCE FOR USERS

Users	Identification in ms	Verification in ms
20	35	87
1,000	63	107
100,000	354	354

First, we test for size of enrolled users. Note, for simplicity, feature length is 11 dimensions, key size is 64bit, and threshold is 3. In Table II, we present arithmetic means for identification and verification for our padding approach. Our results indicate that the overall processing increases with a higher amount of enrolled users. This growth seems linear. Memory management and thread scheduling or configuration and running the DBS cause this increase. Since verification only requires data of one person, the increase is not similar to identification. Due to B+-trees in MySQL, there is an increasing impact according to the size of enrolled users.

TABLE III. PERFORMANCE FOR KEY LENGTH

Key Length	Identification in ms	Verification in ms
64	63	107
128	112	87
512	1001	400
1,024	6933	2188

In Table III, we present results regarding key length. Note, we use 1,000 enrolled users in the DBS and a feature dimensionality of 11. As expected, an exponential growth with an increase of the key length is obvious. Due to our experimental setup (using one machine for all tasks), this growth might be influenced in our experimental setup. However, using a private key only increases the processing time in a small amount. A fast feedback is a user requirement for user acceptance of biometric authentication.

We test different feature vector sizes (11, 69, 100, 250, and 2,048) and present the results in Table IV. Adding new features to the feature vectors requires more comparisons, which result in higher response times. Note, with an increase of the feature vector the codebooks also increase. Due to this, the growth in smaller feature vectors can be explained.

As a last evaluation parameter, we vary the threshold from 3 to 1,000 and present our results in Table V. The threshold parameter is used for quality reasons, see Section IV. Compared to [18], increasing the threshold by 1 means that two additional comparisons have to be computed. Therefore, the increase is linear with the number of enrolled users. Signals with a higher fluctuation, which require a larger range of validity, require more processing time. This has to be examined for each application and evaluated regarding hardware, requirements, and accuracy.

TABLE IV. FEATURE DIMENSIONS PERFORMANCE

Feature Dimensions	Identification in ms	Verification in ms
11	63	56
69	239	81
100	354	321
250	693	571
2,048	1,065	860

TABLE V. PERFORMANCE FOR THRESHOLD

Threshold	Identification in ms	Verification in ms
3	125	99
5	199	104
10	216	114
100	280	208
1,000	1,572	1,311

TABLE VI. PERFORMANCE REGARDING THRESHOLD

System Parameters	Padding Approach in ms	Without Pad in ms
1,000 users, 2,048 features, 64 bit	25,546	26,014
1,000 users, 11 features, 1,024 bit	28,033	27,197
100,000 users, 11 features, 64 bit	35,009	35,403

As a concluding remark, we present our evaluation results regarding our approach compared to the approach presented in [18]. In Table VI, we show three different parameter scenarios exemplary. This table shows unexpected results. In the first and third experiment, the response times for the padding approach are slightly lower than without padding. This might be a result from caching and optimizations that take place in the experiments. However, our results show, the influences of our approach are negligible.

TABLE VII. COMPARISON OF SECURE IDENTIFICATION

Enrolled Users	Encrypted Identification	Unencrypted Identification
20	35 ms	26 ms
1,000	63 ms	47 ms
100,000	354 ms	310 ms

In the last setting, we show differences between encrypted and unencrypted identification in Table VII. We use again a key length of 64bit and 11 feature dimensions. The threshold is set to 3. The results show the cost for encryption. Note, we only use a very small computation effort regarding encryption due to a very short key length. With an increase of the key length the difference for both scenarios increases dramatically.

 ISBN: 978-1-61208-376-6

VI. Summary and Outlook

In this paper, we present an extension to the secure and similarity verification between homomorphically encrypted signals by Rane [17], [18]. Tracing users is possible in the original scenario. We present a padding approach, to overcome this challenge. We extend the original contribution to search on encrypted values and to use a one-time-pad-concept. Furthermore, we develop a evaluation study of our conceptual design to evaluate our approach.

With the padding approach, an advanced search in an encrypted domain is possible. However, if repeated authentication attempts are possible, it is already possible to gain information regarding the template. One can avoid such template reproduction by disabling repeated authentications. Our approach improves data security. We name some security requirements for this purpose.

Processing times in our evaluation reveal that our padding approach comes at very low additional cost compared to [18]. This is an important aspect for user acceptance of such a system. Whereas the size of enrolled users has logarithmic impact on computational effort, the key length impacts with an exponential scheme. The dimensions of the feature vector have logarithmic influence as well and the threshold is linear in the computational effort. All these parameters do not drastically influence the system of Rane [18]. Due to simple operations, such as summation and amount computation, computational overhead is negligible. However, concept of privacy-preserving authentication, discussed in this paper, has a strong influence on computational effort compared to plain-text biometric authentication systems.

In future work, our approach can be adapted for other domains. We propose to semantically shift data to complicate unauthorized decryption attempts, which makes user tracing via duplicate identification unlikely. Particularly, this becomes important, if the co-domain of the biometric feature is smaller than the co-domain of the key. The approach presented in [28] verifies users in the encrypted domain. It is imaginable that the extensions are of interest, too, for this approach, which bases on the homomorphic cryptosystem RSA.

Acknowledgment

We thank Martin Schäler for fruitful discussions on the first draft of this paper. The work in this paper has been funded in part by the German Federal Ministry of Education and Research (BMBF) through the Research Program "DigiDak+ Sicherheits-Forschungskolleg Digitale Formspuren" under Contract No. FKZ: 13N10816 and 13N10818.

References

[1] A. Grebhahn, M. Schäler, and V. Köppen, "Secure deletion: Towards tailor-made privacy in database systems," in BTW-Workshops. Köllen-Verlag, 2013, pp. 99–113.

[2] A. Grebhahn, M. Schäler, V. Köppen, and G. Saake, "Privacy-aware multidimensional indexing," in BTW. Köllen-Verlag, 2013, pp. 133–147.

[3] C. Vielhauer, Biometric User Authentication for IT Security, ser. Advances in Information Security. Springer, 2006, no. 18.

[4] M. Schäler, A. Grebhahn, R. Schröter, S. Schulze, V. Köppen, and G. Saake, "QuEval: Beyond high-dimensional indexing à la carte," PVLDB, vol. 6, no. 14, 2013, pp. 1654–1665.

[5] F. Emekci, O. Sahin, D. Agrawal, and A. E. Abbadi, "Privacy preserving decision tree learning over multiple parties," DKE, vol. 63, no. 2, 2007, pp. 348 – 361.

[6] A. Inan, Y. Saygyn, E. Savas, A. Hintoglu, and A. Levi, "Privacy preserving clustering on horizontally partitioned data," in Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on, 2006, pp. 95–95.

[7] D. Shah and S. Zhong, "Two methods for privacy preserving data mining with malicious participants," Information Sciences, vol. 177, no. 23, 2007, pp. 5468–5483.

[8] M. Leuckert, "Evaluation and extension of secure similarity verification in multi-computer scenarios to sesecure store and communicate biometric data," Master's thesis, Otto-von-Guericke University, 2013.

[9] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in EUROCRYPT, ser. Lecture Notes in Computer Science, J. Stern, Ed., vol. 1592. Springer, 1999, pp. 223–238.

[10] C. Vielhauer, J. Dittmann, and S. Katzenbeisser, "Design aspects of secure biometric systems and biometrics in the encrypted domain," in Security and Privacy in Biometrics, P. Campisi, Ed. Springer, 2013, pp. 25–43.

[11] S. Kiltz, A. Lang, and J. Dittmann, "Taxonomy for computer security incidents," in Cyber Warfare and Cyber Terrorism. IGI Global, 2008, pp. 412–417.

[12] S. Hildenbrand, D. Kossmann, T. Sanamrad, C. Binnig, F. Faerber, and J. Woehler, "Query processing on encrypted data in the cloud," Systems Group, Department of Computer Science, ETH Zurich, Tech. Rep., 2011.

[13] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in 6th ACM Conference on Computer and Communications Security. New York, NY, USA: ACM, 1999, pp. 28–36.

[14] A. Juels and M. Sudan, "A fuzzy vault scheme," Designs, Codes and Cryptography, vol. 38, no. 2, 2006, pp. 237–257.

[15] A. K. Jain, A. Ross, and U. Uludag, "Biometric template security: Challenges and solutions," in In Proceedings of European Signal Processing Conference, 2005.

[16] C. Gentry, "Computing arbitrary functions of encrypted data," Commun. ACM, vol. 53, no. 3, 2010, pp. 97–105.

[17] S. Rane, W. Sun, and A. Vetro, "Secure similarity verification between encrypted signals," US Patent US20 100 246 812 A1, Sep. 30, 2010.

[18] ——, "Secure similarity verification between homomorphically encrypted signals," US Patent US8 249 250 B2, Sep. 30, 2012.

[19] P. Stahlberg, G. Miklau, and B. N. Levine, "Threats to privacy in the forensic analysis of database systems," in Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '07. New York, NY, USA: ACM, 2007, pp. 91–102.

[20] A. Grebhahn, D. Broneske, M. Schäler, R. Schröter, V. Köppen, and G. Saake, "Challenges in finding an appropriate multi-dimensional index structure with respect to specific use cases," in Proceedings of the 24th GI-Workshop "Grundlagen von Datenbanken 2012", I. Schmitt, S. Saretz, and M. Zierenberg, Eds. CEUR-WS, 2012, pp. 77–82, urn:nbn:de:0074-850-4. [Online]. Available: http://ceur-ws.org/Vol-850/paper_grebhahn.pdf

[21] J. Daugman, "How iris recognition works," IEEE Trans. on Circuits and Systems for Video Technology, vol. 14, no. 1, 2004, pp. 21–30.

[22] A. Guttman, "R-trees: A dynamic index structure for spatial searching," SIGMOD Rec., vol. 14, no. 2, 1984, pp. 47–57.

[23] S. Berchtold, C. Böhm, and H.-P. Kriegel, "The Pyramid-technique: Towards breaking the curse of dimensionality," SIGMOD Rec., vol. 27, no. 2, 1998, pp. 142–153.

[24] V. Köppen, M. Schäler, and R. Schröter, "Toward variability management to tailor high dimensional index implementations," in RCIS. IEEE, 2014, pp. 452–457.

[25] R. Bayer and E. McCreight, "Organization and maintenance of large ordered indexes," Acta Informatica, vol. 1, 1972, pp. 173–189.

[26] D. Comer, "The Ubiquitous B-Tree," ACM Comput. Surv., vol. 11, no. 2, 1979, pp. 121–137.

[27] B. Schneier, Secrets & Lies: Digital Security in a Networked World. New York, NY, USA: John Wiley & Sons, Inc., 2000.

[28] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. V. Jawahar, "Efficient biometric verification in encrypted domain," in 3rd International Conference on Advances in Biometrics, 2009, pp. 899–908.

 ISBN: 978-1-61208-376-6

Data Quality and Security Evaluation Tool for Nanoscale Sensors

Leon Reznik
Department of Computer Science
Rochester Institute of Technology
Rochester, New York, USA
e-mail: lr@cs.rit.edu

Sergey Edward Lyshevski
Department of Electrical and Microelectronic Engineering
Rochester Institute of Technology
Rochester, New York, USA
e-mail: Sergey.Lyshevski@mail.rit.edu

Abstract— **The paper proposes a novel approach to data and information management in multi-stream data collection systems with heterogeneous data sources. Data may be produced by novel nanoscale photonic, optoelectronic and electronic devices. Poor quality characteristics are expected. In the proposed approach, we use a set of data quality indicators with each data entity, and, develop the calculus that integrates various data quality (DQ) indicators ranging from traditional data accuracy metrics to network security and business performance measures. The integral indicator will calculate the DQ characteristics at the point of data use instead of conventional point of origin. The DQ metrics composition and calculus are discussed. The tools are developed to automate the metrics selection and calculus procedures for the DQ integration is presented. The user-friendly interactive capabilities are illustrated.**

Keywords **- data quality; computer security evaluation; data accuracy; data fusion**.

I. INTRODUCTION

Recently we entered a new era of an exponential growth of data collected and made available for various applications. The existing technologies are not able to handle such big amounts of data. This phenomenon was called the big data. Photonics and nanotechnology enabled microsystems perform multiple generations and fusions of multiple data streams with various data quality [1-6]. The development and application of quantum-mechanical nanoscale electronic, photonic, photoelectronic communication, sensing and processing devices significantly increase an amount of data which can be measured and stored. These organic, inorganic and hybrid nanosensors operate on a few photons, electrons and photon-electron interactions [1, 2, 4, 6]. Very low current and voltage, high noise, large electromagnetic interference, perturbations, dynamic non-uniformity and other adverse features result in heterogeneous data with high uncertainty and poor quality. The super-large-density quantum and quantum-effect electronic, optoelectronic and photonic nanodevices and waveguides are characterized by: (i) Extremely high device switching frequency and data bandwidth (~1000 THz); (ii) Superior channel capacity (~10^{13} bits); (iii) Low switching energy (~10^{-17} J) [7, 8]. The importance of DQ analysis, data enhancements and optimization is emphasized due to: (1) Low signal-to-noise ratio (ratio of mean to standard deviation of measured signals is ~0.25 in the emerged electrons-photons interaction devices); (2) High probability of errors (p is ~0.001); (3) High distortion measure, reaching ~0.1 to 0.3; (4) Dynamic response and characteristic non-uniformity. These characteristics must be measured, processed and evaluated and provided to a data used along with the data.

New generations of information systems provide communication and networking capabilities to transfer, fuse, process and store data. Various applications require the data delivery from their origin to the point of use that might be far away. The data transfer may lead to information losses, attenuation, distortions, errors, malicious alterations, etc. Security, privacy and safety aspects of data communication and processing systems nowadays play a major role and may have a dramatic effect on the quality of data delivered.

New DQ management methods, quality evaluation and assurance (QE/QA) tools and robust algorithms are needed to ensure security, safety, robustness and effectiveness. As the amount of data available multiplies every year, current information systems are not capable to process these large data arrays to make the best decision. Big data applications require better data selection of high quality inputs. The absence of DQ indicators provided along with the data hinders the recognition of potential calamities and makes data fusion and mining procedures as well as decision making prone to errors.

In this paper we offer a novel approach to the data management in information systems. We propose to associate the DQ indicators with each data entity, and, replace one-dimensional data processing and delivery with multi-dimensional data processing and delivery along with the corresponding DQ indicators. To realize this approach, we describe the structure and content of these DQ indicators, develop the calculus of processing, and, develop interactive tools to automate this process. The current situation in DQ research is described in Section II. The DQ metrics composition is presented in Section III, while the DQ calculus is reported in Section IV. The CAD tools are documented in Section V. The conclusions are outlined in Section VI.

 ISBN: 978-1-61208-376-6

II. CURRENT ENVIRONMENT AND ACHIVEMENTS IN DQ EVALUATION

DQ represents an open multidisciplinary research problem, involving advancements in computer science, engineering and information technologies. The studied problems are directly applicable in various applications. It is essential to develop technologies and methods to manage, ensure and enhance quality of data. Related research in a networking field attempts to investigate how the network characteristics, standards and protocols can affect the quality of data collected and communicated through networks. In sensor networks, researchers started to investigate how to incorporate DQ characteristics into sensor-originated data [9]. Guha *et al.* proposed a single-pass algorithm for high-quality clustering of streaming data and provided the corresponding empirical evidence [10]. Bertino *et al.* investigated approaches to assure data trustworthiness in sensor networks based on the game theory [11] and provenance [12]. Chobsri *et al.* examined the transport capacity of a dense wireless sensor network and the compressibility of data [13]. Dong and Yinfeng attempted to optimize the quality of collected data in relation to resource consumption [14],[15]. Current developments are based on fusing multiple data sources with various quality and creating big data collections. Novel solutions and technologies, such as nano-engineering and technology are emerged in order to enable DQ assessment. Reznik and Lyshevski outlined integration of various DQ indicators representing different schemes ranging from measurement accuracy to security and safety [16], as well as micro- and nano engineering [17]. The aforementioned concepts are verified, demonstrated and evaluated in various engineering and science applications [18],[19].

III. DQ METRICS COMPOSITION

Data may have various quality aspects, which can be measured. These aspects are also known as data quality dimensions, or metrics. Traditional dimensions are as follows, some of them are described in [20],[21]:

- **Completeness:** Data are complete if they have no missing values. It describes the amount, at which every expected characteristic or trait is described and provided.
- **Timeliness**: Timeliness describes the attribute that data are available at the exact instance of its request. If a user requests for data and is required to wait a certain amount of time, it is known as a data lag. This delay affects the timeliness and is not desirable.
- **Validity**: It determines the degree, at which the data conforms to a desired standard or rules.
- **Consistency**: Data are consistent if they are free from any contradiction. If the data conforms to a standard or a rule, it should continue to do so if reproduced in a different setting.
- **Integrity**: Integrity measures how valid, complete and consistent the data are. Data's integrity is determined by a measure of the whole set of other data quality aspects / dimensions.

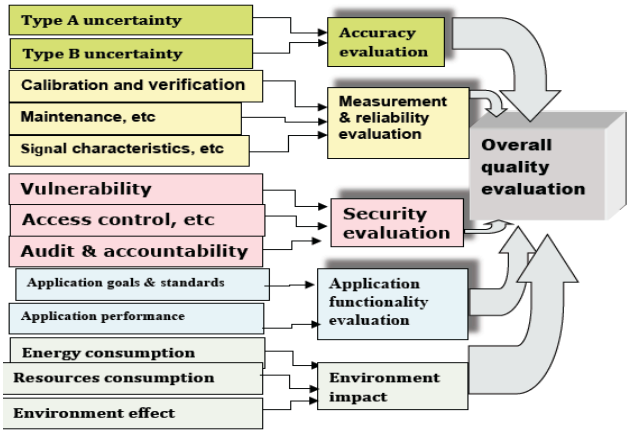


Figure 1. Integral quality evaluation composition

- **Accuracy**: Accuracy relates to the correctness of data and measurement uncertainty. Data with low uncertainty are correct.
- **Relevance**: It is a measure of the usefulness of the data to a particular application.
- **Reliability**: The quality of data becomes irrelevant if the data are not obtained from a reliable source. Reliability is a measure of the extent, to which one is willing to trust the data.
- **Accessibility**: It measures the timeliness of data.
- **Value added**: It is measured as the rate of usefulness of the data.

The methodologies of evaluating the DQ aspects listed above have been developed over the decades. They well represent the quality of the data at the point of their origin at the data source. However, nowadays most of the data are used far away from the point of their origin. In fact, the structured data are typically collected by distributed sensor networks and systems, then transmitted over the computer and communication networks, processed and stored by information systems, and, then, used. All those communication, processing and storage tasks affect the quality of data at the point of use, changing their DQ in comparison to one at the point of origin. The DQ evaluation should integrate accuracy and reliability of the data source with the security of the computer and communication systems. The high quality of the data at the point of their origin does not guarantee even an acceptable DQ at the point of use if the communication network security is low and the malicious alternation or loss of data has a high probability.

We describe the DQ evaluation structure as a multilevel hierarchical system. In this approach, we combine diverse evaluation systems, even if they vary in their design and implementation. The hierarchical system should be able to produce a partial evaluation of different aspects that will be helpful in flagging the areas that need urgent improvement. In our initial design we will classify metrics into five groups (see Figure 1):

 ISBN: 978-1-61208-376-6

TABLE I. SAMPLES OF GENERIC METRICS

Generic Attribute Name	DQ indicator/group (Figure1)	Description
Time-since-Manufacturing	Maintenance/reliability	The measure of the age of the device
Time-since-Service	Maintenance/reliability	The measure of the days since last service was performed in accord with the servicing schedule
Time-since-Calibration	Calibration/reliability	The measure of the days since last calibration was performed in accord with the calibration schedule
Temperature Range	Application/performance	The measure of temperature range within which the device will provide optimum performance
Physical Tampering Incidences	Physical security/security	The number of reported incidents that allowed unauthorized physical contact with the device
System Breaches	Access control/security	The measure of the number of unauthorized accesses into the system, denial of service attacks, improper usage, suspicious investigations, incidences of malicious code.
System Security	Security/security	Measures presence of intrusion detection systems, firewalls, anti-viruses.
Data Integrity	Vulnerabilities/securities	Number of operating system vulnerabilities that were detected.
Environmental Influences	Environment/environment	Number of incidences reported that would subject the device to mechanical, acoustical and triboelectric effects.
Atmospheric Influences	Environment/environment	Number of incidences reported that would subject the device to magnetic, capacitive and radio frequencies.
Response Time	Signals/reliability	Time between the change of the state and time taken to record the change

TABLE II. SAMPLES OF SPECIFIC DQ METRICS (EXAMPLES OF ELECTRIC POWER AND WATER METERS)

Device Name	Application specific Quality indicator	Description
Electric / Power Meters	Foucalt Disk	Check to verify the material of the foucalt disk.
	Friction Compensation	Difference in the measure of initial friction at the time of application of the compensation and the current friction in the device.
	Exposure to Vibrations	Measure of the number of incidences reported which would have caused the device to be subjected to external vibrations
Water Meters	Mounting Position	The measure of the number of days since regulatory check was performed to observe the mounting position of the device.
	Environmental Factors	Number of incidences reported which may have affected the mounting position of the device.
	Particle Collection	Measure of the amount of particle deposition.

(1) Accuracy evaluation;
(2) measurement and reliability evaluation;
(3) security evaluation;
(4) application functionality evaluation;
(5) environmental impact.

While the first three groups include rather generic metrics, groups #4 and #5 are devoted to metrics, which are specific to a particular application. Our metrics evaluation is based on existing approaches and standards, such as [22] for measurement accuracy and [23] for system security. Table I gives a sample of generic metrics representing all first three groups, while Table II lists the metrics, which are considered specific to a particular sensor and an application.

IV. DQ METRICS CALCULUS

In DQ calculus implementation we plan to investigate a wide number of options of calculating integral indicators from separate metrics ranging from simple weighted sums to sophisticated logical functions and systems. Those metrics and their calculation procedures will compose the DQ calculus. To simplify the calculus, we organize it as a hierarchical system calculating first the group indicators and then combining them into the system total. We follow the user-centric approach by offering an application user a choice of various options and their adjustment. We plan to introduce a function choice automatic adjustment,

verification and optimization.

To realize a wide variety of logical functions, the expert system technology is employed as the main implementation technique. The automated tool set includes the hierarchical rule-based systems deriving values for separate metrics, then combining them into groups and finally producing an overall evaluation. This way, the tool operation follows up the metrics structure and composition (see figure 2). This system needs to be complemented by the tools and databases assisting automation of all stages in the data collection, communication, processing and storage for all information available for data quality evaluation. The developed tools facilitate automated collection, communication and processing of the relevant data. Based on the data collected, they not only evaluate the overall data quality but also determine whether or not the data collection practice in place is acceptable and cite areas that are in need of improvement.

In our automated procedures, the DQ score is computed by applying either linear, exponential or stepwise linear reduction series to the maximum score of an attribute. In case an attribute defines a range for ideal working, the linear series is substituted by a trapezoidal drop linear series and exponential is replaced by a bell drop series.

When considering both accuracy and security DQ metrics, assessing whether fusion enhances DQ is not obvious as one has to tradeoff between accuracy, security and other goals. While adding up a more secure data transmission channel improves both security and accuracy indicators, using a more accurate data stream will definitely improve data accuracy but could be detrimental to certain security indicators (see [24] for further discussion). If resources are limited, as in the case of sensor networks, one might consider trying to improve accuracy of the most secure data source versus more or less even distribution of security resources in order to achieve the same security levels on all data channels. The concrete recommendations will depend on the application.

V. GENERIC TOOL DESIGN

The proposed design of the tool divides the procedure for automated data collection in three main stages. First stage involves mainly a device configuration. Since the tool is generic, it provides certain flexibility in configuring a large variety of diverse devices. These devices could be electric meters, power meters, water meters and marine sensors. The second stage computes data quality indicators of the configured device. The final stage performs the detailed analysis of the computed data quality indicators. It highlights low data quality and help flag erroneous data. Also, it provides recommendations on improving low data quality and help ensure that the data being utilized are fit for the purpose it is intended to be used. Figure 2 presents the architecture of the tool. Currently, the first and the second stages are implemented.

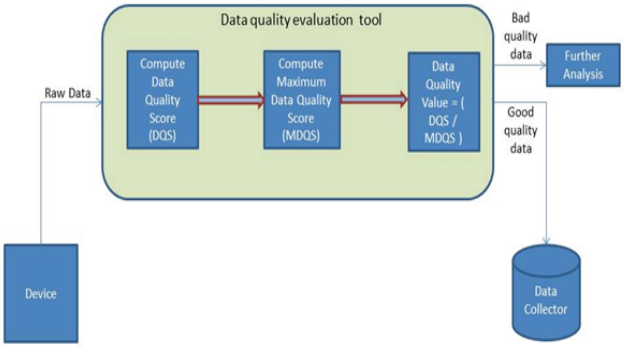


Figure 2. Data quality evaluation procedure

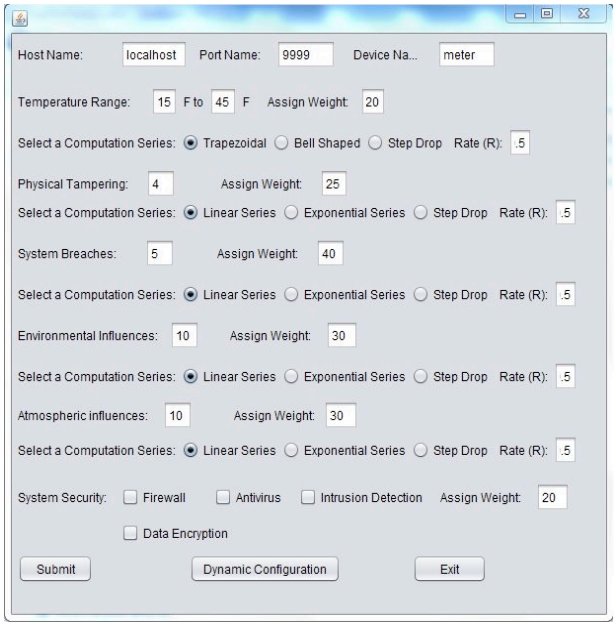


Figure 3. Generic attribute configuration

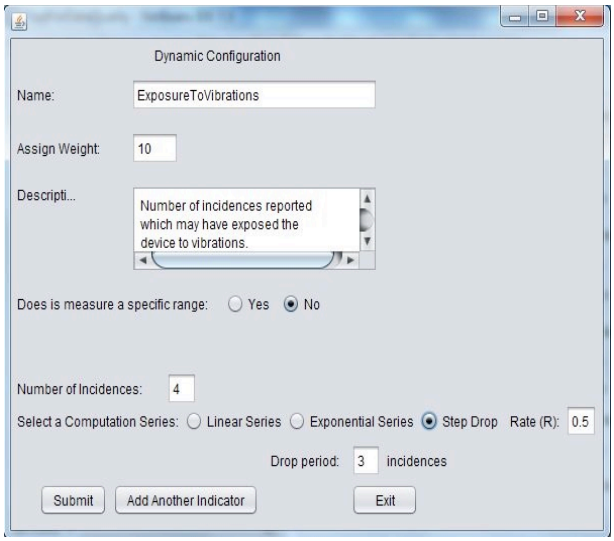


Figure 4. Application specific attribute configuration

The generic tool allows for a configuration of a large variety of devices. Each automated data collection device has DQ factors, which are common to other similar devices. These factors are referred by the tool as generic attributes. Other attributes, which are unique to a particular device are called dynamic attributes. These attributes are assigned the maximum score based on the significance of the contribution they would add to the data quality. The greater the significance, the greater is the score.

The configuration step mainly involves recognizing the generic and application-specific attributes, as well as assigning the max possible score to each of them. Generic attributes are common to most devices, for example, timeliness and quality of common device servicing such as calibration. Application-specific attributes are unique to a device, for example, exposure to vibration, shock and radiation. This is important for a particular application because certain devices, like electric meters, produce misleading results when exposed to the external adversary affects. If, for some reason, a generic attribute does not apply to a particular device, the max score of zero would be applied in order to eliminate the attribute from the analysis. Table I describes the generic attributes being considered by the tool. Figure 3 illustrates configuring some of the generic attributes for an electric meter. Table II describes some application specific attributes, which are device and application specific. Figure 4 illustrates configuring an application-specific attribute for an electric meter, provided as an example.

The second stage involves data quality computation. The configured generic and application specific attributes help compute the individual quality scores. Each attribute is considered a quality indicator, whose significance will be dependent on its max score. These quality indicators produce a quality score using a chosen logic procedure. For example, we can consider a generic attribute called time-since-calibration. Some devices need to get calibrated every year. If a device has not been calibrated for an entire year or a couple of years, the quality factor for that indicator will go down. If the device has never been calibrated since its installation it can affect the quality score even more. The tool allows a user to define the procedure for calculating the application-specific quality indicators.

VI. CONCLUSIONS

The paper introduces a novel approach to data management in data collection and processing systems, which might incorporate SCADA, sensor networks and other systems with nanoscale devices. We associate each data entity with the corresponding DQ indicator. This indicator integrate various data characteristics ranging from accuracy to security, privacy and safety, etc. It considers various samples of DQ metrics representing communication and computing security as well as data accuracy and other characteristics. Incorporating security and privacy measures into the DQ calculus is especially important in the current development as it allows shifting the DQ assessment from the point of data origin to the point of data use.

A unified framework for assessing DQ is critical for enhancing data usage in a wide spectrum of applications because this creates new opportunities for optimizing data structures, data processing and fusion based on the new DQ information use. By providing to an end user or an application the DQ indicators which characterize system and network security, data trustworthiness and confidence, etc. Correspondingly, an end user is in a much better position to decide whether and how to use data in various applications. A user will get an opportunity to understand and compare various data files, streams and sources based on the associated DQ with integral quality characteristics reflecting various aspects of system functionality and to redesign data flows schemes. This development will transform one-dimensional data processing into multi-dimensional data optimization procedures for application-specific data applications. We describe and demonstrate an application of the DQ metrics definition and calculation tools, which enable integration of various metrics to calculate an integral indicator.

REFERENCES

[1] P. W. Coteus, J. U. Knickerbocker, C. H. Lam and Y. A. Vlasov, “Technologies for exascale systems,” IBM Journal of Research and Developments, vol. 55, issue 5, pp. 14.1-14.12, 2011.

[2] *Handbook on Nano and Molecular Electronics*, Ed. S. E. Lyshevski, CRC Press, Boca Raton, FL, 2007.

[3] B. G. Lee et. al., “Monolithic silicon integration of scaled photonic switch fabrics, CMOS logic, and device driver circuits,” Journal of Lightwave Technology, vol. 32, issue 4, pp. 743-751, 2014.

[4] S. E. Lyshevski, Molecular Electronics, Circuits and Processing Platforms, CRC Press, Boca Raton, FL, 2007.

[5] Micro-Electromechanical Systems (MEMS), International Technology Roadmap for Semiconductors, 2011 and 2013 Editions, available at www.itrs.net , accessed on August 1, 2014.

[6] A. Yariv, *Quantum Electronics*, John Wiley and Sons, New York, 1988.

[7] Emerging Research Devices, International Technology Roadmap for Semiconductors, 2011 and 2013 Editions, available at www.itrs.net , accessed on August 1, 2014.

[8] J. Warnock, “Circuit and PD challenges at the 14nm technology node,” Proc. 2013 ACM Int. Symposium on Physical Design, pp. 66-67, 2013.

[9] M. Klein and W. Lehner, "Representing Data Quality in Sensor Data Streaming Environments," J. Data and Information Quality, vol. 1, pp. 1-28, 2009.

[10] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering Data Streams: Theory and Practice," IEEE Trans. on Knowl. and Data Eng., vol. 15, pp. 515-528, 2003.

 ISBN: 978-1-61208-376-6

[11] H. S. Lim, K. M. Ghinita, and E. Bertino "A Game-Theoretic Approach for High-Assurance of Data Trustworthiness in Sensor Networks," presented at the IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA, 2012.

[12] C. Dai, H.S. Lim, and E. Bertino "Provenance-based Trustworthiness Assessment in Sensor Networks," , 7th Workshop on Data Management for Sensor Networks (DMSN), in conjunction with VLDB, DMSN 2010, Singapore, 2010.

[13] S. Chobsri, W. Sumalai, and W. Usaha, "Quality assurance for data acquisition in error prone WSNs," in Ubiquitous and Future Networks, 2009. ICUFN 2009. First International Conference on, 2009, pp. 28-33.

[14] W. Dong, H. Ahmadi, T. Abdelzaher, H. Chenji, R. Stoleru, and C. C. Aggarwal, "Optimizing quality-of-information in cost-sensitive sensor data fusion," in 2011 International Conference on Distributed Computing in Sensor Systems (DCOSS 2011), 27-29 June 2011, Piscataway, NJ, USA, 2011, 8 pp.

[15] W. Yinfeng, W. Cho-Li, C. Jian-Nong, and A. Chan, "Optimizing Data Acquisition by Sensor-channel Co-allocation in Wireless Sensor Networks," in 2010 International Conference on High Performance Computing (HiPC 2010), 19-22 Dec. 2010, Piscataway, NJ, USA, 2010, p. 10 pp.

[16] L. Reznik, "Integral Instrumentation Data Quaity Envaluation: the Way to Enhance Safety, Security, and Environment Impact," 2012 IEEE International Instrumentation and Measurement Technology Conference, Graz, Austria, May 13-16, 2012, 2012.

[17] S. E. Lyshevski and L. Reznik, "Processing of extremely-large-data and high-performance computing," in International Conference on High Performance Computing, Kyiv, Ukraine, 2012, pp. 41-44.

[18] G. P. Timms, P. A. J. de Souza, L. Reznik, and D. V. Smith, "Automated Data Quality Assessment of Marine Sensors," *Sensors,* vol. 11, pp. 9589-9602, 2011.

[19] G. P. Timms, P. A. de Souza, and L. Reznik, "Automated assessment of data quality in marine sensor networks," in OCEANS 2010 IEEE - Sydney, 2010, pp. 1-5.

[20] F. G. Alizamini, M. M. Pedram, M. Alishahi, and K. Badie, "Data quality improvement using fuzzy association rules," in Electronics and Information Engineering (ICEIE), 2010 International Conference On, 2010, pp. V1-468-V1-472.

[21] L. Sebastian-Coleman, Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework: Morgan-Kaufmann Publishers, 2013.

[22] ANSI/NCSL, "US Guide to the Expression of Uncertainty in Measurement," ed, Z540-2-1997.

[23] National Institute of Standards and Technology, "Performance Measurement Guide for Information Security," ed. Geithersburg, MD, July 2008.

[24] L. Reznik and E. Bertino, "Poster: Data quality evaluation: integrating security and accuracy," Proceedings of the 2013 ACM SIGSAC conference on Computer communications security, Berlin, Germany, 2013.

 ISBN: 978-1-61208-376-6

AndroSAT: Security Analysis Tool for Android Applications

Saurabh Oberoi*, Weilong Song†, Amr M. Youssef‡
Concordia Institute for Information Systems Engineering
Concordia University
Montreal, Quebec

***Abstract*—With about 1.5 million Android device activations per day and billions of application installation from Google Play, Android is becoming one of the most widely used operating systems for smartphones and tablets. In this paper, we present *AndroSAT*, a Security Analysis Tool for Android applications. The developed framework allows us to efficiently experiment with different security aspects of Android Apps through the integration of (i) a static analysis module that scans Android Apps for malicious patterns. The static analysis process involves several steps such as n-gram analysis of dex files, de-compilation of the App, pattern search, and analysis of the AndroidManifest file; (ii) a dynamic analysis sandbox that executes Android Apps in a controlled virtual environment, which logs low-level interactions with the operating system. The effectiveness of the developed framework is confirmed by testing it on popular Apps collected from F-Droid, and malware samples obtained from a third party and the Android Malware Genome Project dataset. As a case study, we show how the analysis reports obtained from *AndroSAT* can be used for studying the frequency of use of different Android permissions and dynamic operations, detection of Android malware, and for generating cyber intelligence about domain names involved in mobile malware activities.**

Keywords–Android Security; Static Analysis; Dynamic Analysis.

I. INTRODUCTION

According to a recent report from Juniper Networks [1], smartphone sales have increased by 50% year-on-year. In the third quarter of 2013, more than 250 million smartphones were sold worldwide. This rapid increase of smartphone usage has moved the focus of many attackers and malware writers from desktop computers to smartphones. Today, mobile malware is far more widespread, and far more dangerous, especially in Bring Your Own Device (BYOD) arrangements where mobile devices, which are often owned by users who act as defacto administrators, are being used for critical business and are also being integrated into enterprises, government organizations and military networks [2][3].

Android, being one of the utmost market share holders, not only for smartphones and tablets, but also in other fields such as automotive integration, wearables, smart TVs and video gaming systems, is likely to be facing the highest threat from malware writers. As an open-source platform, Android is arguably more vulnerable to malicious attacks than many other platforms. According to the report from Juniper Networks [1], mobile malware grew 614% for a total of 276,250 malicious Apps from March 2012 to March 2013. Another recent report from Kaspersky [4] shows that 99% of all mobile-malware in the wild is attacking the Android platform. Kaspersky also mentioned that mobile malware is no longer an act of an individual hacker; some rogue companies are investing time and money to perform malicious acts such as stealing credit card details and launching phishing attacks, to gain profit. According to the Kaspersky report, the number of unique banking trojans raised from 67 to 1321 from the start to the end of 2013. Thousands of users were convinced to pay millions of dollars due to the gradual dissemination of infected Apps. In extreme cases, an application with malicious intent can do more than just sending premium text messages–they can turn a phone into a spying tool. These spying tools can track the current location of a smartphone, make phone calls, send and receive text messages and send stolen private information to remote servers without raising any alarm.

In this paper, we present a Security Analysis Tool for Android applications, named *AndroSAT*. The developed framework allows us to experiment with different security aspects of Android Apps. In particular, *AndroSAT* comprises of:

- A static analysis module that scans Android Apps for malicious patterns (e.g., potentially malicious API calls and URLs). This process involves several steps such as n-gram analysis of dex files, de-compilation of the App, pattern search, and extracting security relevant information from the AndroidManifest files.
- A dynamic analysis sandbox that executes Android Apps in a controlled virtual environment, which logs low-level interactions with the operating system.
- Analysis tools and Add-ons for investigating the output of the static and dynamic analysis modules.

In order to demonstrate the effectiveness of our framework, we tested it on popular Apps collected from F-Droid [5], which is a Free and Open Source Software (FOSS) repository for Android applications, and a malware dataset obtained from a third party as well as from the Android Malware Genome Project. The reports produced by our analysis were used to perform three case studies that aim to investigate the frequency of use of different Android permissions and dynamic operations, detection of malicious Apps and generating cyber intelligence about domain names involved in mobile malware activities. The results obtained by the first case study can be utilized to narrow down the list of features that can be used to determine malicious patterns. In the classification experiment, using the features extracted from our analysis reports, we applied feature space reduction, and then performed classification on the resultant dataset. As will be explained in Section V, the obtained classification results are very promising. Finally, in our cyber-intelligence gathering experiment, we used the IP addresses recorded during the static and dynamic analysis of malware Apps to produce a graphical representation of the geographical locations of possible malicious servers (and their ISPs) that communicate with malicious Apps. These three experiments show the versatility as well as the wide variety of possible usages for the information obtained by *AndroSAT*.

The rest of the paper is organized as follows. In the next section, we discuss some related work. A brief review of

 ISBN: 978-1-61208-376-6

Android and its security model is provided in Section III. Section IV details the design of our framework and explains the static and dynamic analysis modules. Our experimental results are presented in Section V. Finally, our conclusion is given in Section VI.

II. Related Work

Due to sudden increase in the number of Android malware, researchers too have moved their focus and resources towards securing the Android platform from this rising threat. Blasing *et al.* [6] developed a system named AASandbox that utilizes a loadable kernel module to monitor system and library calls for the purpose of analyzing Android applications. Wu *et al.* [7] developed a system named DroidMat that extracts the information from an AndroidManifest and, based on the collected information, it drills down to trace the application programming interface(API) calls related to the used permissions. It then uses different clustering techniques to identify the intentions of the Android malware. Reina *et al.* [8] developed a system named CopperDroid, which performs the dynamic analysis of an Android application based upon the invoked system calls. They claimed that their system can detect the behavior of an application whether it was initiated through Java, Java Native Interface or native code execution. Burguera *et al.* [9] focused on identifying system calls made by Android applications and developed a tool named Crowdroid to extract the system calls and then categorize these system calls into either malicious or benign by using K-means clustering. DroidRanger, a system proposed in [10], consists of a permission-based behavioral footprinting scheme that detects new samples of known Android malware families and a heuristics-based filtering scheme that identifies certain inherent behaviors of unknown malicious families.

Spreitzenbarth *et al.* [11] developed a system named Mobile-Sandbox, which is designed to perform integrated analysis and some specific techniques to log calls to non-Java APIs. Alazab *et al.* [12] used DroidBox, a dynamic analysis tool that generates logs, behavior graph and treemap graphs to explain the behavior of an Android App. They collected 33 malicious applications grouped into different families and scanned them with different antivirus. They combined the graphs of the applications within the same family to verify if the graphs eventually reflect the family and then compared it with results from different antivirus companies. Another dynamic analysis tool named TaintDroid is presented in [13], which is capable of simultaneously tracking multiple sources of sensitive data accessed by Android application. In the work reported in [14][15], n-gram features are extracted from benign and malware executables in Windows PE format. The extracted features are then used to generate model with classifiers supported by WEKA.

Compared to other related work, one key feature in our system, *AndroSAT*, is that the developed sandbox allows not only for observing and recording of relevant activities performed by the apps (e.g., data sent or received over the network, data read from or written to files, and sent text messages) but also manipulating, as well as instrumenting the Android emulator. These modifications were made to the Android emulator in order to evade simple detection techniques used by malware writers.

III. Android Overview

Android is an emerging platform with about 19 different versions till date [16]. Table I shows different Android versions with their corresponding release date. As shown in Figure 1, the Android framework is built over Linux kernel [17] that controls and governs all the hardware drivers such as audio, camera and display drivers. It contains open-source libraries such as SQLite, which is used for database purposes, and SSL library that is essential to use the Secure Sockets Layer protocol. The Android architecture contains *Dalvik Virtual Machine* (DVM), which works similar to the Java Virtual Machine (JVM). However, DVM executes *.dex* files whereas JVM executes *.class* files.

TABLE I. ANDROID VERSION HISTORY

Android Version	OS Name	Release Date
1.0	Alpha	09/2008
1.1	Beta	02/2009
1.5	Cupcake	04/2009
1.6	Donut	09/2009
2.0-2.1	Eclair	10/2009
2.2	Froyo	05/2010
2.3.x	Gingerbread	12/2011
3.1-3.2	Honeycomb	02/2011
4.0.3-4.0.4	Ice Cream Sandwich	10/2011
4.1.x-4.3	Jelly Bean	08/2012
4.4	KitKat	09/2013

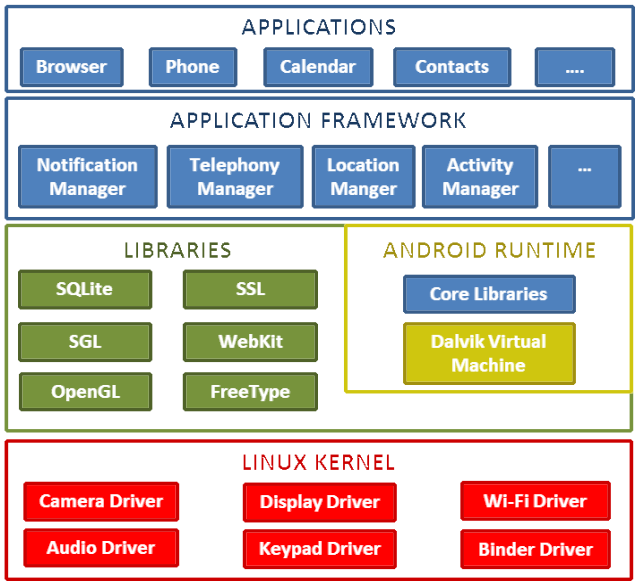


Figure 1. Android architecture [17]

Every application runs in its own Dalvik virtual environment or sandbox in order to avoid possible interference between applications and every virtual environment running an application is assigned a unique *User-ID* (UID). The application layer consists of the software applications with which users interact. This layer communicates with the application framework to perform different activities. This application framework consists of different managers, which are used by an Android application. For example, if an application needs access to an

 ISBN: 978-1-61208-376-6

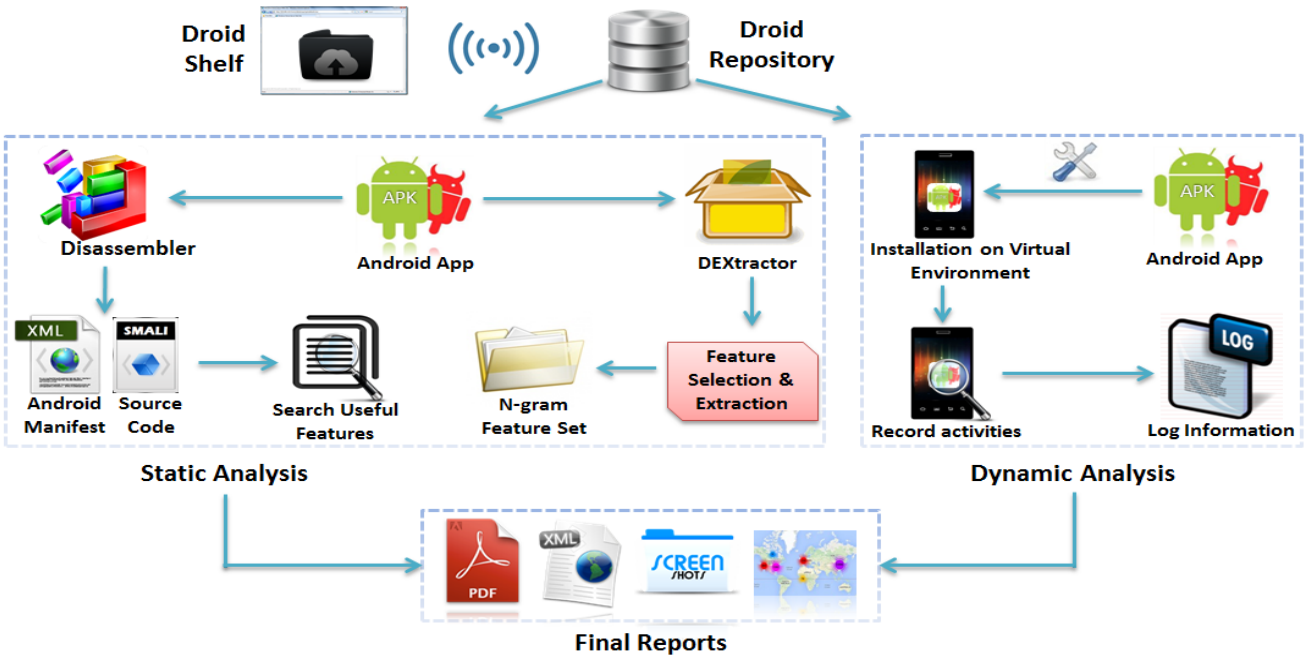


Figure 2. Overview of *AndroSAT*

incoming/outgoing phone call, it needs to access *Telephony-Manager*. Similarly, if an application needs to pop-up some notifications, it should interact with *NotificationManager*.

An Android application, also known as an APK package, consists of AndroidManifest.xml, res, META-INF, assets and classes.dex files. The AndroidManifest.xml file contains information about supported versions, required-permissions, services-used, receivers-used, and features-used [17]. META-INF contains the certificate of the application developer, resource directory (res) contains the graphics used by an applications such as background, icon and layout [17]. Assets directory contains the files used by an Android application, such as SQLite database and images. The classes.dex file is an executable file in a format that is optimized for resource constrained systems.

IV. System Overview

In this section, we provide an overview of *AndroSAT*. A local web-server is setup where we can upload the Android applications into our Droid-Repository (MySQL database of Android applications to be analyzed) through a PHP webpage (DroidShelf). As depicted in Figure 2, *AndroSAT* includes two main modules, namely a static analysis module and a dynamic analysis module, which are used together to produce analysis reports in both XML and pdf formats. The produced XML reports can then processed using several add-ons and analysis tools.

A. Static Analysis

Static analysis techniques aim to analyze Android Apps without executing them. The objective of these techniques is to understand an application and predict what kind of operations and functionalities might be performed by it without executing it. Different forms of static analysis have proved to be very

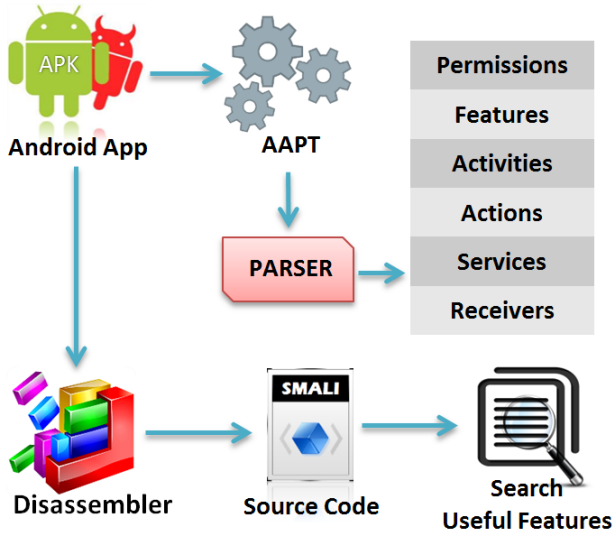


Figure 3. Overview of the static analysis module

useful in detecting malicious Apps. As shown in Figure 2 and Figure 3, the process of static analysis involves several steps such as extracting n-gram statistics of .dex files, disassembling the application, performing pattern search for malicious API calls and URLs, and extracting relevant information (such as used permissions, activities, intents and actions, services, and receivers) from the AndroidManifest file. In order to perform the static analysis process, the analyzed application is first fetched from the Droid-Repository. Then, data from *AndroidManifest* file is extracted from the APK package using the Android Asset Packaging Tool (AAPT). AAPT is used for compiling the resource files during the process of Android App

development, and is included in the Android SDK package [18]. After the n-gram statistics is evaluated from the .dex file, the application is fed to the disassembler, which disassembles the application APK package to obtain the SMALI and Java code of the application. The disassembly process is performed using APKTool [19] and Apk2java [20], which are open source reverse engineering tools. Once the source code is obtained from the Android application undergoing analysis, we search for malicious functions/API calls, URLs and IP addresses. In what follows we provide some further details on the n-gram analysis process and the different features extracted from both AndroidManifest and source code.

1) N-gram Analysis: Different forms of n-gram analysis have been previously used for malware detection in the Windows and Linux/Unix environments. Different from Portable Executables (PE) but similar to MSI packages in Windows, Android OS has Android application package file (APK) as the file format used to install application software. The APK file can be looked at as a zip compression package containing all of the application bytecode including *classes.dex* file, compiled code libraries, application resource (such as images, and configuration files), and an XML file, called AndroidManifest. The *classes.dex* file holds all of application bytecode and implementing any modification in the application behavior will lead to a change in this file. The process of n-gram analysis is performed by extracting application bytecode files (i.e., *classes.dex*), calculating byte n-gram, and then performing a dimensionality reduction step for these calculated n-gram features. The byte n-grams are generated from the overlapping substrings collected using a sliding window where a window of fixed size slides one byte every time. The n-gram feature extraction captures the frequency of substrings of length n byte. Since the total number of extracted features is very large, we apply feature reduction to select appropriate features for malware detection. We chose Classwise Document Frequency [14] as our feature selection criterion. *AndroSAT* applies feature reduction on bigram and trigram sorted by CDF value and top k features are selected. The obtained feature vectors are saved in the analysis reports and can then be used as inputs for classifier to generate models for malicious Apps. Surprisingly, as will be shown in the experimental results Section V, applying this simple analysis method to .dex files even without any pre-processing or normalization for the byte code yields very promising results and allows us to differentiate between malicious and benign applications with relatively very good accuracy.

2) Features extracted from the AndroidManifest file: Throughout the analysis process, the following features are extracted from the AndroidManifest file of analyzed applications:

- Requested Permissions: An Android application does not need any permission unless it is trying to use a private or system related resource/functionality of the underlying Android device. There are numerous permissions that developers can add into an Android application to provide better experience to users. Example of these permissions include CAMERA, VIBRATE, WRITE_EXTERNAL_STORAGE, RECEIVE_SMS, and SEND_SMS [5]. Permissions requested by an application inform user about what they can expect from the application and a smart user can easily realize if an application is asking for more than it should supposedly do. For example, an application claiming to show weather reports should raise suspicion if it requests a SEND_SMS permission.
- Features Used: An Android application can use hardware or software features. The features available (e.g., bluetooth, and camera) vary with different Android devices. Therefore, many applications use feature as a preference, i.e., they can still function even if the feature is not granted. Features come with a required attribute that helps a developer to specify whether the application can work normally with or without using the specific feature.
- Services-Used: Services-Used lists all the services recorded in the application AndroidManifest file. In an Android application, a service can be used to perform operations that need to run at the background for a long time and without any user interaction. The service keeps on running even if the user switches to another application. An attacker can make use of a service to perform malevolent activities without raising an alarm.
- Receivers Used: In Android, events send out a broadcast to all the applications to notify their occurrence. A broadcast is triggered once the event registered with the corresponding broadcast receiver [21] occurs. The main purpose of using a broadcast receiver is to receive a notification once an event occur. For example, if there is a new incoming message, a broadcast about the new incoming message is sent out and applications that use the corresponding receiver, i.e., *SMS_RECEIVED* receiver will get the incoming message. Malicious applications can use the broadcast receiver in numerous ways such as receive the incoming messages and monitor the phone state.
- Intents and Actions: An intent or action specifies the exact action performed by the broadcast receiver used in an application. Some of the most widely used broadcast receivers include SMS_RECEIVED, and BOOT_COMPLETED.
- Activities Used: Activities-used is a list of all the activities used in an Android application. In Android, every screen that is a part of an application and with which users can interact is known as an activity. An application can have more than one activity.

3) Feature Extraction from Source Code: In this section, we list the features extracted from the decompiled SMALI and Java source code.

- getLastKnownLocation: This function is used to get the last know location from a particular location provider. Getting this information does not require starting the location provider, which makes it more dangerous and invisible. Even if this information is stale, it is always useful in some contexts for malicious App developers.
- sendTextMessage: This function is most widely used by many malware developers to earn money or to send bulk messages while hiding their identity. The biggest advantage that attackers have when utilizing this function is that it sends text messages in the background and does not require any user intervention.

 ISBN: 978-1-61208-376-6

- getDeviceId: This function is used to obtain the *International Mobile Station Equipment Identity* (IMEI) number of the Android device. Every device has its own unique IMEI that can be used by the service provider to allow the device to access the network or block its access to the network. IMEIs of stolen phones are blacklisted so that they never get access to the network. An attacker with malevolent intentions can use this unique code to make a clone and then performs illegal activities or blacklists the IMEI so that the user can never put it back onto the cellular network.

All the features extracted by the static analysis module are then fed to a parser module in order to remove redundant data. The extracted relevant information is then saved in both XML and PDF formats.

B. Dynamic Analysis

The main advantage of the static analysis described above is that it can be performed relatively very efficiently without the need to execute the Apps and hence avoid any risk associated with executing malicious Apps. On the other hand, some malware writers use different obfuscation and cryptographic techniques that make it almost impossible for static analysis techniques to obtain useful information, which makes it essential to use dynamic analysis. Dynamic analysis is most widely used to analyze the behavior and interactions of an application with the operating system. Typically, dynamic analysis is performed using a virtual machine controlled environment in order to avoid any possible harm that can result from running the malware on actual mobile devices. Furthermore, using the virtual environment makes it easier to prepare a fresh image and install the new application in question on it for analysis.

The main disadvantage of dynamic analysis, however, is that the usefulness of the analysis is somewhat correlated to the length of the analysis interval and some malicious activities may not be invoked by the App during the, usually short, analysis interval either because the conditions to trigger these events do not happen during the dynamic analysis process or because the malicious App is able to detect that it is being monitored. Anti-debugging and virtual machine detection techniques have long been used by Windows malware writers. To make the virtual environment look like a genuine smartphone, we made some changes to the Android emulator, e.g., we modified IMEI, IMSI, SIM serial number, product, brand and other information related to the phone hardware.

During dynamic analysis, the application is installed onto the system and its activities or interactions are logged to analyze its actions. We use a sandbox to execute the Android application in question in a controlled virtual environment. Figure 4 shows an overview of our dynamic analysis module. The main part of this module is based on an open source dynamic analysis tool for Android applications named DroidBox [22]. However, as mentioned above, we performed some modifications in order to improve the resistance of the emulator against detection. *AndroSAT* launches the emulator using DroidBox that uses its own modified image making it possible to log the system and API level interactions of an Android application with the emulator. Once the emulator is up and running, the App is installed using *Android Debug Bridge* (ADB) for further analysis. Immediately after the successful

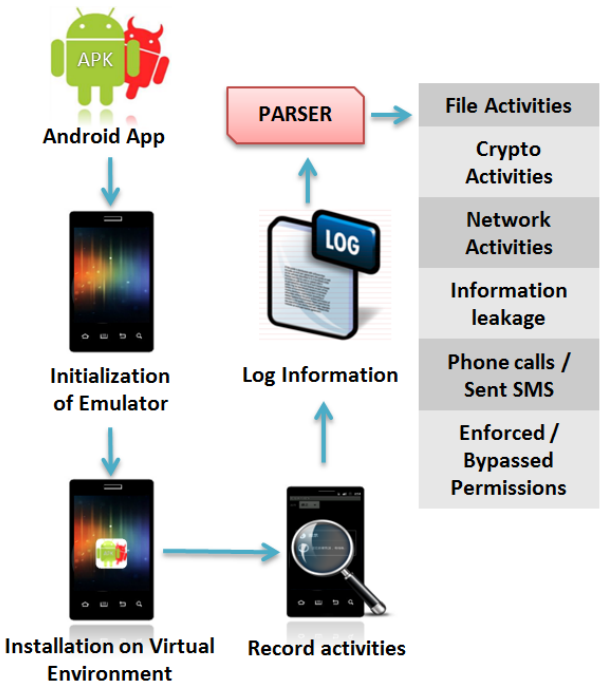


Figure 4. Overview of the dynamic analysis module

installation of the application, the module starts DroidBox to further analyze the APP for a configurable interval (the default is two minutes). Meanwhile, it launches the main activity of the installed application onto the emulator automatically, performs random gestures on it and takes screen shots of the application using the MonkeyRunner tool [23], while DroidBox consistently logs any system or API level interactions of the application with the operating system. The following features are collected during our dynamic analysis:

- File Activities: File activities consist of information regarding any file, which is read and/or written by the application. This information includes timestamps for these file activities, absolute path of accessed files and the data, which was written to/read from these files.
- Crypto Activities: Crypto Activities consist of information regarding any cryptographic techniques used by the application. It includes information regarding the type of operation (e.g., key generation, encryption, and decryption) performed by the application, algorithms used (e.g., AES, DES), key used and the data involved.
- Network Activities: It unveils the connections opened by the application, packets sent and received. It also provides detailed information about all these activities including the timestamp, source, destination and ports.
- Dex Class Initialized: In Android, an application can initialize a dex file, which is not a part of its own package. In the most malicious way, an application can download a dex file to the Android device and then executes it using the *DexClassLoader*. This way, an application under analysis will come out clean, which makes it almost impossible for any malware analyzer or sandbox to detect the malicious activities

 ISBN: 978-1-61208-376-6

performed by the application. DroidBox logs relevant details whenever an application initializes any dex class.

- Broadcast Receiver: As explained earlier, the use of broadcast receiver helps improve the user experience of an application. However, an attacker can use this functionality to easily gain access to private/critical data without raising an alarm in the users' mind. We log information regarding any broadcast receiver used by the application and record the name of the broadcast and the corresponding action.
- Started Services: Services play a very critical role in Android applications. They are used to execute the code in the background without raising an alarm. Started services provide the information about any service, which is started or initialized during the runtime of the application.
- Bypassed permissions: Lists the permission names, which are bypassed by the application. This aims to detect scenarios where an application can perform the task that needs a specific permission without explicitly using that permission. For example, an Android application can direct the browser to open a webpage without even using the Internet permission [24].
- Information Leakage: Information leakage can occur through files, SMS and network. Leakage may occur through a file if the application tries to write or read any confidential information (e.g., IMEI, IMSI, and phone number) of an Android device to or from a file. Leakage occurs through SMS if the information is sent through an SMS. Timestamp, phone number to which the information is sent, information type, and data involved are also logged. Leakage occurs through network if the application sends critical data over the Internet. Timestamp, destination, port used, information type and data involved is recorded. Detailed information about the absolute path of the file, timestamp, operation (read or write), information type (IMEI, IMSI or phone number) and data are logged.
- Sent SMS: If an Android application tries to send a text message, timestamp, phone number and the contents of the text message are logged.
- Phone call: If an Android application tries to make a phone call, timestamp and phone number are recorded.

Dynamic analysis module logs all these features into a text file, which is then sent to the parser module to remove any redundant data. The extracted relevant information is then saved in XML and PDF formats.

V. EVALUATION

The reports generated by our framework contain useful information regarding the analyzed Android applications, which in turn can be used in many Android security related applications. To confirm the effectiveness of our proposed framework, we analyzed a total of 1932 Android applications, out of which 970 are benign and 962 are malicious. We collected the malicious samples from the Android Malware Genome Project [25] and from another third party. The benign samples were obtained from F-Droid [18], which is a Free and Open Source Software (FOSS) repository for Android applications. We also verified the applications collected from F-Droid are benign using VirusTotal [26].

Results from the dynamic analysis show that 254 out of 962 (i.e., 26%) malicious applications and none of the 970 benign applications lead to private data leakage through network. Many malware writers use cryptographic techniques to hide the malicious payload in order to make it impossible for a signature based malware analyzer to understand the malicious intentions of an application. Among the analyzed Apps 41 out of 962 malicious applications and 2 out of 970 benign applications use cryptography at runtime. The experimental results also suggest that an Android application with different versions might have different package contents and hence the checksum of the packages might differ. However, the checksum of *classes.dex* file of some different versions came out to be the same. This tells us that malware writers might add junk data in the APK package to make an application look different while the content of *classes.dex* file remains the same.

We incorporated the reports generated by our framework and used them to perform three case studies, namely performing frequency analysis for the different permissions and operations used by Android Apps, cyber-intelligence and classification.

A. Frequency analysis of Android permissions and dynamic operations

Figure 5(a) shows the top 15 permissions used by the analyzed malicious applications and their frequency as compared to the benign ones. It is interesting to find out that some permissions are used by most of the malicious applications. As depicted in Figure 5(a) READ_PHONE_STATE permission is used by $\approx$86% of the malicious Apps as compared to $\approx$12.% of the benign Apps. This permission is most widely used to obtain system specific data such as IMEI, IMSI, phone number and SIM serial. Similarly, the frequency of use of INTERNET, ACCESS_WIFI_STATE, ACCESS_NETWORK_STATE, READ_SMS, and WRITE_SMS show noticeable differences. Figure 5(b) shows the top 15 permissions used by the analyzed benign applications and their frequency as compared to the analyzed malicious applications. In total, 962 malicious applications used 10,203 permissions, which come to an average of 10.6 permissions per application. On the other hand, 970 benign applications used 3,838 permissions, which come to an average of around 3.95 permissions per application. These results confirm that, on average, the number of permissions requested by a benign application is less than the number of permissions requested by a malicious application.

Figure 5(c) shows the top 15 dynamic operations performed by the analyzed malicious applications. As shown in the figure, there are many operations that are dominantly performed by the malicious applications. These include BroadcastReceiver(BOOT_COMPLETED), OpenNetworkConnection:80, and DataLeak_Network. Applications with malicious intents use BOOT_COMPLETED broadcast receiver to receive a notification whenever an Android device boots up so that they can perform the intended malicious activity or launch malicious services that keep running in the background. Another deciding factor is data leakage through network, which has a high occurrence in our malicious dataset, i.e., 254 as compared to 0 in the analyzed benign ones. Figure

 ISBN: 978-1-61208-376-6

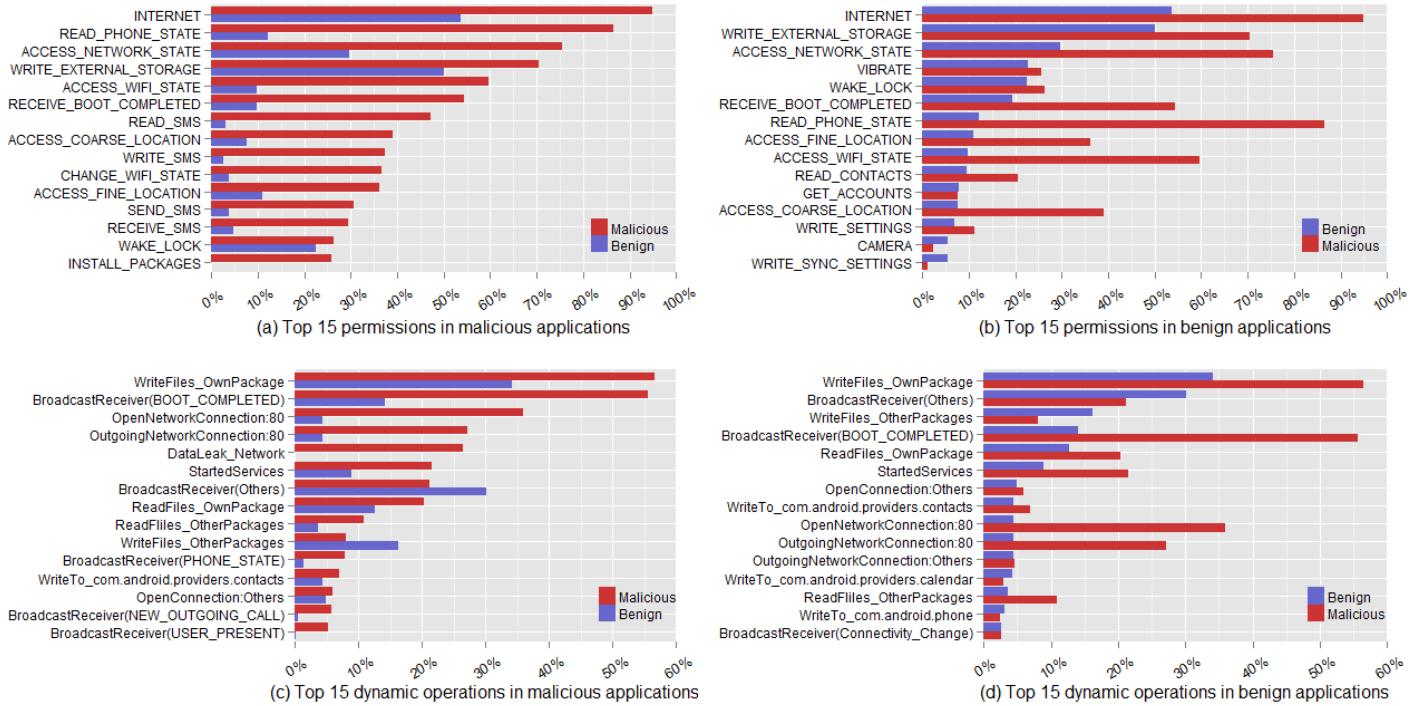


Figure 5. Most frequently used permissions and dynamic operations for the analyzed Apps

5(d) shows the top 15 dynamic operations performed by the benign applications in our dataset.

B. *Cyber-intelligence*

One of the main objectives of cyber-intelligence is to track sources of online threats. In this work, we used the URLs and IP addresses recorded during the static and dynamic analysis of malware Apps to produce a graphical representation of the geographical locations of possible malicious servers (and their ISPs) that communicate with these malicious Apps. Figure 6 shows a sample output of this analysis (IP addresses are not shown for privacy and liability concerns).

Figure 6. Geographical Presentation of the locations of suspected IPS

C. *Malware detection*

Throughout this experiment, we incorporated 134 static, 285 dynamic and 400 n-grams based features. We performed our classification task on 1932 Android applications using different combinations of these features, namely static analysis features, dynamic analysis features, n-grams based features, combination of static & dynamic features (S+D), combination of static & n-grams based features (S+N) and combination of dynamic & n-grams based features (D+N). We also combined features from all three analysis techniques, i.e., static, dynamic and *n*-grams (S+D+N) and performed feature space reduction using *classwise document frequency* to obtain a feature-set containing the top features for classification. We employed five different algorithms supported by WEKA [27] for classification with 10-fold cross-validation: SMO [28], IBK [28], J48 [28], AdaBoost1(J48 as base classifier) [28], and RandomForest [28]. Our experimental results show that AdaBoost1 and RandomForest models achieve a better accuracy compared to the other models. Figure 7 shows the results obtained for the five different feature sets in terms of accuracy, precision, and recall. From Figure 7(a), it is clear that *n*-gram features using AdaBoost1, D+N features using AdaBoost1 and S+D+N features using RandomForest provide the highest accuracy $\approx$98%. Figure 7(b) and Figure 7(c) shows the corresponding precision and recall, respectively. The low accuracy obtained when using dynamic analysis only can be explained by noting that, throughout our dynamic analysis process, we do not interact with the applications with carefully chosen gestures. Consequently, there is no guarantee that we check complete paths that can be traversed by the application or even a good portion of it. Furthermore, the short dynamic analysis interval might not be enough to trigger some of the bad events performed by malicious Apps. On the other hand, it should not be noted that the relatively high accuracy obtained with the combined features should also be interpreted with care since it might have resulted because of the limited variance in the

 ISBN: 978-1-61208-376-6

characteristics of the analyzed samples.

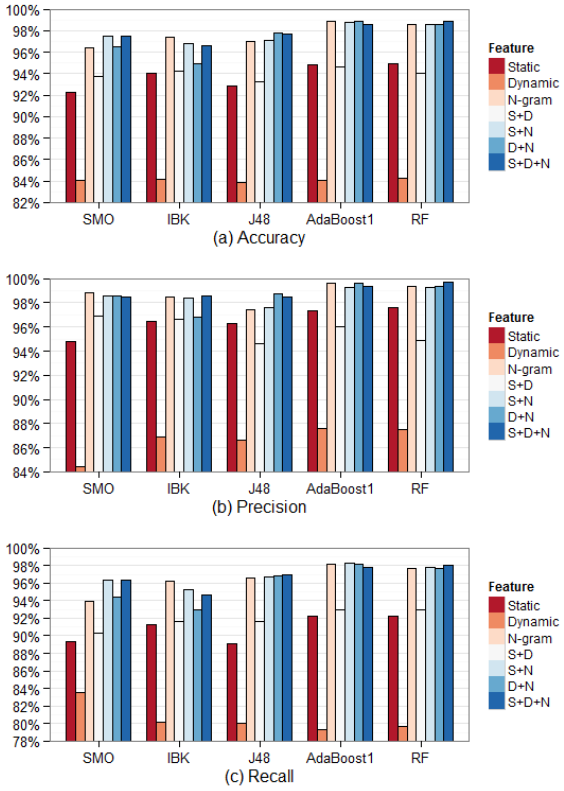


Figure 7. The classification results

VI. Conclusion

The increasing popularity of the Android operating system has led to sudden escalation in Android malware. In this work, we developed a framework to analyze Android applications using static and dynamic analysis techniques (*AndroSAT*). The effectiveness of *AndroSAT* was tested by analyzing a dataset of 1932 applications. The information obtained from the produced analysis reports proved to be very useful in many Android security related applications. In particular, we used the data in these reports to perform three case studies: analyzing the frequency of use of different Android permissions and dynamic operations for both malicious and benign Apps, producing cyber-intelligence information, and maleware detection. The implemented prototype can be further extended to allow for more useful add-ons that can be used to provide further investigation of the security of Android applications.

References

[1] "Third Annual Mobile Threats Report," 2013, URL: http://www.juniper.net/us/en/local/pdf/additional-resources/3rd-jnpr-mobile-threats-report-exec-summary.pdf [accessed: 2014-09-05].

[2] Q. Li and G. Clark, "Mobile security: a look ahead," Security & Privacy, IEEE, vol. 11, no. 1, 2013, pp. 78–81.

[3] C. Miller, "Mobile attacks and defense," Security & Privacy, IEEE, vol. 9, no. 4, 2011, pp. 68–70.

[4] "Kaspersky: forget lone hackers, mobile malware is serious business," Feb. 2014, URL: http://www.theguardian.com/technology/2014/feb/26/kaspersky-android-malware-banking-trojans [accessed: 2014-09-05].

[5] "Android Permissions," URL: http://developer.android.com/reference/android/Manifest.permission.html [accessed: 2014-09-05].

[6] T. Blasing, L. Batyuk, A. D. Schmidt, S. A. Camtepe, and S. Albayrak, "An android application sandbox system for suspicious software detection," in Proceedings of the 5th international conference on Malicious and unwanted software (MALWARE). IEEE, 2010, pp. 55–62.

[7] D. J. Wu, C. H. Mao, T. E. Wei, H. M. Lee, and K. P. Wu, "Droidmat: Android malware detection through manifest and api calls tracing," in Proceedings of the Seventh Asia Joint Conference on Information Security (Asia JCIS). IEEE, 2012, pp. 62–69.

[8] A. Reina, A. Fattori, and L. Cavallaro, "A system call-centric analysis and stimulation technique to automatically reconstruct android malware behaviors," EuroSec, Apr. 2013.

[9] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behavior-based malware detection system for android," in Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices. ACM, 2011, pp. 15–26.

[10] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang, "Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets." in NDSS, 2012.

[11] M. Spreitzenbarth, F. Freiling, F. Echtler, T. Schreck, and J. Hoffmann, "Mobile-sandbox: having a deeper look into android applications," in Proceedings of the 28th Annual ACM Symposium on Applied Computing. ACM, 2013, pp. 1808–1815.

[12] M. Alazab, V. Monsamy, L. Batten, P. Lantz, and R. Tian, "Analysis of malicious and benign android applications," in Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW). IEEE, 2012, pp. 608–616.

[13] W. Enck, P. Gilbert, B. G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information flow tracking system for real-time privacy monitoring on smartphones," Communications of the ACM, vol. 57, no. 3, 2014, pp. 99–106.

[14] S. Jain and Y. K. Meena, "Byte level n–gram analysis for malware detection," in Computer Networks and Intelligent Computing. Springer, 2011, pp. 51–59.

[15] D. K. S. Reddy and A. K. Pujari, "N-gram analysis for computer virus detection," Journal in Computer Virology, vol. 2, no. 3, 2006, pp. 231–239.

[16] "Android version history," URL: http://en.wikipedia.org/wiki/Android_version_history [accessed: 2014-09-05].

[17] S. Brahler, "Analysis of the android architecture," Karlsruhe institute for technology, 2010.

[18] "The Android Asset Packaging Tool," URL: http://developer.android.com/tools/building/index.html [accessed: 2014-09-05].

[19] "Android APKTool: A tool for reverse engineering Android apk files," URL: https://code.google.com/p/android-apktool/ [accessed: 2014-09-05].

[20] "Apk2java: Batch file to automate apk decompilation process," URL: http://code.google.com/p/apk2java/ [accessed: 2014-09-05].

[21] "Android Broadcast Receiver," URL: http://developer.android.com/reference/android/content/BroadcastReceiver.html [accessed: 2014-09-05].

[22] "DroidBox: Android Application Sandbox," URL: https://code.google.com/p/droidbox/ [accessed: 2014-09-05].

[23] "MonkeyRunner," URL: http://developer.android.com/tools/help/monkeyrunner_concepts.html [accessed: 2014-09-05].

[24] A. Lineberry, D. L. Richardson, and T. Wyatt, "These aren't the Permissions you're Looking for," in DEFCON 18, Las Vegas, NV, 2010.

[25] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," in Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP). IEEE, 2012, pp. 95–109.

[26] "VirusTotal," URL: https://www.virustotal.com/ [accessed: 2014-09-05].

[27] "WEKA," URL: http://www.cs.waikato.ac.nz/ml/weka/ [accessed: 2014-09-05].

[28] I. H. Witten and E. Frank, Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2005.

 ISBN: 978-1-61208-376-6

Involvers' Behavior-based Modeling in Cyber Targeted Attack

Youngsoo Kim and Ikkyun Kim
Cyber Security Research Laboratory
Electronics & Telecommunications Research Institute
Daejeon, Korea
e-mail: {blitzkrieg, ikkim21}@etri.re.kr

Abstract— **Cyber targeted attack has sophisticated techniques using malwares to exploit vulnerabilities in systems and an external command and control is continuously monitoring and extracting data off a specific target. Since this attacking process is working continuously and uses diverse malicious codes and attacking routes, it is considered to be difficult to detect in advance. In this paper, we categorized cyber targeted attacks into four steps and defined potential behaviors of involvers like attackers or victims, in order to make a model. Each behavior of our model can include a couple of methods. Furthermore, we applied our behavior-based model to the real targeted attacks, "3.20 South Korean Malware Attack" and "The Targeted Attack for SK Communications".**

Keywords-APT; Targeted Attacks; Behavior-based Modeling; Malicious Codes; 3.20 DarkSeoul.

I. INTRODUCTION

Cyber targeted attack, which is also known as Advanced Persistent Threat (APT), is a kind of intelligent attacking method having a goal of acquiring classified information or control of critical infrastructure, by penetrating networks of targets in a stealthy way and staying there in the long term. It usually targets organizations or nations for business or political motives. It has complicated techniques using malicious codes to take advantage of vulnerabilities in systems and an outer command and control is constantly observing and deriving data from a specific target [1]. This attacking method is working continuously and utilizes various malwares and attacking routes, so it is deemed to be hard to discover beforehand. In Section 2, we classified advanced persistent threat and described possible behaviors of involvers like attackers or victims for modeling. Each behavior of our model can include several methods. In Section 3, we introduced a couple of real targeted attacks and indicated that our behavior-based model is fit to depict them and described some useful cases of proposed modeling map, and conclude with some remarks and further works in Section 4.

II. EACH STEP OF BEHAVIORS/METHODS FOR CYBER TARGETED ATTACKS

Cyber targeted attacks can be divided into 4 phases: The preparation phase, the penetration phase, the control phase, and the achievement phase. Figure 1 depicts the detailed behaviors of attackers and victims.

In the preparation phase, attackers collect and analyze diverse data of targeting web sites, and they hack servers with vulnerabilities and make them Command and Control (C&C) servers. Also, they use various ways for triggering download of malicious codes. In the penetration phase, attackers try to acquire user authority using diverse methods and user's Personal Computers (PCs) can be infected with malicious codes by running malicious attached files, updating falsified software, or using unauthorized USBs. In the control phase, attackers try to acquire additional user authorities and collect additional information using various ways. They can also control victimized systems with backdoors or web-shells and spread malicious codes to all connected devices. In the achievement phase, attackers can acquire critical information using remote commands or web-mails and they can also emasculate systems using automatic termination or remote starting.

A. The Preparation Phase

If attackers decide attacking targets, they visit targeting web pages for looking into vulnerabilities [2]. They could acquire user information by falsifying URLs of targets. First, they register the targeting website and try to read web-board messages requiring the higher-level accessing authority. Even though they are rejected to access, they can watch an URL of web-board message which they want to read using right-hand mouse-clicking. And then, they try to falsify that URL to acquire user information without reading authority.

They can collect and analyze information related to targeting victims using web-crawlers or bots, which look around enormous web pages, including web sites providing Social Network Service (SNS), such as Facebook, Twitter, etc., to get information. They can also use meta-search engines connecting diverse searching engines for same reason.

Attackers hack servers (e.g., web-board, web-mail server, and web-disk) with vulnerabilities and make them C&C servers. After they acquire authorities of accessing the targeting servers using malicious codes, they falsify them to play a role of the C&C servers.

After that, attackers prepare for inducing the victims to download malicious codes in diverse ways. They could send e-mails including attached malicious codes to targeting users or attach them to web-board messages for triggering downloads. Also, they could falsify software of updating servers in case of installing. If they can do it, users download them and install falsified updating software unconsciously. They could use Cross Site Script (XSS) vulnerabilities in two

 ISBN: 978-1-61208-376-6

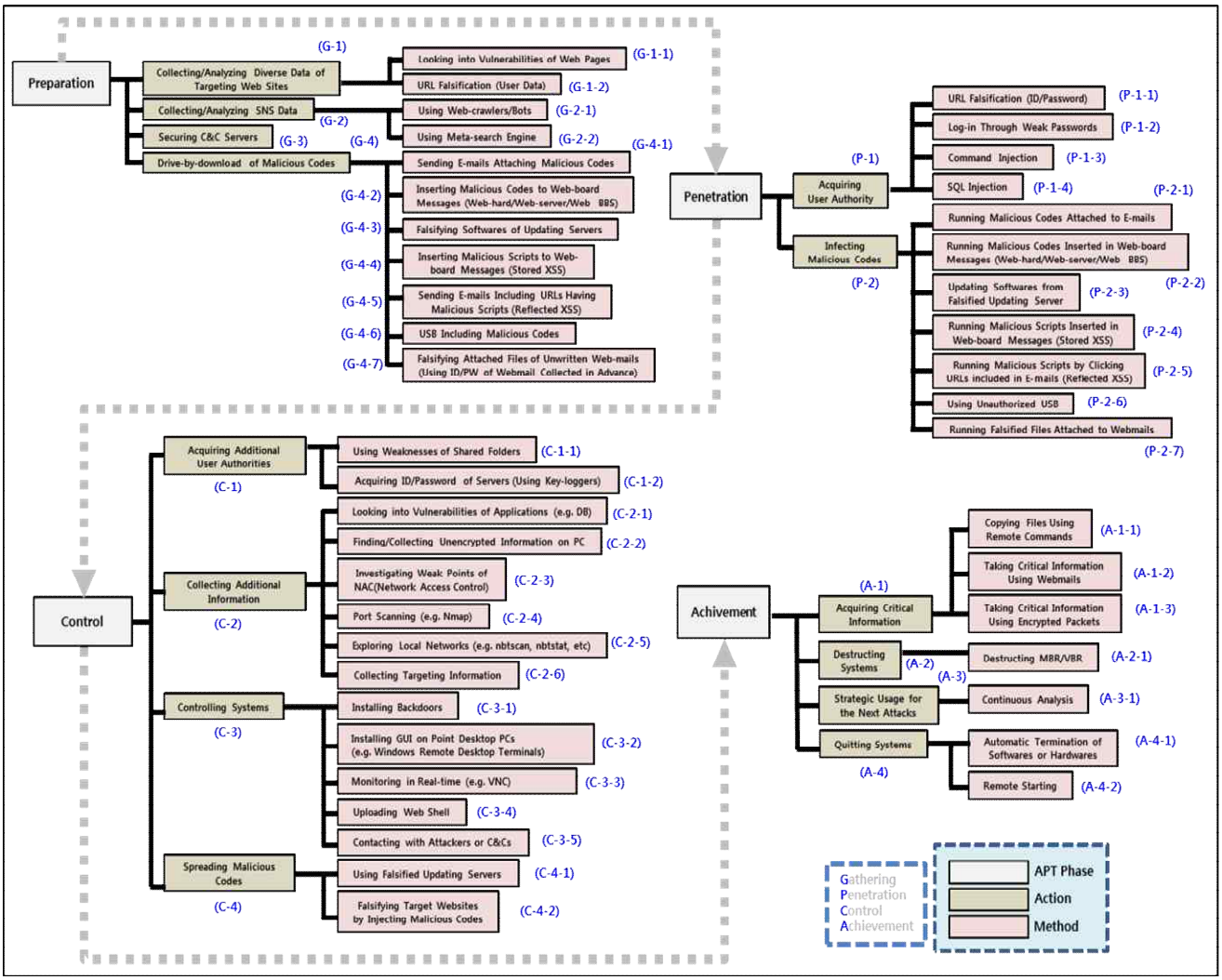


Figure 1. Each Step Behaviors/Methods for Cyber Targeted Attacks

ways, i.e., Stored XSS and Reflected XSS. Attackers can insert malicious scripts to web-board messages using XSS-vulnerability of the targets for triggering infection of malicious scripts. After they find that XSS vulnerability, they write web-board messages with malicious scripts and post those messages on the web-board [3]. They could also send e-mails including URL links of targeting web sites having malicious scripts to users for infection. After they find the XSS vulnerability of targeting web sites, they write e-mail messages including URL links having malicious scripts and send them to the targeting users in order to be infected by clicking those URL links. Sometimes, they prepare USB sticks including malicious codes and putting them on somewhere in the targeted area [4]. Also, attackers could replace attached files of unwritten web-mails with malicious files using ID/password of web-mail collected in advance. After they collect many pairs of ID/password, they check unwritten web-mails using collected ID/passwords and replace attached files with malicious files.

B. *The Penetration Phase*

After preparing, attackers try to acquire user authority using diverse methods, such as URL falsification, weak passwords, command injection, or SQL injection. They acquire user authority by falsifying the URL of targeting web site. First, they check a pattern of URL attributes of the targets and repeatedly access to those web sites by typing randomly changed URL links. They could login and acquire user authorities if randomized user-codes are matched. Sometimes, attackers acquire weak passwords using password cracking tools. Additionally, they run local system commands remotely because of the vulnerability of insufficient authorizing input variables [5]. First, they check the address of targeting web site. After injecting a system command to this address, they could access with this changes address. And then, they could see and get some system information. Finally, attackers could watch, falsify, or delete database data by fabricating input data of database application [6]. It occurs since database applications do not check the validity of input data from users. They could

 ISBN: 978-1-61208-376-6

bypass certificating process for users or administrators using SQL injection. After checking if they could input special characters to log-in windows, they input SQL commands to log-in windows and then, they become able to log-in without a certificating process.

User PCs can be infected with malicious codes in various methods. A user runs attached files of received e-mails including malicious codes and he becomes infected. When a user receives an e-mail attaching malicious files, he reads the message of that e-mail. And then, if he activates malicious attached files, he will become infected. If a user runs malicious files attached in web-board messages and he becomes infected. When a user clicks a message of web-hard, web-server, or web-board having attached malicious files, he runs attached malicious files. As a result, he becomes infected. Furthermore, a user could be infected through automatic activating of updating software falsified in advance. First, updating software having been falsified in advance are executed automatically. If a user activates the updating process or automatic updates are activated, the user's PC is infected. A user could be infected by reading a web-board message including malicious scripts. After accessing the web, a user read a web-board message having malicious scripts. And then, contents of the web-board message including malicious scripts are sent to the user. Finally, the user's PC is infected and the user's cookies are sent to the attacker. Sometimes, if a user receives an e-mail with a malicious URL link and accesses to that URL link, he could be infected by malicious scripts. First, he receives an e-mail including a malicious URL link and clicks that link. As a result, he can access a link having a malicious script, and the user PC is infected since the malicious script is activated. Finally, the user's cookies are sent to the attacker. Local systems could be infected by bringing and executing infected unauthorized USB sticks. If a user brings infected unauthorized USB sticks and he put them on local systems like PCs or servers, local systems are infected. A user could also be infected by running a falsified attached file of a web-mail. If a user logs in his web-mail account and checks an attached file of web-mail falsified by an attacker in advance, his PC is infected by executing the attached file.

C. The Control Phase

To achieve final goals, attackers try to acquire additional user authorities using weakness of shared folders or key-loggers and collect additional information using various ways such as port scanning, weak points of Network Access Control (NAC), vulnerabilities of applications, etc. Furthermore, they can control victimized systems by installing backdoors or uploading web-shells and spread malicious codes to all connected devices by falsifying updating servers or web servers of targets.

Attackers could access the shared PC if a pair of ID/password of the infected PC is same as that of the shared PC. After that, they could access the agent server at a time, if they install a scheduler like at.exe at the shared PC. If they log in with a pair of ID/password of infected PC, they try to access to the shared PC using the same pair. In case that the pair is the same, they could access. If they install a scheduler at the shared PC, they could access the agent server at a time. Additionally, all keyboard-typing logs on infected PCs could be recorded in real-time using key-loggers. First, attackers install a key-logging program on infected PCs. If users make use of the infected PCs, the key-logging program records all keyboard-typing logs in real-time. After receiving recorded logging data, attackers check and get some pairs of ID/password.

Attackers can get additional information by analyzing vulnerabilities of various web applications such as SQL injection, file upload, XSS, path traversal, cookies, parameter manipulation, configuration setting errors, admin page, backup/temporal files, etc. Attackers can also collect additional information by finding unencrypted folders or files in infected PCs. After logging in the infected PCs, they search all folders or files. If they find unencrypted ones, they can get additional information. Sometimes, attackers can find vulnerabilities of NAC by checking security policies or security solutions, and then, can access to the local network without authority checks. They find vulnerabilities of NAC by checking security policies or security solutions in order to access to the targeting local network. And then, they can access servers and check data to get information. Attackers can access the infected PCs and scan all ports to check whether they are open or not. They access the infected PCs and check opened ports using nmap command or port-scanning tools to get information. They can also check the status of local network, for example some PCs are grouping or some devices are powered off, by scanning such as nbtscan, nbtstat, etc. Additionally, attackers harvest target information related to the final goals. They access to the admin computer dealing with local information and find and harvest the target information.

Attackers can hide backdoor files in advance, and they get root authority by activating them to control the target system [7]. First, they access the infected PCs and make backdoor files. Backdoor files are compiled at temp directory. And then, they run backdoor files in a general account, they get root authority and become to be able to control the target system. Attackers can also control the targeting system by installing terminal programs for remote control and database accessing tools, after finding PCs operating 24 hours a day. First, they find PCs operating 24 hours a day, and they install database related tools and terminal programs for remote control. And then, they check database and logs on the mainframe computer. As a result, they become to control the targeting system and gets information. Sometimes, attackers acquire critical information and control the targeting system by monitoring the infected PCs with Virtual Network Computing (VNC). After accessing the infected PCs, they install a VNC program. If administrators or developers use the infected PCs, attackers can watch what they do with VNC. As a result, they can control the targeting system and acquire critical information. Attackers can acquire control of the target system by uploading a web-shell, a web script file (e.g., asp, jsp, php, and sci) usually made maliciously in order to run instructions on the targeting web server remotely [8]. First, they make a web script file and upload it on local web-board. After searching a URL enabling them to move

 ISBN: 978-1-61208-376-6

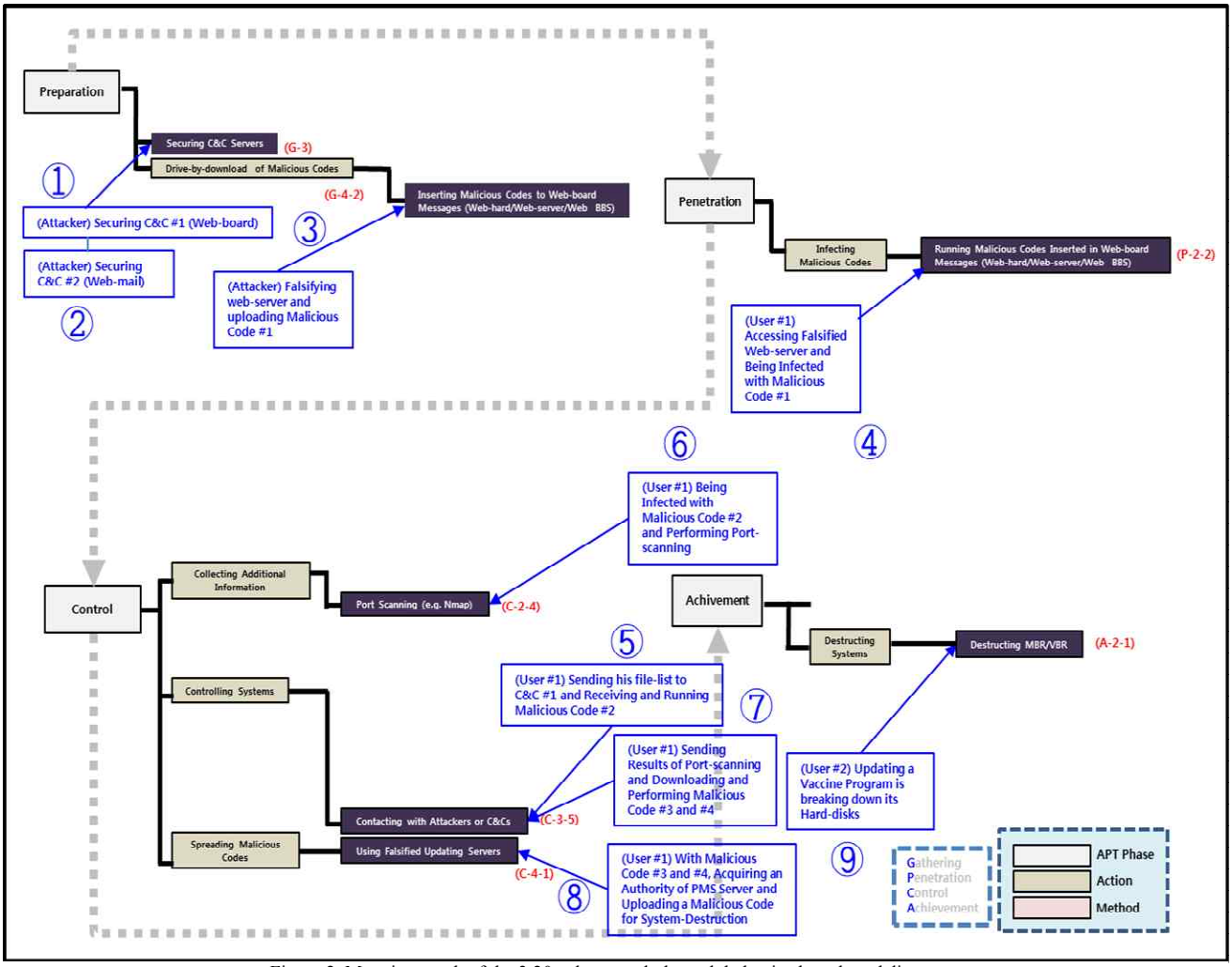


Figure 2. Mapping result of the 3.20 cyber-attack through behavior-based modeling map

into the uploading location using file attributes, they access the shell by entering this URL. And then, they get some system information by using some commands. Additionally, attackers connect the infected PCs with themselves or C&Cs for sending instructions or additional malicious codes to the infected PCs or receiving the targeting system information from them. After getting control of system, they send instructions or additional malicious codes to the infected PCs or receive the targeting system information by way of C&Cs.

Attackers can spread malicious codes using the updating server including falsified updating software. All PCs activating automatic updates can be infected. They can also trigger infection of malicious codes through adding falsified web pages or banners enabling users to access and click on them. They add web pages or banners including malicious codes to the target web server. And then, if users visit the targeting web site, their PCs are infected.

D. The Achievement Phase

In this phase, attackers can acquire critical information by copying files with remote commands or using web-mails or encrypted packets. Also, they can emasculate systems by destructing Master Boot Record (MBR)/Volume Boot Record (VBR), for example, or quit systems using automatic termination or remote starting.

Attacker can control infected PCs and take away critical files from them using a remote command like scp or web-mails. They can also encrypt packets of critical data using packet-extracting commands/tools and take them. Attackers run destruction commands for MBR/VBR of infected PCs and preclude system booting. Additionally, they can catch the following decisive opportunities by monitoring and analyzing infected PCs. Sometimes, attackers terminate the targeting system using commands for automatic termination of specific software or hardware or quit the targeting system using commands for remote starting.

III. MODELLING FOR REAL CASES

We applied our behavior-based model to two real targeted attacks. One was occurred in July of 2011 and the other was occurred in March 20, 2013.

 ISBN: 978-1-61208-376-6

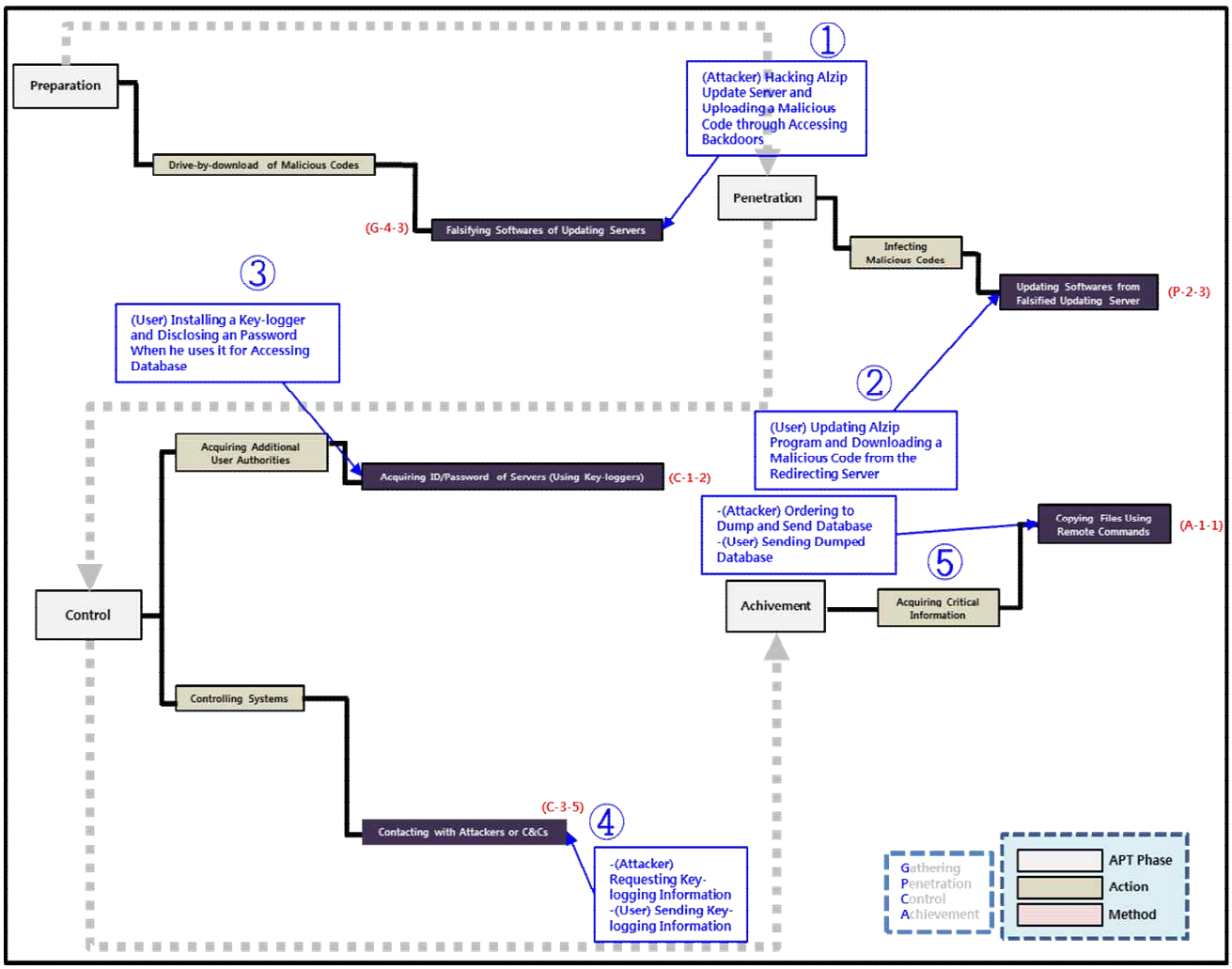


Figure 3. Mapping result of the Targeted Attack for SK Communications through behavior-based modeling map

A. *Modelling for 3.20 South Korean Malware Attack*

The attack, dubbed DarkSeoul, against South Korean media and banking organizations severely disrupted a handful of organizations with a coordinated distribution of "wiper" malware designed to destroy data on hard drives and render them unbootable [9]. It is known that the malware will overwrite MBR and VBR. The records and files overwritten by the malware so far have been wiped with patterns of 'HASTATI' or 'PR!NCPES'. We referenced some analysis reports and described detailed processes of this attack as follows [10]. We mapped 8 steps of this attack to potential behaviors we categorized. Figure 2 depicts the mapping result of the 3.20 cyber-attack.

1. An attacker secures C&Cs using vulnerabilities of web-boards or web-mails.
2. The attacker falsifies the (C&C #1 and C&C #2) web-server and uploads malicious code #1.
3. A user (User #1) accesses falsified web-server and is infected with malicious code #1.
4. User #1 sends his file-list to C&C #1 and receives and runs malicious code #2.
5. User #1 is infected with malicious code #2 and performs port-scanning.
6. User #1 sends results of port-scanning and downloads/performs malicious code #3 and #4.
7. With malicious code #3 and #4, user #1 acquires an authority of Patch Management System (PMS) server and uploads a malicious code for system-destruction.
8. Other users update vaccine programs and their hard-disks are broke down.

B. *Modeling for Targeted Attack for SK Communications*

Between 18 and 25 July 2011, attackers infected over 60 SK Communications computers and used them to gain access to the user databases. They infected these computers by first compromising a server belonging to a South Korean

 ISBN: 978-1-61208-376-6

software company, used to deliver software updates to customers (including SK Communications). Attackers modified the server so that the SK Communications computers would receive a trojaned update file when they conducted their routine checks for software updates [11]. We mapped 5 steps of this attack to potential behaviors we categorized. Figure 3 depicts the mapping result of the 3.20 cyber-attack.

1. An attacker hacks Alzip update server and uploads a malicious code through accessing backdoors [12].
2. A user updates Alzip program and downloads a malicious code from the redirecting server unconsciously.
3. The key-logger is installed in that user's computer and user's password can be logged when he uses it for accessing database.
4. The attacker requests and receives key-logging information.
5. The attacker orders to dump and send database and gets it from the key-loggers of user's PC.

This proposed involver's behavior-based model of cyber targeted attack could be useful for the following cases.

First, attacking methods are very diverse, so our model can be a basic scale for deciding whether the attack is cyber targeted attack or not. Second, generally cyber targeted attack can occur over a long period of time. If some behaviors related to cyber targeted attack can be found in its middle stages, the following potential behaviors can be prevented in advance, referencing to our model. Third, if some attacking behaviors are found in its middle stages or final stages, we can guess what happened in the beginning stages using our model. Fourth, since our model includes analysis points at each phase, it can be a guidance map for analyzing causes of hacking accidents. If cause analysis can be achieved rapidly, services delayed due to this hacking accident could be restored faster. Finally, according to 44 methods of our model, detailed analyses of devices relating to involvers can be done.

IV. CONCLUSION AND FUTURE WORK

We categorized cyber targeted attacks into 4 steps and defined potential behaviors of involvers like attackers or victims, in order to make a model. Each behavior of our model can include a couple of methods. Furthermore, we applied our behavior-based model to the real targeted attacks, "3.20 South Korean Malware Attack" and "The Targeted Attack for SK Communications" and described use cases. For testing, we are building the cyber hacking test-bed including routers, switches, servers, PCs, and notebooks. We have plans to make some APT-scenarios similar to real targeted attack like DarkSeoul, and implement them on the cyber hacking test-bed to verify our model.

REFERENCES

[1] N. Virvilis and D. Gritzalis, "The big four-what we did wrong in protecting critical ICT infrastructures from Advanced Persistent Threat detection?," The Eighth International Conference on Availability, Reliability & Security (ARES 2013), IEEE Press, Sep. 2013, pp. 248-254, doi:10.1109/ARES.2013.32.

[2] W. Gary and S. Zhendong, "Sound and precise analysis of web applications for injection vulnerabilities," Conference on Programming Language Design and Implementation (PLDI 2007), ACM, Jun. 2007, pp. 32-41, ISBN: 978-1-59593-633-2.

[3] M. Michael and L. Monica, "Automatic generation of XSS and SQL injection attacks with goal-directed model checking," Proc. of the Conference on Security Symposium (SS 2008), USENIX Association Berkeley, Jul. 2008, pp. 31-43.

[4] C. Harlan and A. Cory, "Tracking USB storage: Analysis of windows artifacts generated by USB storage devices," Digital Investigation, vol. 2, Jun. 2005, pp. 94-100, doi:10.1016/j.diin.2005.04.006.

[5] S. Zhendong and W. Gary, "The essence of command injection attacks in web applications," Conference record of the 33rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Language (POPL 2006), ACM, Jan. 2006, pp. 372-382, ISBN: 1-59593-027-2.

[6] W. Halfond, J. Viegas, and A. Orso, "A classification of SQL-injection attacks and countermeasures," Proc. of the IEEE International Symposium on Secure Software Engineering, IEEE, Mar. 2006, pp. 13-15.

[7] S. Gaspers and S. Stefan, "Backdoors to Satisfaction," The Multivariate Algorithmic Revolution and Beyond, Springer Berlin Heidelberg, pp. 287-317, Nov. 2012, ISBN: 978-3-642-30890-1.

[8] A. Straniery and Z. John, "WebShell: The development of web based expert systems," Research and Development in Intelligent Systems XVIII. Springer London, Dec. 2001, pp. 245-258, ISBN: 978-1-85233-535-9.

[9] US-CERT, "South Korean Malware Attack", 2013. https://www.us-cert.gov/sites/default/files/publications [Retrived: Oct, 2014].

[10] IssueMakersLab, "Operation 1Mission aka 3.20 DarkSeoul," http://www.issuemakerslab.com [Retrived: Oct, 2014].

[11] L. Moon-young, "Personal Information Hack Traced to Chinese IP address," The Hankyoreh Media Company, 2011. http://english.hani.co.kr/arti/english_edition/e_national/491514.html [Retrived: Oct, 2014]

[12] Altools, http://www.altools.com [Retrived: Oct, 2014]

 ISBN: 978-1-61208-376-6

Test Case Generation Assisted by Control Dependence Analysis

Puhan Zhang
China Information Technology Security Evaluation Center
Beijing, China
zhangph2008@gmail.com

Qi Wang
Renmin University of China
Beijing, China
China Telecom Corporation Beijing Company
Beijing, China
wangq@163.com

Guowei Dong
China Information Technology Security Evaluation Center
Beijing, China
dgw2008@163.com

Bin Liang, Wenchang Shi
School of Information
Renmin University of China
Beijing, China
{liangb, wenchang}@ruc.edu.cn

***Abstract*—The paper proposes and develops a new test case generation tool named Symbolic Execution & Taint Analysis (SYTA) that can capture implicit information flows by control dependence analysis. When running, SYTA traces execution paths to track constraints on symbolic variables. Some equivalence relationship asserts will be constructed to store the equivalence information among variables for control dependence analysis. If a security sink is reached, SYTA builds a constraint, path conditions and equivalence relationship asserts, which are to be sent to a constraints solver. The test cases will be generated from possible counterexamples in constraint solving. Compared with traditional static analysis tools, SYTA can track implicit information flows, and generate test cases by control dependences analysis effectively.**

Keywords-test case generation; control dependence; implicit information flow; symbolic execution

I. INTRODUCTION

Nowadays, test case generation has become the most important step of code testing, which is usually realized by the symbolic execution approach. If there exists a bug, the test cases can help programmers to find the spot that causes the error.

A traditional Fuzzing approach is a form of blackbox testing which randomly mutates well-formed inputs and use these variants as test cases [1][2]. Although Fuzzing can be remarkably effective, the limitations of Fuzzing are that it usually provides low code coverage and cannot drive deeper into programs because blind modification destroys the structure of inputs [3]. In a security context, these limitations mean that potentially serious security bugs, such as buffer overflows, are possibly missed because the code containing the bugs is even not exercised.

Combining general static analysis with taint analysis to test applications and draw test cases is presently the hottest research technique, such as TaintScope [6]. Taint analysis allows a user to define the taint source and propagate the taint following specific propagation policy during execution, and finally, trigger a particular operation if the predetermined security sink is hit.

Unfortunately, this smart Fuzzing technique bears many pitfalls [4], among which missing the implicit information flows is the most critical one. Contrary to explicit information flows caused by direct assignment, implicit information flows are a kind of information flow consisting of information leakage through control dependence. The example shown in Figure 1 discloses the nature of implicit information flows. There is no direct assignment between variables *h* and *l* in the sample program, but *l* can be set to the value of *h* after the if-then-else block by control dependence. Even though the early attention and definition of the implicit flow problem dated back to 1970's [5], no effective solution has been found. Some newly-developed tools, such as TaintScope [6], detour implicit information flows and limit their analysis only to explicit information flows, which incur the following three problems:

- Missing implicit information flows may lead to a under-tainting problem and false negative. As a result, the security vulnerabilities caused by control dependence will not be detected. Especially, it is critical to capture implicit flows in privacy leak analysis.
- Control dependence is also a common programming form in benign programs. For example, some routines may use a switch structure to convert internal codes to Unicode in a Windows program such as the following code segment. switch(x){ case a: y = a; break; case b: y = b; break; ……}. It indicates that it is necessary to analyze the implicit information flows for common software testing.
- To counter the Anti-Taint-Analysis technique, implicit information flows must be analyzed effectively [7]. Malware can employ control dependence to propagate sensitive information so as to bypass traditional taint analysis.

To address these limitations and generate test cases with tainting techniques, we propose and develop a new tool called Symbolic Execution & Taint Analysis (SYTA), which can generate test cases by considering implicit information flows. Compared with traditional static analysis tools, SYTA can track implicit information flows and generate test cases

```
1:h := h mod 2;
2:if h = 1 then
3:   l := 1;
4:else
5:l := 0;
6:end if
```

Figure 1. A sample program of implicit information flow

 ISBN: 978-1-61208-376-6

by control dependences analysis effectively. Though it is hard to say what percentage of a program can be classified as implicit information flow, it may reveal some vulnerabilities that explicit information flow is unable to.

The rest of the paper is organized as follows. Section 2 briefly analyzes the target problem. Section 3 discusses our methodology and design of SYTA. Section 4 evaluates our approach. Section 5 summarizes related work. Finally, Section 6 concludes the paper and discusses future-work directions.

II. PROBLEM ANALYSIS

This section describes the problem we encounter by walking the readers through the testing of a sample program shown in Figure 3 (a). Despite its small size, it illustrates the most common characteristics of implicit information flows. There exist three bugs related to control dependence in the sample program.

1) Array bound overflow in line 29. The program implies that variable *k* will be equal to variable *i* under a specific condition. If 2 is assigned to *i* by users, *k* will be set to 2 through four control branches, including three 'if' and one loop statements. In line 28, the value to which pointer *p* points is 4. Eventually, an array bound overflow will be triggered when dereferencing *p* as the index of array *a* in line 29.

2) Divide-by-zero in line 30. If 3 is assigned to variable *i* by users, through several control branches, **p* will be set to 0 in line 28, then the divisor *t* becomes 0 in line 30.

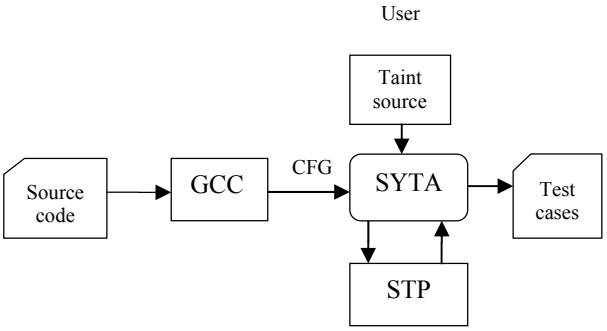
3) Denial of service in line 31. If 1 is assigned to variable *i*, the result of a DoS attack may occur in the program in line 31.

In traditional analysis tools, the test cases cannot be generated for above bugs due to the absence of control dependence analysis. Take EXE [8] and KLEE [9] as examples, they are totally based on explicit information flows analysis. When being applied to the sample program, though variable *i* is symbolically executed and analyzed, those tools can not produce effective test cases, because there are not any direct assignment relationships among *i* and some other variables, such as *k*, *t*, and *p* etc.

Our solution is to take implicit information flows into consideration, in which the flow of taint is propagated from variable *i* to variables *j*, *tmp* and *k* (in line 18, 25 and 30, respectively, in the source code). Variable *p* in line 33 is tainted because of the data flow and it is possible to identify the bugs and automatically obtain the test case to hit them.

III. METHODOLOGY

SYTA, as a test case generator, actually functions as a combination of an intermediate language interpreter, a symbolic execution engine and a taint analyzer. During each symbolic execution, some lists are built to store information for taint propagation. Test programs are firstly parsed by a compiler front-end and converted to an intermediate language. The corresponding Control Flow Graphs (CFGs) are constructed as the inputs of SYTA. SYTA will traverse each CFG and run symbolic execution. It will perform two kinds of taint propagations during symbolic execution, collect symbolic path conditions, record the equivalence information among variables and generate Satisfiability Modulo Theories (SMT) constraints eventually. An SMT solver will be employed to solve and check these constrains to detect potential bugs. If some bugs are found, test cases will be generated and reported.



Note: STP is an SMT solver.

Figure 2. The architecture of SYTA

A. Overview

The core of SYTA is an interpreter loop that selects a path, composed of basic blocks and directed by edges of CFG, to symbolically execute each statement of the basic blocks and perform two kinds of taint propagations (explicit and implicit). The loop continues until no basic blocks remain, and generates test cases if some bugs are hit. The architecture is illustrated in the Figure 2.

For two kinds of taint analysis, we maintain the Explicit Information Flow Taint (EFT) lists and Implicit Information Flow Taint (IFT) lists. Besides, an Equivalence Relationships (ER) list is maintained to record equivalence information among variables in condition statements for control dependence analysis.

At the very beginning of testing, users appoint some interested variables as the taint sources which are recorded into the EFT and IFT lists in proper forms. The two lists involve different taint propagation policies that we design for explicit and implicit information flows respectively.

When a security sink is encountered, SYTA will invoke an SMT solver to carry out a query considering the operation related to current security sink. Current path conditions and expressions drawn from the ER list will act as the context of the query, namely, asserts of solving. By running the query, SYTA checks if any input value exists that may cause a bug. If a bug is detected, the SMT solver will produce a counterexample as a test case to trigger the bug.

B. Implicit Information Flow Taint Propagation

The intuition of taint propagation over implicit information flows can be illustrated using a sample program shown in Figure 4.

In this sample program, a conditional branch statement *br*, namely if ($i >= 4$) in line 5, decides which statements *st* should be executed ($j = 5$ in line 6 or $j = 0$ in line 9). The

 ISBN: 978-1-61208-376-6

value of *i* affect the value of *j*. Therefore, based on control dependence, the taint state should be propagated from the source operand of *br*, namely the variable *i*, to *st*'s destination operands, the variable *j*. To achieve this result, SYTA needs to compute and record post dominance relationships at the basic-block level before symbolic execution.

At first, a user appoints variables as taint sources and SYTA calculates the immediate post-dominant basic block of the corresponding basic block containing the taint sources. Insert the pair <*i*, *ipdom_bb*> into the IFT list, where *i* stands for the tainted variable, and *ipdom_bb* means the immediate post-dominate basic block of the current basic block.

During path travelling, when a basic block is reached, SYTA compares it with all the *ipdom_bb*s in the control-flow based taint pairs in the IFT list in an attempt to find matches and then remove the matching pairs. After removing, if the IFT list is not empty, the *ipdom* of the current basic block will be calculated and the taint pairs are formed together with every variable *v* referenced in the current basic block. These pairs are added to the IFT list one by one. In other words, if the target variable *i* is marked as tainted, the variables in the current basic block will also be marked as tainted according to the control dependence relationship. No further operations will be performed if the IFT list is NULL and only the explicit information flow taint propagation goes on.

In a CFG, basic block *m* post-dominates (*ipdom*) *n* means all directed paths from *m* to the exit basic block contain *n*. If there is no node *o* such that *n pdom o* and *o pdom m*, we call *m* is immediate post-dominates *n*. Just like that in Figure 4, BB5*ipdom* BB2.

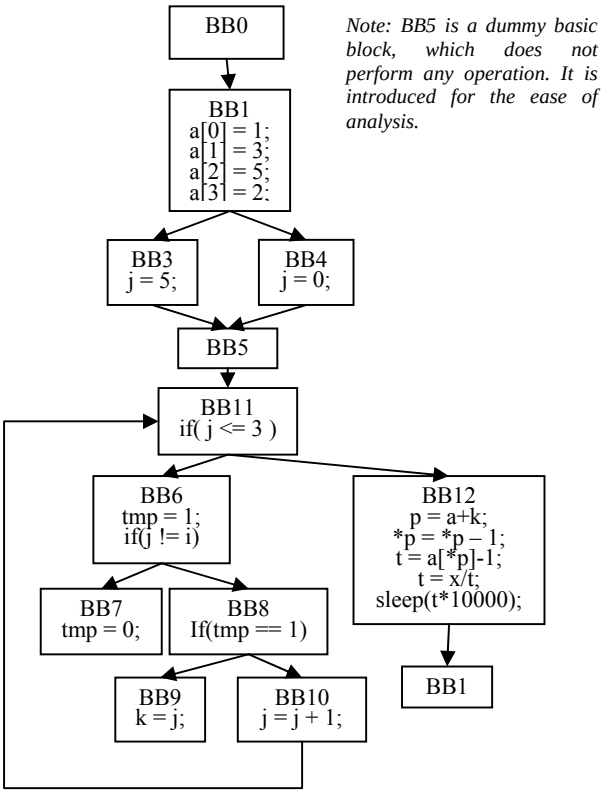
Take the program in Figure 4 (a) as an example again, whose CFG and post-dominance tree are shown in Figure 4 (b) and (c), respectively. We assume that variable *i* is chosen as a taint source. At first, the IFT list is initialized to be empty. When line 5 is executed, SYTA will identify the current statement as a condition statement. The corresponding *ipdom* is BB5, a pair <*i*, BB5> will be added into the IFT list. The symbolic execution forks here and finds both paths are feasible. The true branch would be executed first and line 6 is reached. At this time, the index of the current basic block is 3, and there are not matching pairs in the IFT list. The destination operand of the statement, the left-value *j*, would be added into IFT list together with its *ipdom* BB5 in the form of <*j*, BB5>. All these two pairs will be removed when line 11 is reached because BB5 matches either of them.

C. Explicit Information Flow Taint Propagation

Explicit information flow taint propagation is quite straightforward compared with the implicit one. Only direct data dependence, such as assignment operations, needs to be

```
1:  void main(void) {
2:     unsigned int i;
3:     unsigned int t;
4:     int a[4] = { 1, 3, 5, 1 };
5:     int *p;
6:     int tmp;
7:     int j;
8:     int k;
9:     int x = 100;
10:    scanf("%d",&i);
11:    if(i >= 4){
12:           j = 5;
13:    }
14:    else{
15:           j = 0;
16:    }
17:    for(j; j<4;j++)
18:    {
19:           tmp = 1;
20:           if( j != i){
21:                  tmp = 0;
22:           }
23:           if (tmp == 1){
24:                  k = j;
25:           }
26:    }
27:    p = a+k;
28:    *p = *p - 1;
29:    t = a[*p]-1;
30:    t = x / t;
31     sleep (t*10000);
32:  }
```

(a)



(b)

Figure 3. The source code and CFG of a sample program under testing

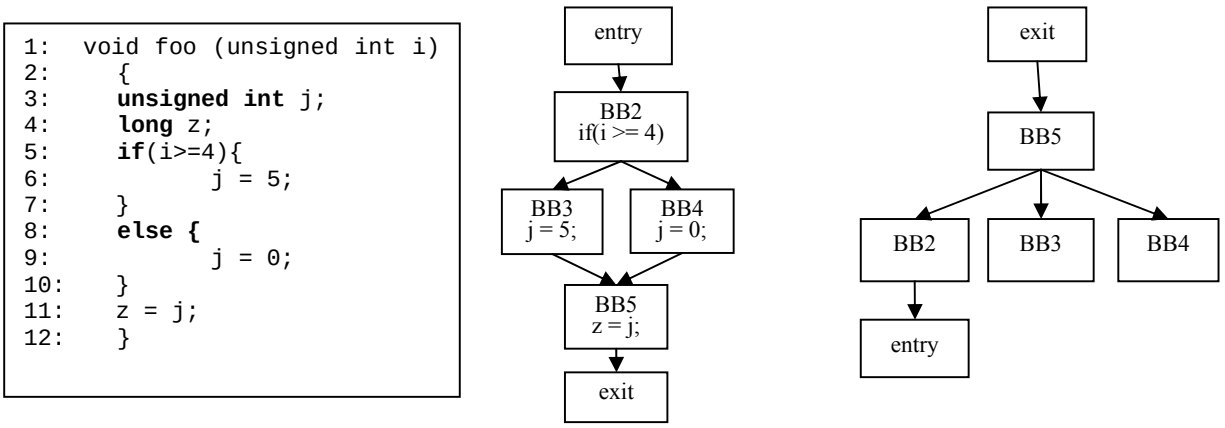


Figure 4. A fragment of the sample program in Figure 3 and its CFG, post-dominate tree

considered in taint propagation.

When an assignment statement is encountered, SYTA will check whether the operands on the right side of the statement are included in the EFT list. If the answer is positive, SYTA will insert the left operand into the EFT list.

In addition, when new pairs are added into the IFT list, the corresponding variables of the pairs should meanwhile be inserted into the EFT list too. This approach is adopted because the information flow among variables maybe proceeds alternately between the two forms. This is a generally ignored problem. Let's still take the case in Figure 4 as an example, in the program, there is no explicit information flow from variable *i*, exists only an implicit information flow from variable *i* to variable *j* caused by the if-then-else clause. But the information flow from *j* to variable *z* in line 11 is explicit. If the two kinds of information flows are processed separately, then in line 11, the variable *z* will not be tainted because the variable *j* is only tainted with implicit information flow. As a result, no taint markings will the variable *z* has, which leads to false negatives because the value of variable *z* is influenced by variable *i*.

D. Test Case Generation

In KLEE, the context of constraints solving only contains path conditions. In order to capture the implicit information flows, the indirect equivalence relationships between variables are also identified by SYTA and sent to the SMT solver as asserts. Take the program in Figure 3 as an example, there exists an implicit equivalence relationship between *k* and *j* (i.e., *k* == *j*) after executing line 24. When the branch condition is not satisfied in line 20, both the relationships *j* == *i* and *k* == *j* will hold after line 24. SYTA will record both equivalent variables pairs in the ER list rather than only one explicit pair (i.e., *j* == *i*).

When a security sink is encountered, two kinds of asserts will be sent to the SMT solver as the context. One is the path condition of the current path, the other is a Conjunctive Normal Form (CNF) formed with pairs in the ER list. As illustrated in the CFG in Figure 3(b), which is expressed in the intermediate language, when the execution path is (BB0 → BB1 → BB4 → BB5 → BB11 → BB6 → BB8 → BB9 → BB10 → BB11 → BB12), the current security sink is a reference to array *a*. At this time, the path condition is ($i \leqslant 3$ && $j \leqslant 3$ && $j == i$ && $tmp == 1$ && $j(1) \geqslant 3$); the assert drawn from the ER list is ($k == i$), *j(1)* is an alias of variable *j*. All these expressions are set to be asserts of the SMT solver. The query submitted is ($*p \geqslant 0$ && $*p \leqslant 3$). The counterexamples the SMT solver provides are ($i = 2$; $j = 2$; $*p = 4$). The test case is ($i = 2$).

Three kinds of bugs are considered in SYTA: (1) array bound overflow, (2) divide-by-zero, and (3) denial-of-service.

(1) If the index variable is marked as tainted in a reference to an array, a query is constructed as (*index* >= 0 && *index* <= *upperbound* – 1) and be sent to the SMT solver. Under certain contexts, there exists an array bound overflow if all the constraints are satisfied and the query is not.

(2) If the operator is a divisor, and the divisor *m* is tainted. Then the sink query (*m* != 0) is constructed and sent to the SMT solver together with all the asserts gathered till now. Divide-by-zero is found if all the constraints are satisfied and the query is not.

(3) When the function *sleep* is called, and its parameter is marked as tainted, then the query (*sleep* <= 10000) is constructed and sent to the SMT solver together with all the asserts. Then the DoS bug exists if all the constraints are satisfied and the query is not.

In a word, when SYTA encounters a security sink, it will gather all the path conditions preceding the current statement and asserts from ER list, the query will be sent to the SMT solver. If the query is unsatisfied, a test case is generated and reported with the bug name.

Based on the above discussion, as shown in Table I, three test cases are generated to detect bugs in the sample program in Figure 3. They can be used to trigger the array bound overflow, divide-by-zero and DoS bugs, respectively.

 ISBN: 978-1-61208-376-6

TABLE I. TEST CASES OF THE SAMPLE PROGRAM BY SYTA

Taint Sources	Tainted Variables	Vulnerability Type
$i = 2$;	$j = 2$; $*p = 4$;	array bound overflow
$i = 3$;	$j = 3$; $*p = 1$; $t = 0$;	divide-by-zero
$i = 1$;	$j = 1$; $*p = 2$; $t = 25$;	dinal-of-service

E. Implementation

As shown in Figure 2, we employ GCC 4.5.0 as the compiler front-end of SYTA. Source code will be parsed and convert to GIMPLE intermediate representation; its CFGs are also built by leveraging GCC. SYTA is implemented as a pass of GCC, analysis will be performed at the GIMPLE level. Finally, we choose the commonly used constraints solver STP [16] as the SMT solver in SYTA.

IV. EVALUATION

We illustrate two cases that show how SYTA can detect errors. In the program shown in Figure 5, the control dependence relationships are based on the *switch-case* structure. During analysis, we leverage the GIMPLE intermediate representation of GCC to process the *switch-case* structure. In GIMPLE, a *switch-case* will be regarded as a normal *if-else* structure. When the original taint source is variable *n*, a counterexample ($n = 245$) can be got and the

```
void foo(int n)

{
  Unsigned int y[256];
  Unsigned int x[256];

  for(int i=0; i<256; i++)
  {
    y[i] = (char)i;
  }

  for(int j=0; j<n; j++)
  {
    switch(y[j])
    {
      case 0:
             x[j] = 13;
             break;
      case 1:
             x[j] = 14;
             break;
      case 2:
             x[j] = 15;
             break;
      ……
      case 256:
             x[j] = 12;
             break;
    }
  }
  n = y[n]/x[n-1];
}
```

Figure 5. The first case study

```
void foo(int h)
{
     int a[5] = {1,2,3,4,5};
     int l = 10;
     int k = 0;
     if(h < 0){
             l = 0;
     }
     while(l != 0){
             if(l <= 5){
                     k++;
             }
             l--;
     }
     l = a[k];
}
```

Figure 6. The second case study

assignment statement ($n = y[n] / x[n-1];$) may trigger a *divide-by-zero* bug.

In the program shown in Figure 6, there is no explicit *else* branch in the if ($h < 0$) statement. In order to capture the taint propagation through the missing *else* branch, an assisting *else* branch is inserted into the intermediate representation, which includes a dummy statement ($l = l$). Using the dummy statement, a counterexample ($h = 2$) would be found as the test case for the array bound overflow bug at the last statement.

In this paper, we try to extend the test case generation technique to cover implicit information flows rather than only explicit information flows. In theory, it is impossible to track and analyze all forms of implicit information flows. Our study shows that some typical forms of implicit information flows can be effectively tracked to support test case generation. In this section, we employ two proof-of-principle samples to demonstrate the ability of SYTA to track typical forms of implicit information flows. We also use KLEE (with LLVM v2.7) to analyze the two samples and the program shown in Figure 3(a). Compared with SYTA, KLEE, as shown in Figure 7, only provides two test cases (i.e., $i = 1$ and $i = 2147483648$) for feasible execution paths of the program shown in Figure 3(a), but these cases can not trigger and report the array-bound-overflow and divide-by-zero vulnerabilities. Nevertheless, frankly

```
KLEE: output directory = "klee-out-0"
KLEE: done: total instructions = 68
KLEE: done: completed paths = 2
KLEE: done: generated tests = 2
$ ktest-tool --write-ints klee-last/test000001.ktest
  ktest file : 'klee-last/test000001.ktest'
  args       : ['test.o']
  num objects: 1
  object    0: name: 'i'
  object    0: size: 4
  object    0: data: 1

  $ ktest-tool --write-ints klee-last/test000002.ktest
  ...
  object    0: data: 2147483648
```

Figure 7. The KLEE analysis result for the program in Figure 3(a)

speaking, analyzing a large scale real-work system will require much more computing overhead.

V. RELATED WORK

Even though early attention and definition of implicit information flow dated back to 1970's, no effective solution has been found. Lots of newly-developed tools, like TaintScope [6], detour the implicit information flow problem and limit their applications only to explicit information flows. Some other work limits the processing of control dependence to predetermined forms, for example, Heng Yin et al. deal with the API function containing control dependence specially in their tool Panorama [11]; The system designed by Wei Xu et al. [12] process only two specific kinds of control flow; Dongseok Jang et al. [13] only process the branching but not the whole program leading to low coverage and false negatives.

Some dynamic analysis testing tools are more comprehensive, like Dytan [10] by James Clause, it can construct implicit information flow on the binary code but cannot get the control dependence information from indirect jump instructions. In DTA++ developed by Min Gyung Kang et al., information preserving implicit information flows are traced [14], but the simple dichotomy approach is too rough and may cause under-tainting problem.

VI. CONCLUSION AND FUTURE WORK

We presented a static analysis tool, SYTA, capable of automatically generating test cases using symbolic execution and taint analysis techniques. Using the control flow graph of the target program and user-appointed taint sources as inputs, SYTA follows execution paths to track the constraints on symbolic variables, and maintains two taints lists for explicit and implicit information flows respectively. The test cases will be generated from possible counterexamples in a constraint solving process. Compared with traditional static analysis tools, SYTA can track implicit information flows, generates test cases by control dependence analysis effectively.

At present, in tracking implicit information flows, SYTA cover only three kinds of sink points, concerning array bound overflow, divide-by-zero, and denial-of-service, respectively. By expending taint source points and sink points, it may cover other kinds of vulnerabilities related to taint data. For example, by regarding untrusted input interface functions as taint source points and function *memcpy* and the like as sink points, it can detect buffer overflow vulnerability led to by ineffective input validation.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments that helped improve the presentation of this paper. The work has been supported in part by the National Natural Science Foundation of China (61070192, 61170240, 61272493, 61100047), the Natural Science Foundation of Beijing (4122041), the National Science and Technology Major Project of China (2012ZX01039-004), and the National High Technology Research and Development Program of China (2012AA012903).

REFERENCES

[1] D. Bird and C. Munoz, "Automatic Generation of Random Self-Checking Test Cases," IBM Systems Journal, Vol. 22, No. 3, 1983, pp. 229-245.

[2] Protos, Web page: http://www.ee.oulu.fi/research/ouspg/protos/, [retrieved: August, 2014].

[3] J. Offutt and J. Hayes, "A Semantic Model of Program Faults," in Proceedings of ISSTA'96 (International Symposium on Software Testing and Analysis), San Diego, January 1996, pp. 195-200.

[4] E. J. Schwartz, T. Avgerinos, and D. Brumley, "All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask)," in Proceedings of the IEEE Symposium on Security and Privacy, May 2010, pp. 317-331.

[5] D. E. Denning and P. J. Denning, "Certification of programs for secure information flow," Comm. of the ACM, vol. 20, no. 7, July 1977, pp. 504-513.

[6] T. Wang, T. Wei, G. Gu, and W. Zou, "TaintScope: A checksum-aware directed fuzzing tool for automatic software vulnerability detection," in Proceedings of the 31st IEEE Symposium on Security and Privacy, Oakland, California, USA, May 2010, pp. 497-512.

[7] L. Cavallaro, P. Saxena, and R. Sekar, Anti-taint-analysis: Practical evasion techniques against information flow based malware defense. Technical report, Stony Brook University, 2007.

[8] C. Cadar, D. Dunbar, and D. Engler, "Klee: Unassisted and automatic generation of high-coverage tests for complex systems programs," in Proceedings of the USENIX Symposium on Operating System Design and Implementation, 2008, pp. 209-224.

[9] C. Cadar, V. Ganesh, P. Pawlowski, D. Dill, and D. Engler, "EXE: A system for automatically generating inputs of death using symbolic execution," in Proceedings of the ACM Conference on Computer and Communications Security, October 2006, pp.322-335.

[10] J. Clause, W. Li, and A. Orso, "Dytan: a generic dynamic taint analysis framework," in International Symposium on Software Testing and Analysis, 2007, pp. 196-206.

[11] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda, "Panorama: Capturing system-wide information flow for malware detection and analysis," in Proceedings of the ACM Conference on Computer and Communications Security, October 2007, pp. 116-127.

[12] W. Xu, E. Bhatkar, and R. Sekar, "Taint-enhanced policy enforcement: A practical approach to defeat a wide range of attacks," in Proceedings of the USENIX Security Symposium, 2006, pp. 121-136.

[13] D. Jang, R. Jhala, S. Lerner, and H. Shacham, "An empirical study of privacy-violating information flows in JavaScript web applications," in Proceedings of the ACM Conference on Computer and Communications Security, 2010, pp. 270-283.

[14] J. Clause, W. Li, and A. Orso, "Dytan: a generic dynamic taint analysis framework," in International Symposium on Software Testing and Analysis, 2007, pp. 196-206.

[15] M. G. Kang, S. McCamant, P. Poosankam, and D. Song, "DTA++: Dynamic Taint Analysis with Targeted Control-Flow Propagation," in Proceedings of the Network and Distributed System Security Symposium, February 2011, pp. 205-219.

[16] V. Ganesh and D. L. Dill, "A decision procedure for bit-vectors and arrays," in Proceedings of the 19th International Conference on Computer Aided Verification, 2007, pp. 519-531.

 ISBN: 978-1-61208-376-6

Implementation Issues in the Construction of Standard and Non-Standard Cryptography on Android Devices

Alexandre Melo Braga, Eduardo Moraes de Morais
Centro de Pesquisa e Desenvolvimento em Telecomunicações (Fundação CPqD)
Campinas, São Paulo, Brazil
{ambraga,emorais}@cpqd.com.br

***Abstract*—This paper describes both the design decisions and implementation issues concerning the construction of a cryptographic library for Android Devices. Four aspects of the implementation were discussed in this paper: selection of cryptographic primitives, architecture of components, performance evaluation, and the implementation of non-standard cryptographic algorithms. The motivation behind the special attention given to the selection of alternative cryptographic algorithms was the recently revealed weakness found in international encryption standards, which may be intentionally included by foreign intelligence agencies.**

Keywords-Cryptography; Surveillance; Security; Android.

I. INTRODUCTION

Currently, the proliferation of smartphones and tablets and the advent of cloud computing are changing the way software is being developed and distributed. Additionally, the use in software systems of cryptographic techniques is increasing as well.

This paper discusses the construction of a cryptographic library for Android devices. The paper focuses on design decisions as well as on implementation issues of both standard and non-standard algorithms. This work contributes to the state of the practice by discussing the technical aspects and challenges of cryptographic implementations. This work is part of an effort to build security technologies into an integrated framework for mobile device security [2]. The evaluation of several cryptographic libraries on Android devices was reported in a previous work [1], showing that there is a lack of sophisticated cryptographic primitives, such as elliptic curves and bilinear pairings. Moreover, the majority of assessed schemes implements only standard algorithms, and, as far as authors know, there is no practical design that concerns alternative, non-standard cryptography.

The motivation behind the special attention given to the selection of alternative cryptographic algorithms was the recently revealed weakness, which may be intentionally included by foreign intelligence agencies in international encryption standards [16][26]. This fact alone raises doubt on all standardized algorithms, which are internationally adopted. In this context, a need arose to treat what has been called "alternative" or "non-standard" cryptography in opposition to standardized cryptographic schemes. The final intent was strengthening the implementation of advanced cryptography and fostering their use. Non-standard cryptography provides advanced mathematical concepts, such as bilinear pairings and elliptic curves, which are not fully standardized by foreign organizations, and suffer constant improvements.

The remaining parts of the text are organized as follows. Section II offers background on the subject of cryptographic implementation on Java and Android. Section III details the implementation aspects. Section IV presents a performance evaluation and comparison with other libraries. Section V concludes this text.

II. BACKGROUND AND RELATED WORK

This section briefly describes topics of interest: the Java Cryptographic Architecture (JCA) as a framework for pluggable cryptography; the Java Virtual Machine (JVM) with its Garbage Collector (GC) and Just-in-Time (JiT) compilation; and The Dalvik Virtual Machine (DVM) for Android.

A. JCA

The JVM is the runtime software ultimately responsible for the execution of Java programs. In order to be interpreted by JVM, Java programs are translated to bytecodes, an intermediary representation that is neither source code nor executable. The JCA [17] is a software framework for use and development of cryptographic primitives in the Java platform. The JCA defines, among other facilities, Application Program Interfaces (APIs) for digital signatures and secure hash functions [17]. On the other hand, APIs for encryption, key establishment and message authentication codes (MACs) are defined in the Java Cryptography Extension (JCE) [19]. Since version 1.4, the JCE was incorporated by JCA, being treated in practice as a single framework, named JCA or JCE [20].

The benefit of using a software framework, such as JCA, is to take advantage of good design decisions, reusing the whole architecture. The API keeps the same general behavior regardless of specific implementations. The addition of new algorithms is facilitated by the use of a standard API [20].

B. GC on JVM

An architectural feature of the JVM has great influence in the general performance of applications: the GC [35][37]. Applications have different requirements of GC. For some applications, pauses during garbage collection may be tolerable, or simply obscured by network latencies, in such a way that throughput is an important metric of performance.

 ISBN: 978-1-61208-376-6

However, in others, even short pauses may negatively affect the user experience.

One of the most advertised advantages of JVM is that it shields the developer from the complexity of memory allocation and garbage collection. However, once garbage collection is a major bottleneck, it is worth understanding some aspects of its implementation.

The JVM incorporates a number of different GC algorithms that are combined using generational collection. While simple GC examines every live object in the heap, generational collection explores other hypothesis in order to minimize the work required to reclaim unused objects. The hypothesis supporting generation GC is corroborated by observed behavior of applications, where most objects survive for only a short period of time. Some objects can be reclaimed soon by memory management, because they have died shortly after being allocated. For example, iterator objects are often alive for the duration of a single loop.

On the other hand, some objects do live longer. For instance, there are typically some objects allocated at initialization that live until the program terminates. Between these two extremes are objects that live for the duration of some intermediate computation. For example, external loop variables live longer than inner loop variables. Efficient GC is made possible by focusing on the fact that a majority of objects die young.

Collections are clearly identified in diagrams, as shown in Figure 1. The figure shows the time consumed by the first 500 of 10.000 executions of pure-Java implementation of the AES encryption algorithm.

C. JiT Compilation

Other import consideration on performance of Java programs is the JiT Compilation [10][35]. Historically, Java bytecode used to be fully interpreted by the JVM and presented serious performance issues. Now a days, the technology known as HotSpot uses JiT Compilation not only to compile Java programs, but also to optimize them, while they execute. The result of JiTC is an application that has portions of its bytecode compiled and optimized for the targeted hardware, while other portions are still interpreted.

It is interesting to notice that JVM has to execute the code before to learn how to optimize it. The very first moments of an application show a relatively poor performance, since the bytecode is been interpreted, analyzed for optimizations, and compiled at the same time.

After this short period, the overall performance of the application improves and the execution tends to stabilize at an acceptable level of performance. Once again, the period of optimization and compilation is clearly identified in diagrams, as is shown in Figure 1.

A feature referred by Oracle as JVM Ergonomics was introduced in Java 5.0 with the goal of providing good performance with little or no tuning of configuration options for JVM. Instead of using fixed defaults, JVM ergonomics automatically selects GC, heap size, and runtime compiler at JVM startup. The result of ergonomics is that the choice of a GC does not matter to most applications. That is, most applications can perform well under the choices made by JVM, even in the presence of pauses of modest frequency and duration.

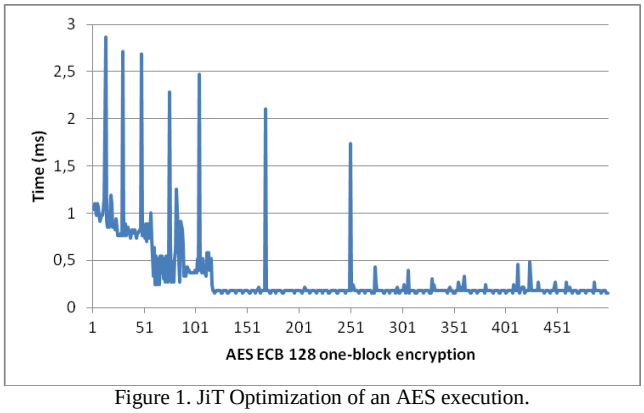


Figure 1. JiT Optimization of an AES execution.

Unfortunately, there is a potential negative side to security in the massive use of JiT Compilation. Security controls put in place into source code, in order to avoid side-channels, can be cut off by JiT optimizations. JiTC is not able to capture programmer's intent that is not explicitly expressed by Java's constructs. That is exactly the case of constant time computations needed to avoid timing attacks. Security-ware optimizations should be able to preserve security decisions and not undo protections, when transforming source code for cryptographic implementations to machine code. Hence, to achieve higher security against this kind of attacks, it is not recommended to use JiTC technology, what constitutes a trade-off between security and performance.

D. DVM

The DVM [7] is the virtual hardware that executes Java bytecode in Android. DVM is quite different from the traditional JVM, so that software developers have to be aware of those differences, and performance measurements over a platform independent implementation have to be taken in both environments.

Compared to JVM, DVM is a relatively young implementation and did not suffered extensive evaluation. In fact, the first independent evaluation of DVM was just recently published [13]. There are three major differences between DVM and JVM. First of all, DVM is a register-based machine, while JVM is stack-based. Second, DVM applies trace-based JiTC, while JVM uses method-based JiTC. Finally, former DVM implementations use mark-and-sweep GC, while current JVM uses generation GC.

Also, results from that DVM evaluation [13] suggest that current implementations of DVM are slower than current implementations of JVM. Concerning cryptographic requirements, a remarkable difference between these two environments is that the source of entropy in DVM is significantly different from the one found on JVM.

III. DESCRIPTION OF THE IMPLEMENTATION

In order to facilitate the portability of the cryptographic library for mobile devices, in particular for the Android

 ISBN: 978-1-61208-376-6

platform, the implementation was performed according to standard cryptographic API for Java, the JCA, its name conventions, and design principles [14][17]-[20].

Once JCA was defined as the architectural framework, the next design decision was to choose the algorithms minimally necessary to a workable cryptographic library. The current version of this implementation is illustrated by Figure 2 and presents the cryptographic algorithms and protocols described in the following paragraphs. The figure shows that frameworks, components, services and applications are all on top of JCA API. The Cryptographic Service Provider (CSP) is in the middle, along with BouncyCastle and Oracle providers. Arithmetic libraries are at the bottom.

Figure 2 shows the CSP divided in two distinct cryptographic libraries. The left side shows only standardized algorithms and comprises a conventional cryptographic library. The right side features only non-standard cryptography and is an alternative library. The following subsections describe these two libraries.

A. *Standard Cryptography*

This subsection details the implementation choices for the standard cryptographic library. The motivations behind this implementation were all characteristics of standardized algorithms: interoperability, documentation, and testability. The standard cryptography is packaged as a pure-Java library according to the JCA specifications.

The programming language chosen for implementation of this cryptographic library was Java. The block cipher is the AES algorithm, which was implemented along with thee of operation: ECB, and CBC [27], as well as the GCM mode for authenticated encryption [28]. PKCS#5 [3] is the simplest padding mechanism and was chosen for compatibility with other CSPs. As GCM mode for authenticated encryption only uses AES encryption, the optimization of encryption received more attention than AES decryption. Implementation aspects of AES and other cryptographic algorithms can be found on the literature [15][24][34], in particular [29].

The asymmetric algorithm is the RSA-PSS that is a Probabilistic Signature Scheme constructed over the RSA signature algorithm. PSS is supposed to be more secure than ordinary RSA [23][34]. Asymmetric encryption is provided by the RSA-OAEP [23][34].

Two cryptographically secure hashes were implemented, SHA-1 [22] and MD5. It is well known by now that MD5 is considered broken and is not to be used in serious applications, it is present for ease of implementation. In current version, there is no intended use for these two hashes. Their primary use will be as the underling hash function in MACs, digital signatures and PRNGs. The Message Authentication Codes chosen were the HMAC [25] with SHA-1 as the underling hash function, and the GMAC [28], which can be directly derived from GCM mode. SHA-2 family of secure hashes supplies the need for direct use of single hashes.

The need for a key agreement was fulfilled by the implementation of Station-to-Station (STS) protocol, which

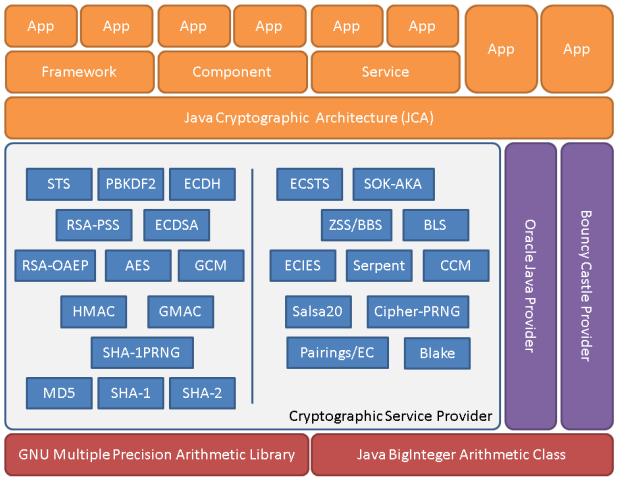


Figure 2. Cryptographic Service Provider Architecture.

is based upon Authenticated Diffie-Hellman [38], and provides mutual key authentication and confirmation [4][39].

Finally, the mechanism for Password-based Encryption (PBE) is based on the Password-Based Key Derivation Function 2 (PBKDF2) [3], and provides a simple and secure way to store keys in encrypted form. In PBE, a key-encryption-key is derived from a password.

B. *Non-standard Cryptography*

This subsection details the implementation choices for the alternative cryptographic library. The non-standard cryptography is a dynamic library written in C and accessible to Java programs through a Java Native Interface (JNI) connector, which acts as a bridge to a JCA adapter.

By the time of writing, this alternative library was under the final steps of its construction. The most advanced cryptographic protocols currently implemented are based upon a reference implementation [5] and are listed below.

a) ECDH [8]. The key agreement protocol ECDH is a variation of the Diffie-Hellman protocol using elliptic curves as the underlying algebraic structure;
b) ECDSA [21]. This is a DSA-based digital signature using elliptic curves. ECSS [8] is a variation of ECDSA that does not require the computation of inverses in the underlying finite field, obtaining a signature algorithm with better performance;
c) SOK [8]. This protocol is a key agreement for Identity-Based Encryption (IBE). Sometimes, it is called SOKAKA for SOK Authenticated Key Agreement;
d) BLS [6]. A short digital signature scheme in which given a message m, it is computed S = H (m), where S is a point on an elliptic curve and H() is a secure hash;
e) ZSS [11]. Similar to the previous case, it is a more efficient short signature, because it utilizes fixed-point multiplication on an elliptic curve rather arbitrary point;
f) Blake [32]. Cryptographic hash function submitted to the worldwide contest for selecting the new SHA-3 standard and was ranked among the five finalists;
g) ECIES [8]. This is an asymmetric encryption algorithm over elliptic curves. This algorithm is non-deterministic

 ISBN: 978-1-61208-376-6

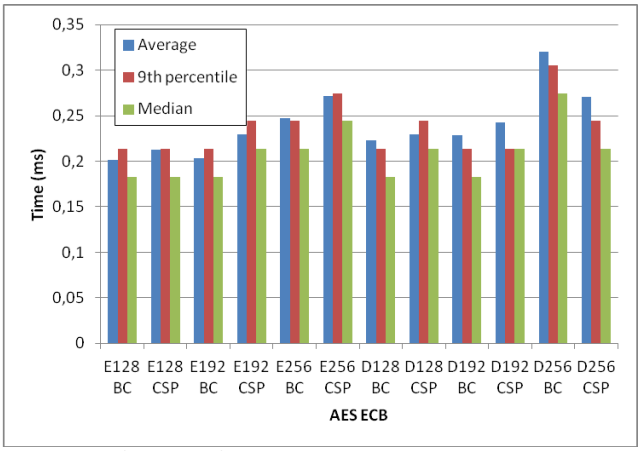


Figure 4. Performance of AES in pure-Java - average, 9th percentile, and median of 10.000 iterations.

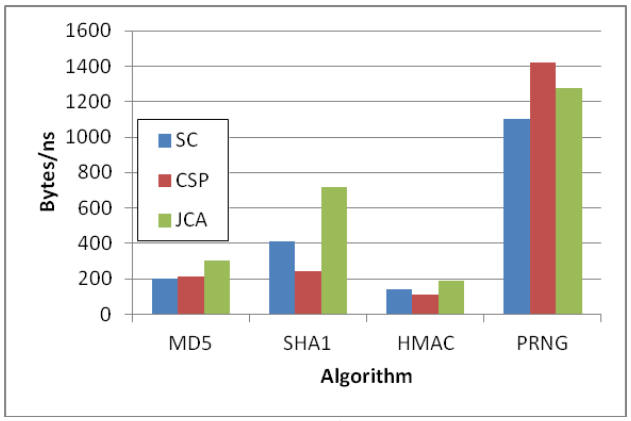


Figure 3. Throughput of implementations.

and can be used as a substitute of the RSA-OAEP, with the benefit of shorter cryptographic keys;

h) ECSTS [8]. Variation of STS protocol using elliptic curves and ECDH as a replacement for ADH;

i) Salsa20 [9]. This is a family of 256-bit stream ciphers submitted to the ECRYPT Project (eSTREAM);

j) Serpent [31]. A 128-bit block cipher designed to be a candidate to contest that chose the AES. Serpent did not win, but it was the second finalist and enjoys good reputation in the cryptographic community.

C. *Security decisions for non-standard cryptography*

Among the characteristics that were considered in the choice of alternative cryptographic primitives, side channels protection was a prevailing factor and had distinguished role in the design of the library. For instance, schemes with known issues were avoided, while primitives that were constructed to resist against such attacks are currently being regarded for inclusion in the architecture. Furthermore, constant-time programming techniques, like for example in table accessing operations for AES, are being surveyed in order to became part of the implementation.

Concerning mathematical security of non-standard cryptography, the implementation offers alternatives for 256-bit security for both symmetric and asymmetric encryption. For instance, Serpent-256 corresponds to AES-256 block cipher, while the same security level is achieved in asymmetric world using elliptic curves over 521-bit finite fields, what can only be possible in standard cryptography using 15360-bit RSA key size. Thus, in higher security levels, non-standard primitives performance is significantly improved in relation to standard algorithms, but an extensive analysis of this scenario, with concrete timing comparisons, is left as future work.

A final remark about the use of non-standard cryptography is that working with advanced cryptographic techniques that have not been sufficiently analyzed by the scientific community has its own challenges and risks. There are occasions when the design of a non-standard cryptographic library has to be conservative in order to preserve security.

For instance, a recent improvement in mathematics [12][30] had eliminated an entire line of research in theoretical cryptography. Such advancement affected elliptic curve cryptography using a special kind of binary curves called supersingular curves, but had no effect on the bilinear pairings over primes fields or encryption on ordinary (common) binary curves. Thus, these two technologies remain cryptographically secure. Unfortunately, the compromised curves were in use and had to be eliminated from the cryptographic library.

As pairings on prime fields can still be securely used in cryptographic applications, the implementation was adapted to that new restricted context. Additionally, ordinary elliptic curves may still be used for cryptographic purposes, considering they are not supersingular curves, and the implementation had to adapt to that fact too.

IV. PERFORMANCE EVALUATION

Performance evaluation of Java programs, either in standard JVM or DVM/Android, is a stimulating task due to many sources of interference that can affect measurements. As discussed in previous sections, GC and JiTC have great influence over the performance of Java programs. The intent of performance evaluations presented in this section is to provide and describe a realistic means to compare cryptography implementations in Java.

Two approaches of measurement have been used for the evaluation of cryptographic functions implemented in pure-Java programs. The first one was the measurement of elapsed time for single cryptographic functions processing a single block of data. This approach suffers from the interference of GC and JiTC. The JiTC interference can be eliminated by discarding all the measurements collected before code optimization. The GC interference cannot be completely eliminated, though.

 ISBN: 978-1-61208-376-6

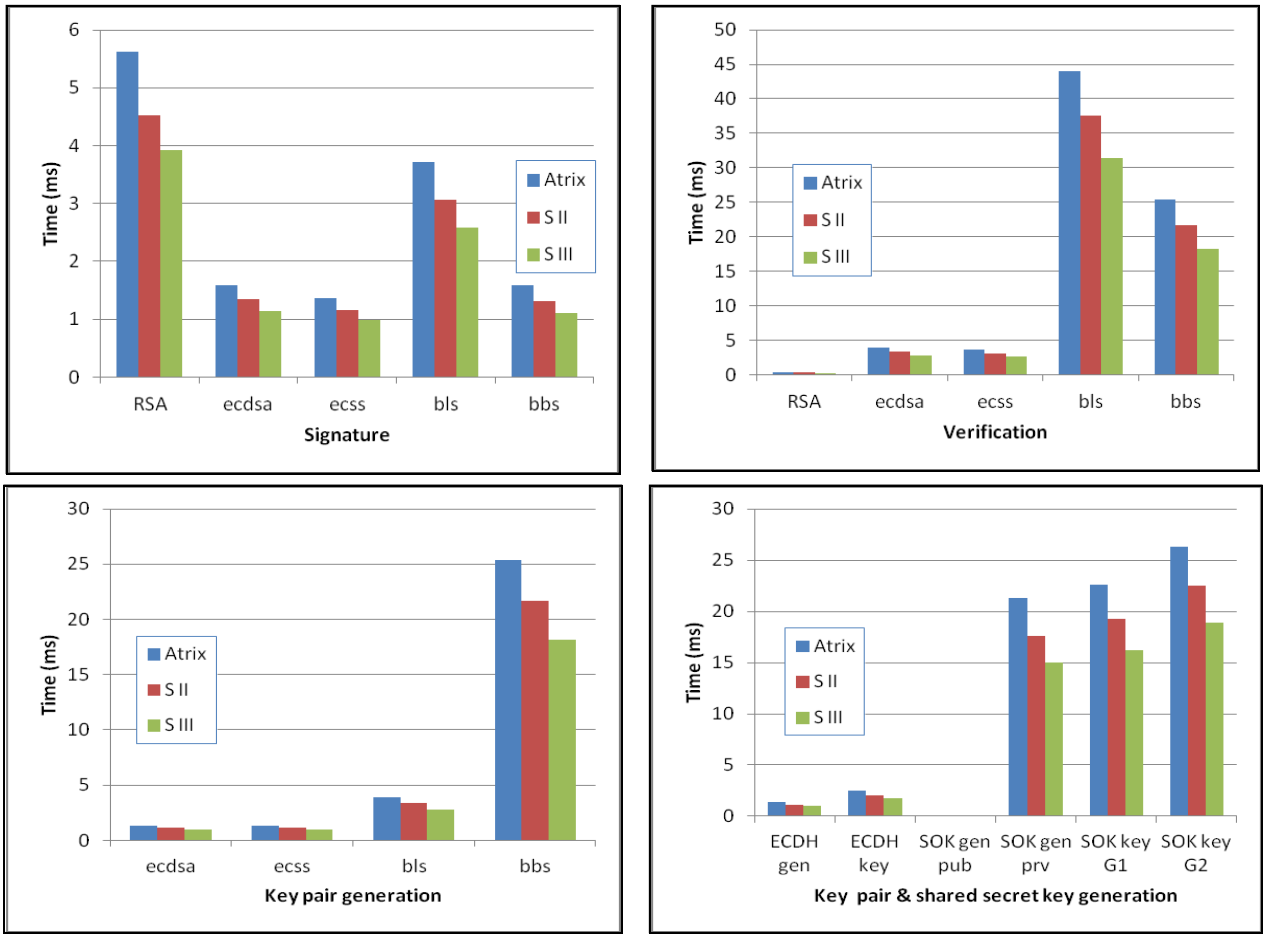


Figure 5. Performance evaluation of non-standard cryptography. Digital signatures: signature generation (top-left) and signature verification (top-right). Key Agreement: key pair generation (bottom-left), secret-key generation (bottom-right).

Figure 4 exemplifies the first approach and shows the comparative performance of AES encryption, in ECB mode, of a single block of data for two Java CSPs: This text CSP and BouncyCastle (BC) [36]. The measurements were taken on an LG Nexus 4 with 16GB of internal storage, 2 GB of RAM, 1.5 GHz Quad-core processor, and Android 4.3. The procedure consisted of processing a single block of data in a loop of 10.000 iterations. AES were setup to three key sizes (128, 192 and 256) in both encryption and decryption.

In order to inhibit the negative influence of GC and JiTC, three metrics were taken: the average of all iterations, the 9th percentile and the median. None of them resulted in a perfect metric. However, these measures do offer a realistic comparison of CSP and BC. They show similar performance.

The second approach for performance evaluation has to consider that final users of mobile devices will not tuning their Java VMs with obscure configuration options in order to achieve maximum performance. On the contrary, almost certainly they will use default configurations, with minor changes on device's settings. Thus, the responsiveness of an application tends to be more relevant to final users than the performance of single operations.

The second approach of measurement takes into account the responsiveness of cryptographic services and considers the velocity with which a huge amount of data can be processed, despite the interferences of GC and JiTC. The amount of work performed per unit of time is called the throughput of the cryptographic implementation.

Figure 3 shows the throughput of four cryptographic services implemented by CSP compared to BC and JCE: MD5, SHA-1, HMAC and SHA1PRGN. The measurements were taken on a smartphone of type Samsung Galaxy S III (Quad-core 1.4 GHz Cortex-A9 processor, 1GB of RAM, and Android 4.1). The procedure consisted of processing an input file of 20 MB, in a loop of 10 iterations. All cryptographic algorithms were setup with a 128-bit key. BouncyCastle has a deployment for Android, called SpongeCastle (SC) [33]. It is interesting to observe that the three CSPs are quite similar in performance.

The previous paragraphs suggest that the pure-Java package of CSP, with standard cryptography only, is quite competitive in performance when compared to other CSP and its use might not be considered a bottleneck to applications.

 ISBN: 978-1-61208-376-6

Unfortunately, the case for non-standard cryptography is not that simple, despite been implemented in C and not been subjected to GC and JiTC influences. Non-standard cryptography usually has no standard specifications or safe reference implementations. Neither it is in broad use by other cryptographic libraries. Because of that, comparisons among implementations of the same algorithm are barely possible. On the other hand, it is feasible to compare alternative and standard cryptography, considering the same type of service.

For the non-standard cryptography implementations, performance measurements were taken in three smartphones: (i) Motorola Atrix with processor of 1 GHz, 1 GB of RAM and 16GB of storage; (ii) Samsung Galaxy S II with processor of 1.2 GHz dual-core ARM Cortex-A9, 1 GB of RAM and 16GB of storage; and (iii) Samsung Galaxy S III with processor of 1.4 GHz quad-core Cortex-A9, 1 GB of RAM, and 16 GB of storage.

Figure 5 shows two types of services: digital signatures at the top and key agreement (KA) at the bottom. The bar chart at top-left quadrant shows generation of digital signatures for five algorithms: RSA, ECDSA, ECSS, BLS and ZSS (BBS). Traditionally, RSA is the slowest one. Elliptic curve cryptography, as in ECDSA and ECSS, is faster. Short signatures, such as BLS and ZSS (BBS), are not as fast as EC.

Bar chart at top-right quadrant shows verification of digital signatures for five algorithms: RSA, ECDSA, ECSS, BLS and ZSS (BBS). Traditionally, RSA verification is the fastest one. Elliptic curve cryptography, as in ECDSA and ECSS, is not that fast. Short signatures, such as BLS and ZSS (BBS), are terribly slow, due to complex arithmetic involved in bilinear pairings computations. The bottom-left quadrant contains a bar chart showing key pair generation for ECDSA, ECSS, BLS, and ZSS (BBS). Again, performance is slow for BLS and ZSS (BBS) due to complex arithmetic involved in bilinear pairings.

Bar chart in bottom-right quadrant shows operations for two KA schemes: ECDH and SOK. ECDH is quite fast in generating parameters (both public and private), as well as in generating the shared secret. But, pairings based KA schemes are relatively slow in both operations.

V. CONCLUDING REMARKS

This paper discussed implementation issues on the construction of a cryptographic library for Android smartphones. The library actually consists of both standard and non-standard cryptographic algorithms. Performance measurements were taken in order to compare CSP with other cryptographic providers. Despite all difficulties for obtain realistic data, experiments have shown that standard CSP can be competitive to other implementations. On the other hand, non-standard cryptography has shown low performance that can possibly inhibit its use in real time applications. However, their value consists in offering secure alternatives to possibly compromised standards. Future work will focus on correctness, security (particularly in the context of side channel attacks) and performance optimization. Correctness of implementation in the absence of formal verification is a primary concern and should be taken seriously, particularly for non-standard cryptography.

Finally, regarding recent global surveillance disclosures, non-standard cryptographic primitives can be faced as part of the usual trade-offs that directs the design of cryptographically secure applications.

ACKNOWLEDGMENT

The authors acknowledge the financial support given to this work, under the project "Security Technologies for Mobile Environments – TSAM", granted by the Fund for Technological Development of Telecommunications – FUNTTEL – of the Brazilian Ministry of Communications, through Agreement Nr. 01.11. 0028.00 with the Financier of Studies and Projects - FINEP / MCTI.

REFERENCES

[1] A. Braga and E. Nascimento, Portability evaluation of cryptographic libraries on android smartphones. In Proceedings of the 4th international conference on Cyberspace Safety and Security (CSS'12), Yang Xiang, Javier Lopez, C.-C. Jay Kuo, and Wanlei Zhou (Eds.). Springer-Verlag, Berlin, Heidelberg, 2012, pp. 459-469.

[2] A. Braga, Integrated Technologies for Communication Security on Mobile Devices. In MOBILITY, The Third International Conference on Mobile Services, Resources, and Users, 2013, pp. 47-51.

[3] B. Kaliski, RFC 2898. PKCS #5: Password-Based Cryptography Specification Version 2.0. Available in: http://tools.ietf.org/html/rfc2898.

[4] B. O'Higgins and W. Diffie and L. Strawczynski, R. do Hoog, Encryption and ISDN - A Natural Fit, 1987 International Switching Symposium (ISS87), 1987.

[5] D. Aranha and C. Gouvêa, RELIC, RELIC is an Efficient LIbrary for Cryptography, Available in: http://code.google.com/p/relic-toolkit.

[6] D. Boneh and B. Lynn and H. Shacham, Short signatures from the Weil pairing. J. Cryptology, Extended abstract in Proceedings of Asiacrypt 2001, Sept. 2004, 17(4): pp. 297–319.

[7] D. Bornstain, Dalvik, VM Internals. Available in: http://sites.google.com/site/io/dalvik-vm-internals.

[8] D. Hankerson and S. Vanstone and A. Menezes, Guide to elliptic curve cryptography, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[9] D. J. Bernstein, The Salsa20 family of stream ciphers. Available in: http://cr.yp.to/papers.html#salsafamily.

[10] Ergonomics in the 5.0 Java Virtual Machine. Available in: http://www.oracle.com/technetwork/java/ergo5-140223.html

[11] F. Zhang and R. Safavi-Nainia and W. Susilo, An Efficient Signature Scheme from Bilinear Pairings and Its Applications., in F. Bao and R. H. Deng and J. Zhou, ed., Public Key Cryptography, Springer, 2004, pp. 277-290.

[12] G. Anthes, "French team invents faster code-breaking algorithm", Communications of the ACM, v. 57,n. 1, January 2014, pp. 21-23.

[13] H. Oh and B. Kim and H. Choi and S. Moon, Evaluation of Android Dalvik virtual machine. In Proceedings of the 10th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES '12), ACM, New York, NY, USA, 2012, pp. 115-124.

[14] How to Implement a Provider in the Java Cryptography Architecture. Available in: http://docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/HowToImplAProvider.html.

[15] J. Bos and D. Osvik and D. Stefan, Fast Implementations of AES on Various Platforms, 2009. Available in http://eprint.iacr.org/2009/501.pdf.

 ISBN: 978-1-61208-376-6

[16] J. Menn, Experts report potential software "back doors" in U.S. standards. Available in: http://www.reuters.com/article /2014/07/15/usa-nsa-software-idUSL2N0PP2BM20140715?irpc=932.

[17] Java Cryptography Architecture Oracle Providers Documentation for Java Platform Standard Edition 7. Available in: http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html.

[18] Java Cryptography Architecture Standard Algorithm Name Documentation for Java Platform Standard Edition 7. Available in: http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html.

[19] Java Cryptography Extension Unlimited Strength Jurisdiction Policy Files 7 Download. Available in: http://www.oracle.com/technetwork/pt/java/javase/downloads/jce-7-download-432124.html.

[20] Java™ Cryptography Architecture (JCA) Reference Guide. Available in: http://docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/CryptoSpec.html.

[21] NIST FIPS PUB 186-2, Digital Signature Standard (DSS). Available in: http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf.

[22] NIST FIPS-PUB-180-4. Secure Hash Standard (SHS). Available in: http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf, March 2012.

[23] NIST FIPS-PUB-186. Digital Signature Standard (DSS). Available in: http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf.

[24] NIST FIPS-PUB-197. Announcing the ADVANCED ENCRYPTION STANDARD (AES). Federal Information Processing Standards Publication 197 November 26, 2001.

[25] NIST FIPS-PUB-198. The Keyed-Hash Message Authentication Code (HMAC). Available in: http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf.

[26] NIST Removes Cryptography Algorithm from Random Number Generator Recommendations. Available in: http://www.nist.gov/itl/csd/sp800-90-042114.cfm.

[27] NIST SP 800-38A. Recommendation for Block Cipher Modes of Operation. 2001. Available in: http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf.

[28] NIST SP 800-38D. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. 2007. Available in: http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf.

[29] Paulo Barreto's AES Public Domain Implementation in Java. Available in: www.larc.usp.br/~pbarreto/JAES.zip.

[30] R. Barbulescu, P. Gaudrey, A. Joux,, and E. Thomé, "A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic", June 2013, preprint available at http://eprint.iacr.org/2013/400.pdf.

[31] SERPENT, A Candidate Block Cipher for the Advanced Encryption Standard. Available in: www.cl.cam.ac.uk/~rja14/serpent.html.

[32] SHA-3 proposal BLAKE. Available in: https://131002.net/blake.

[33] SpongyCastle, Spongy Castle: Repackage of Bouncy Castle for Android, Bouncy Castle Project. Available in: http://rtyley.github.com/spongycastle/, 2012.

[34] T. St. Denis and S. Johnson, Cryptography for Developers. Syngress, 2006.

[35] The Java HotSpot Performance Engine Architecture. Available in: www.oracle.com/technetwork/java/whitepaper-135217.html.

[36] The Legion of the Bouncy Castle. Legion of the Bouncy Castle Java cryptography APIs. Available in: www.bouncycastle.org/java.html.

[37] Tuning Garbage Collection with the 5.0 Java Virtual Machine. Available in: http://www.oracle.com/technetwork/ java/gc-tuning-5-138395.html.

[38] W. Diffie and M. Hellman, New Directions in Cryptography, IEEE Trans. on Inf. Theory, vol. 22, no. 6, Nov. 1976, pp. 644-654.

[39] W. Diffie and P. C. van Oorschot, M. J. Wiener, Authentication and Authenticated Key Exchanges, Designs, Codes and Cryptography (Kluwer Academic Publishers) 1992, 2 (2): pp. 107–125.

 ISBN: 978-1-61208-376-6

Threshold Proxy Signature Based on Position

Qingshui Xue
Dept. of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, China
xue-qsh@cs.sjtu.edu.cn

Fengying Li
School of Continuous Education
Shanghai Jiao Tong University
Shanghai, China
fyli@sjtu.edu.cn
zfcao@cs.sjtu.edu.cn

Zhenfu Cao
Dept. of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, China

Abstract—**Position-based cryptography has attracted lots of researchers' attention. In the mobile Internet, there are many position-based security applications. For the first time, one new conception, threshold proxy signature based on positions is proposed. Based on one secure positioning protocol, one model of threshold proxy signature based on positions is proposed. In the model, positioning protocols are bound to threshold proxy signature tightly, not loosely. Further, one position-based threshold proxy signature scheme is designed, its correctness is proved, and its security is analyzed. As far as we know, it is the first threshold proxy signature scheme based on positions.**

Keywords-position; threshold proxy signature; proxy signature; UC security; model; scheme.

I. INTRODUCTION

In the setting of mobile Internet, position services and position-binding security applications become one key requirement, especially the latter. Position services include position inquiring, secure positioning and so forth. Position inquiring consists of inquiring your own position and positioning of other entities. The technology of inquiring your own position has Global Positioning System (GPS) and other satellite service system. The technology of positioning of other entities has radar and so on [2]-[6]. As we all know, the positioning of other entities is more challenging one. Position-binding security applications such as position-based encryption and position-based signature and authentication are increasingly necessary for us. For example, when one mobile user sends messages to one specific position, which is one either physical address or logical address (such as Internet Protocol address), it is desirable for us that only the user who is at that address or has been at that address can receive and decrypt messages encrypted. Even if other mobile users at that position receive messages, but they can't decrypt them. Or the specified receiver at that position due to some reasons temporarily leaves his/her position, it will not be able to receive or decrypt messages any more. In addition, if the specified receiver at that place moves to another place, and he/she hopes he/she can receive messages at the new place. Take one application about position-based signature and authentication as an example. One mobile or fixed user signs messages at one place and sends them to another mobile user. The receiver can receive the signed message and verify whether or not received messaged is indeed signed at the place by the signer. Even if the signer moves to another address, it will not affect the receiving and verification of signed messages.

Currently, the research on position-based cryptography focuses on secure positioning about which some work had been proposed [1]. These positioning protocols are based on one-dimension, two-dimension or three-dimension spaces, including traditional wireless network settings [1], as well as quantum setting [7]-[9]. It seems to us that position-based cryptography should integrate secure positioning with cryptographic primitives. If only or too much concentrating on positioning protocols, perhaps we will be far away from position-based cryptography. In other words, nowadays positioning is bound loosely with related security applications, not tightly, as results in the slow progress of position-based cryptography and applications. Relying on the thoughts, in the paper, our main contributions are as follows.

(1)One model of threshold proxy signature based on positions is proposed. Position-based threshold proxy signature is one kind of threshold proxy signature, but a novel one. The definition is given and its model is constructed. In the meantime, its security properties are defined.

(2)To realize the kind of threshold proxy signature, one secure-positioning-protocol based threshold proxy signature scheme is proposed and its security is analyzed as well.

The rest of the paper is organized as follows. In Section 2, the function of positioning and one secure positioning protocol are introduced. In Section 3, the model and definition of threshold proxy signature based on positions are constructed. One position-based threshold proxy signature scheme is designed in Section 4. The correctness of the scheme is proved in Section 5. The security of the proposed scheme will be analyzed in Section 6. Finally, the conclusion is given.

 ISBN: 978-1-61208-376-6

II. POSITION PROTOCOLS

In this section, the function of positioning protocols and one secure positioning protocol are introduced.

A. Function of Positioning Protocols

The goal of positioning protocol is to check whether one position claimer is really at the position claimed by it. Generally speaking, in the positioning protocol, there are at least two participants including position claimers and verifiers, where the verifiers may be regarded as position infrastructure. According to destination of the positioning, there are two kinds of positioning protocol, i.e., your own position positioning protocol and others' position positioning protocol. As of now, lots of work on your own position positioning protocol have been done [2]-[6]. Nevertheless, research on others' positions positioning protocol is far less and there are still many open questions to resolve. In our model and scheme, we will make full use of the two varieties of positioning protocol.

B. One Secure Positioning Protocol

Here, one others' positions secure positioning protocol is introduced. Compared with your own position positioning protocol, others' positions positioning protocol is more complex.

In this section, N. Chandran et al.'s secure positioning protocol in 3-dimensions is reviewed [1], which can be used in mobile Internet.

In the protocol, 4 verifiers denoted by $V_1, V_2, ..., V_4$, which can output string X_i, are used. The prover claims his/her position, which is enclosed in the tetrahedron defined by the 4 verifiers. Let $t_1, ..., t_4$ be the time taken for radio waves to arrive at the point P from verifier $V_1, V_2, ..., V_4$ respectively. When we say that $V_1, V_2, ..., V_4$ broadcast messages such that they "meet" at P, we mean that they broadcast the messages at time $T - t_1, T - t_2, T - t_3$ and $T - t_4$ respectively so that at time T all the messages are at position P in space. The protocol uses a pseudorandom generator namely an ε-secure $PRG: \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^m$. They select the parameters such that $\varepsilon + 2^{-m}$ is negligible in the security parameters. X_i denotes a string chosen randomly from a reverse block entropy source. The protocol is given as follows:

Step 1. $V_1, ..., V_3$ and V_4 pick keys $K_1, ..., K_3$ and K_4 selected randomly from $\{0,1\}^m$ and broadcast them through their private channel.

Step 2. For the purpose of enabling the device at P to calculate K_i for $1 \le i \le 4$, the verifiers do as follows. V_1 broadcasts key K_1 at time $T - t_1$. V_2 broadcasts X_1 at time $T - t_2$ and meanwhile broadcasts $K_2^{'} = PRG(X_1, K_1) \oplus K_2$. Similarly, at time $T - t_3$, V_3 broadcasts $(X_2, K_3^{'} = PRG(X_2, K_2) \oplus K_3)$, and V_4 broadcasts $(X_3, K_4^{'} = PRG(X_3, K_3) \oplus K_4)$ at time $T - t_4$.

Step 3. At time T, the prover at position P calculates messages $K_{i+1}^{'} = PRG(X_i, K_i) \oplus K_{i+1}$ for $1 \le i \le 3$. Then it sends K_4 to all verifiers.

Step 4. All verifiers check that the string K_4 is received at time $(T + t_i)$ and that it equals K_4 that they pre-picked. If the verifications hold, the position claim of the prover is accepted and it is supposed to be indeed at position P. Otherwise, the position claim is invalid.

III. THE MODEL OF POSITION-BASED THRESHOLD PROXY SIGNATURE

The model, definition and security properties are proposed in this section.

A. The model

In the model, there are four kinds of participants including the original signer (OS), the proxy signer group (PSG), which consists of n proxy signers $\{PS_1, PS_2, ..., PS_n\}$, the verifier (V) and position infrastructure (PI). OS takes responsibility of confirmation of position of his/her own and at one position delegates his/her signing power to the proxy signer group to sign messages at these proxy signers' positions on behalf of OS. $t(t \le n)$ or more proxy signers cooperate to sign one message at their individual positions after their positions are confirmed by PI and are the same as the ones in the proxy signing delegation warrant, whereas, less than t proxy signers can't. V checks that the proxy signature is generated by the actual proxy signers at their individual positions on behalf of OS. PI, which is one trusted third party, is used to provide position services for the related parties. The model is illustrated in Figure 1.

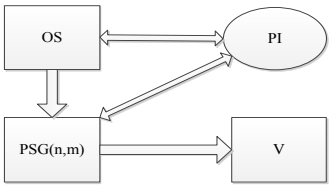


Figure 1. Model of position-based threshold proxy signature.

B. Definition

Position-based threshold proxy signature.

Simply speaking, the kind of proxy signature combines proxy signature, threshold proxy signature and positioning protocols as one single scheme. It is mainly composed of three modules of threshold proxy signing power delegation, threshold proxy signing and threshold proxy signature verifying. In the module of threshold proxy signing power delegation, OS first sends one request to PI for the purpose of delegating signing power to PSG. Then PI runs one positioning protocol to confirm OS and the proxy signers' positions. If their positions are valid, PI sends acknowledge to OS and the proxy signers. After that, PI produces proxy signing key packages for individual proxy signers and sends them to each proxy signer. OS produces proxy delegation warrant to all proxy signers. In the module of threshold proxy

 ISBN: 978-1-61208-376-6

signing, each proxy signer who wants to actually attend to sign has to first check that his/her position is at the designated position, which is specified in the proxy delegation warrant. If it holds, each actual proxy signer can use his/her proxy signing key package to sign the message for only once and sends individual proxy signature to one clerk who collects all individual signatures and generates final threshold proxy signature. In the module of threshold proxy signature verifying, V uses OS and all proxy signers' identities and positions to check the validity of threshold proxy signatures based on positions.

Remark 1. During the module of threshold proxy signing power delegation, if OS and all of proxy signers don't run positioning protocols with PI to confirm their own positions, OS is unable to delegate his/her signing power to the proxy signer group. Moreover, if neither OS nor each proxy signer can confirm its position with PI, OS can't fulfill his delegation of signing power. In the module of threshold proxy signing, if each of proxy signers doesn't perform positioning protocols to check the validity of his/her position, he/she is not able to generate individual proxy signature by individual proxy signature key package. That's to say, before the proxy signer group wants to sign one message on behalf of OS, each member has to confirm its position. Even if each proxy signer passes individual position's confirmation, he/she can sign one message for only once. During the module of threshold signature verifying, it is unnecessary for the verifier to confirm OS and all proxy signers' positions.

In the model, it will be seen that we regard the three modules as three primitives. Therefore, in our model, the positioning protocol is bound tightly with the delegation of signing power and threshold proxy signature generation, instead loosely.

Thus, in the model, the positioning-based threshold proxy signature is composed of four primitives:Initialization, PropTProxyDelegate, PropTProxySign and PropTProxyVerify.

Initialization. PI takes as input secure parameter 1^k and outputs system master key mk and public parameter pp, in the meantime, the system distributes user identity ID_i for user i.

PropTProxyDelegate. When OS wants to delegate his/her signing power to the proxy signer group, OS first sends his/her requests to PI. After PI gets OS's request, PI checks the validity of positions $Pos_{OS}, Pos_{PS_1}, Pos_{PS_2}, ..., Pos_{PS_n}$ of OS and all proxy signers by running positioning protocol with OS and PS. If OS and all proxy signers' positions are valid, PI sends the acknowledgment to OS. According to the acknowledgment from PI, OS generates delegation warrant dw and sends it to each proxy signer PS_i $(i = 1, 2, ..., n)$. At the same time, PI produces proxy signing key package $pskp_i$ for PS_i $(i = 1, 2, ..., n)$. dw contains OS and all proxy signers' identities and positions, n, t (threshold value), message types to sign, expiry date and so forth. $pskp_i$ encapsulates positioning protocol, proxy signing key, the i^{th} proxy signer's identity ID_{PS_i} and position Pos_{PS_i}, signing algorithm, etc.

PropTProxySign. Before the proxy signer group wants to sign the message m on behalf of OS, actual proxy signer PS_i $(i = 1, 2, k. t \leq k \leq$ (here, assume that PS_i $(i = 1, 2, ..., k)$ are the proxy signers participating in the signing, denoted by aps)first executes individual proxy signing key package $pskp_i$ to run positioning protocol to confirm the validity of his/her position Pos_{PS_i} with PI. If his/her current position Pos_{PS_i} is identical to the one in the delegation warrant dw, he/she is able to use proxy signing key package $pskp_i$ to sign the message m for only once and sends corresponding individual proxy signature (m, s_i, dw, pp) to the Clerk, who collects and verifies individual signatures, and generates final threshold proxy signature. The Clerk checks the validity of individual signature s_i by using the identity ID_{PS_i} and position Pos_{PS_i} of PS_i and corresponding verification algorithm. If the number of the actual proxy signers k is equal to or more than t, and less than or equal to n, and k individual signatures are valid, the Clerk will generate the final threshold proxy signature (m, s, dw, asp, pp) and send it to V. Here, simply denote s by $s = \prod_{i=1}^{k} s_i$.

PropTProxyVerify. After receiving the threshold proxy signature (m, s, dw, asp, pp) from the proxy signer group, V takes as input the identities $ID_{OS}, ID_{PS_1}, ID_{PS_2}, ..., ID_{PS_n}$, positions $Pos_{OS}, Pos_{PS_1}, Pos_{PS_2}, ..., Pos_{PS_n}$, asp and pp to check whether or not s is the threshold proxy signature on the message m by using corresponding threshold proxy signature verification algorithm. If it holds, V can be sure that the message m was signed by actual proxy signers PS_i $(i = 1, 2, ..., k)$ at position Pos_{PS_i} $(i = 1, 2, ..., k)$ on behalf of OS who delegated his/her signing power to the proxy signer group at the position Pos_{OS}.

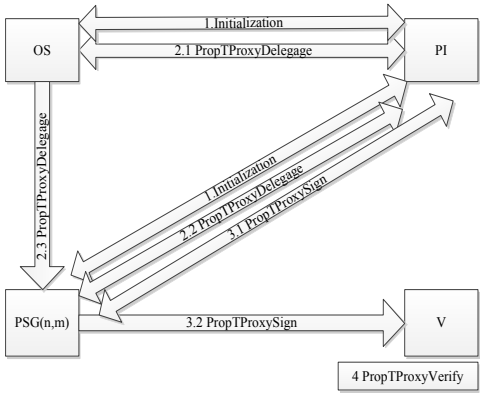


Figure 2. Position-based threshold proxy signature.

The model is illustrated in Figure 2.

C. Security Properties of Position-Based Threshold Proxy Signature

Besides security properties of threshold proxy signature, this kind of threshold proxy signature has the security properties as follows.

(1) Positioning protocol binding. In the module of PropTProxyDelegate, without confirming of positions of OS and all proxy signers by running positioning protocol with PI, OS is unable to fulfill his/her delegation of signing power. In addition, the proxy signing key package of each proxy signer produced by PI is tightly bound with positioning protocol, as means that if all of proxy signers want to use proxy signing key packages, each of them has to run positioning protocol with PI. In the module of PropTProxySign, if the proxy signer group needs to sign one message on behalf of OS, in order to get the proxy signing key (implicitly), each of proxy signers has to make use of individual proxy signing key package to run positioning protocol with PI. If each of proxy signers is indeed at the position specified in the delegation warrant dw, he/she will be able to obtain (implicitly) the proxy signing key to sign one message for once only. Actually, each of proxy signers can't get real individual proxy signing key, which is encapsulated in the proxy signing key package.

Remark 2. In the model, for each time, if the proxy signers want to cooperate to sign messages on behalf of OS, each of them has to run positioning protocol with PI to confirm the validity of individual positions. One maybe thinks we should make use of one-time digital signing algorithm or one-time signing private key. In fact, in the model, using one-time signing key is optional. On one hand, if the model uses fixed signed private key encapsulated in the proxy signing key package, each of proxy signers can't use it at will and can sign one message for only once. If the proxy signer group wants to sign another message, all of them still need to communicate with PI once again. In the sense, each of proxy signers actually has no the knowledge of its own proxy signing private key. On the other hand, if one-time signing private keys are used in the model, it is reasonable. That is to say, for each time, each proxy signing key package will release one random proxy signing key. In addition, because position-based applications are closely related with position instant authentication or confirmation, we think that position-based cryptography should be deeply researched with respect to online cryptography, which focuses on instant cryptographic algorithms and security processing. Of course, in the eyes of ours, it is one open question as well.

IV. One Position-Based Threshold Proxy Signature Scheme

In this section, one position-based threshold proxy signature scheme, in which there exist one original signer and n proxy signers, is proposed. The scheme mainly includes four kinds of participants:the original signer (still denoted as OS), the proxy signer group (PSG) $PSG = \{PS_1, PS_2, ..., PS_n\}$, the verifier (V) and PI. PI will make use of the secure positioning protocol mentioned in Section 2.2 to provide services of position for the original signer and n proxy signers. In addition, PI will be regarded as the trusted third party and system authority. The scheme is composed of four primitives:Initialization, PropTProxyDelegate, PropTProxySign and PropTProxyVerify. As primitives, it means that they either fully run or do nothing. The four primitives are detailed as follows.

A. Initialization

PI takes as input secure parameter 1^k and outputs system master key mk and public parameter pp, at the same time, PI distributes user identity ID_i for user i. Here, rewrite the primitive as Initialization (k, mk, pp).

B. PropTProxyDelegation

Step 1. The original signer sends his/her requests $(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1}, ID_{PS_2}, Pos_{PS_2}, ..., ID_{PS_n}, Pos_{PS_n}, req_{deleg})$ of delegating signing power to the proxy signer group to PI.

Step 2. After PI gets OS's request, PI checks the validity of the positions $Pos_{OS}, Pos_{PS_1}, Pos_{PS_2}, ..., Pos_{PS_n}$ of OS and all proxy signers by running positioning protocol with OS and each proxy signer.

Step 3. If OS and all proxy signers' positions $Pos_{OS}, Pos_{PS_1}, Pos_{PS_2}, ..., Pos_{PS_n}$ are valid, as means that OS is indeed at the position Pos_{OS} and Pos_{PS_i} is one valid position of PS_i $(i = 1, 2, ..., n)$, PI sends the acknowledgement $(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1}, ID_{PS_2}, Pos_{PS_2}, ..., ID_{PS_n}, Pos_{PS_n}, ack_{deleg})$ to OS; otherwise PI sends $(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1}, ID_{PS_2}, Pos_{PS_2}, ..., ID_{PS_n}, Pos_{PS_n}, rej_{deleg})$ to OS.

Step 4. If OS receives $(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1}, ID_{PS_2}, Pos_{PS_2}, ..., ID_{PS_n}, Pos_{PS_n}, ack_{deleg})$ from PI, he/she generates delegation warrant $dw(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1}, ID_{PS_2}, Pos_{PS_2}, ..., ID_{PS_n}, Pos_{PS_n}, Sign_{OS}(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1}, ID_{PS_2}, Pos_{PS_2}, ..., ID_{PS_n}, Pos_{PS_n}))$ where $Sign_{OS}(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1}, ID_{PS_2}, Pos_{PS_2}, ..., ID_{PS_n}, Pos_{PS_n})$ is the digital signature on $(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1}, ID_{PS_2}, Pos_{PS_2}, ..., ID_{PS_n}, Pos_{PS_n})$ generated by OS, and sends it to each proxy signer.

Step 5. PI produces proxy signing key package $pskp_i$ for each proxy signer and sends it to PS_i $(i = 1, 2, ..., n)$. $pskp_i$ encapsulates positioning protocol, proxy signing key, signing algorithm, the identity and position of PS_i, etc. Anyone wanting to use proxy signing key and signing algorithm in $pskp_i$ has to run the proxy signing key package $pskp_i$.

 ISBN: 978-1-61208-376-6

C. PropTProxySign

Step 1. When the proxy signer group wants to sign the message m on behalf of OS, each actual proxy signer PS_i $(i=1,2,...,k, t \le k \le n)$ (here, assume that PS_i $(i=1,2,...,k)$ are the actual proxy signers, denoted by asp)runs proxy signing key package $pskp_i$ for executing positioning protocol to confirm the validity of his/her position Pos_{PS_i} with PI.

Step 2. If $PS_i's$ current position Pos_{PS_i} is identical to the one in the delegation warrant dw, proxy signing key package $pskp_i$ prompts PS_i to input the message m to $pskp_i$. Thus proxy signing key package $pskp_i$ produces the individual proxy signature s_i and send it to the Clerk; if PS's current position Pos_{PS_i} is not identical to the one in the delegation warrant dw, PS_i is unable to perform the function of proxy signing and stops $(i=1,2,...,k)$.

Step 3. After the Clerk receives the individual proxy signature s_i , he/she checks s_i is the individual proxy signature by using verification algorithm, the identity and position of PS_i $(i=1,2,...,k)$.

Step 4. If all $s_i's$ verification hold, the Clerk generates the final threshold proxy signature s by processing all individual proxy signatures s_i $(i=1,2,...,k)$. Here, simply denote s by $s=\prod_{i=1}^{k} s_i$.

Step 5. The clerk sends (m,s,dw,aps,pp) to the proxy signature verifier V.

D. PropTProxyVerify

Step 1. After receiving the threshold proxy signature (m,s,dw,aps,pp) , V takes as input the identities $ID_{OS}, ID_{PS_1}, ID_{PS_2},..., ID_{PS_n}$, positions $Pos_{OS}, Pos_{PS_1}, Pos_{PS_2}$, $..., Pos_{PS_n}$ from dw and pp to check that the proxy delegation warrant dw is valid. If it is valid, the scheme continues, or V fails to stop.

Step 2. V takes as input the identities $ID_{OS}, ID_{PS_1}, ID_{PS_2},..., ID_{PS_n}$, positions $Pos_{OS}, Pos_{PS_1}, Pos_{PS_2}$, $..., Pos_{PS_n}$ from dw, aps and pp to check whether or not s is the threshold proxy signature on the message m , and $t \le k \le n$. If it holds, V can be sure that the message m was signed by actual proxy signers at individual position Pos_{PS_i} $(i=1,2,...,k)$ on behalf of OS who delegated his/her signing power to the all proxy signers at the position Pos_{PS_i} $(i=1,2,...,n)$, and all proxy signers.

V. CORRECTION OF THE ABOVE SCHEME

In fact, the proof of correctness of above scheme is simple. The following theorem about it is given.

Theorem 1:If the scheme accurately and sequentially runs according to the primitives above, the verifier V can confirm that the threshold proxy signature is generated by the actually proxy signers asp at individual position Pos_{PS_i} $(i=1,2,...,k)$ on behalf of the original signer OS who at the position Pos_{OS} delegates his/her signing power to the group of proxy signers, and all proxy signers.

Proof.

In the primitive of PropTProxyDelegation, PI checks the validity of positions $Pos_{OS}, Pos_{PS_1}, Pos_{PS_2},..., Pos_{PS_n}$ of OS and all proxy signers by running positioning protocol with OS and each proxy signer. If all of positions are valid, as means that OS is actually at the position Pos_{OS} and Pos_{PS_i} is one valid position of PS_i $(i=1,2,...,n)$, PI sends the acknowledgement $(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1}, ID_{PS_2}, Pos_{PS_2},$ $..., ID_{PS_n}, Pos_{PS_n}, ack_{deleg})$ to OS. Then OS can generates delegation warrant $dw(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1}, ID_{PS_2},$ $Pos_{PS_2},..., ID_{PS_n}, Pos_{PS_n}, Sign_{OS}(ID_{OS}, Pos_{OS}, ID_{PS_1}, Pos_{PS_1},$ $ID_{PS_2}, Pos_{PS_2},..., ID_{PS_n}, Pos_{PS_n}))$ and sends it to each proxy signer. At the same time, OS produces proxy signing key package $pskp_i$ for each proxy signer and sends it to $PS_i's$ $(i=1,2,...,n)$. In the primitive of PropTProxySign, when the proxy signer group wants to sign the message m on behalf of OS, the actual proxy signers PS_i $(i=1,2,...,k)$ runs proxy signing key package $pskp_i$ for executing positioning protocol to confirm the validity of his/her position Pos_{PS_i} with PI. If $PS_i's$ current position Pos_{PS_i} is identical to the one in the delegation warrant dw , proxy signing key package $pskp_i$ prompts PS_i to input the message m to $pskp_i$. Thus proxy signing key package $pskp_i$ produces the individual proxy signature s_i and sends it to the Clerk. After the Clerk receives the individual proxy signatures s_i , he/she checks s_i is the individual proxy signature by using verification algorithms, the identities and positions of PS_i $(i=1,2,...,k)$. If all $s_i's$ verification hold, the Clerk generates the final threshold proxy signature s by processing all individual proxy signatures s_i $(i=1,2,...,k)$. Finally, the Clerk sends (m,s,dw,asp,pp) to the proxy signature verifier V. In the primitive of PropTProxyVerify, V takes as input the identities $ID_{OS}, ID_{PS_1}, ID_{PS_2},..., ID_{PS_n}$, positions $Pos_{OS}, Pos_{PS_1}, Pos_{PS_2}$, $..., Pos_{PS_n}$ from dw and pp to check that the proxy delegation

warrant dw is valid. Next, V takes as input the identities $ID_{OS}, ID_{PS_1}, ID_{PS_2}, ..., ID_{PS_n}$, positions $Pos_{OS}, Pos_{PS_1}, Pos_{PS_2}, ..., Pos_{PS_n}$ from dw, asp and pp to check whether or not s is the threshold proxy signature on the message m. If it holds, V can be sure that the message m was signed by the actual proxy signers at individual position Pos_{PS_i} $(i = 1, 2, ..., k)$ on behalf of OS who delegated his/her signing power to all proxy signers at the position Pos_{PS_i} $(i = 1, 2, ..., k)$, and all proxy signers. Thus, it is proved.

□

VI. SECURITY ANALYSIS OF THE PROPOSED SCHEME

In the proposed scheme, three sorts of technology, i.e., secure positioning protocol including others' positions positioning protocol and your own position positioning, proxy signature and threshold proxy signature, are used. That means, the security of the proposed scheme depends on the security of used three kinds of technology. Because the proposed scheme or the model is one component framework, it is proper that its security is analyzed by the Universal Composition (UC) framework [10]. That is, by constructing the components of proxy signature model, digital signature model and positioning protocol model, and attack modeling from internal attackers, external attackers and conspiracy attackers, the security analysis of the above scheme can be made. The internal attackers are from the original signer and proxy signers; the external attackers can be any software systems or entities; the conspiracy attackers mainly are among proxy signers, partially between both the original signer and some malicious proxy signers. By adding these models into the idealistic environment in UC framework, if the security of the scheme in the idealistic environment can be proved secure, its security in the real environment can be proved as well. Its security analysis will be deeply done in the further study.

VII. CONCLUSION AND FUTURE WORK

In the paper, according to security requirements of the mobile Internet, one model of position-based threshold proxy signature is constructed. Its definition, security properties and construction are given. As far as we know, it is the first model of combining positioning protocols, proxy signature and threshold proxy signature. In the meantime, one position-based threshold proxy signature scheme is proposed and analyze its security. We will further improve relevant models and schemes. It is believed by us that the research on positioning-protocol-based cryptographic models or schemes will become one focus in the setting of the mobile Internet.

ACKNOWLEDGMENT

I would like to thank so many anonymous reviewers for their advices of modification and improvements. In addition, this paper is supported by NSFC under Grant No. 61170227, Ministry of Education Fund under Grant No. 14YJA880033, and Shanghai Projects under Grant No. 2013BTQ001, XZ201301 and 2013001.

REFERENCES

[1] N. Chandran, V. Goyal, R. Moriarty, and R. Ostrovsky, "Position Based Cryptography," CRYPTO 2009, Aug. 2009, pp. 391-407, doi: 10.1007/978-3-642-03356-8_23.

[2] S.M. Bilal, C.J. Bernardos, and C. Guerrero, "Position-based routing in vehicular networks: A survey," Journal of Network and Computer Applications, vol. 36, Feb. 2013, pp. 685-697, doi: doi: 10.1016/j.jnca.2012.12.023.

[3] D. Singelee and B. Preneel, "Location verification using secure distance bounding protocols," IEEE Conference on Mobile Adhoc and Sensor Systems Conference, Nov. 2005, pp. -840 doi: 10.1109/MAHSS.2005.1542879.

[4] A. Fonseca and T. Vazão, "Applicability of position-based routing for VANET in highways and urban environment," Journal of Network and Computer Applications, vol. 36, Mar. 2013, pp. 961-973, doi: 10.1016/j.jnca.2012.03.009.

[5] S. Capkun and J. P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," IEEE INFOCOM, Mar. 2005, pp. 1917-1928, doi: 10.1109/INFCOM.2005.1498470.

[6] S. Capkun, M. Cagalj, and M. Srivastava, "Secure localization with hidden and mobile base stations," IEEE INFOCOM, Apr. 2006, pp. 1-10, doi: 10.1109/INFOCOM.2006.302.

[7] H. Buhrman et. al., "Position-Based Quantum Cryptography: Impossibility and Constructions," CRYPTO 2011, Aug. 2011, pp. 429-446, doi: 10.1007/978-3-642-22792-9_24.

[8] H. Buhrman et. al., "Position-Based Quantum Cryptography: Impossibility and Constructions," SIAM J. Comput., vol. 43, Jan. 2014, pp. 150–178, doi: 10.1137/130913687.

[9] T.Y. Wang and Z.L. Wei, "One-time proxy signature based on quantum cryptography," Quantum Information Processing, vol. 11, Feb. 2012, pp. 455-463, doi: 10.1007/s11128-011-0258-6.

[10] R. Canetti, "Universally composable security:a new paradigm for cryptographic protocols,"., 2001. Proceedings. 42nd IEEE Symposium on Foundations of Computer Science, Oct. 2001, pp. 136-145, doi: 10.1109/SFCS.2001.959888.

 ISBN: 978-1-61208-376-6

Linearity Measures for Multivariate Public Key Cryptography

Simona Samardjiska
Department of Telematics, NTNU
Trondheim, Norway
FCSE, "Ss Cyril and Methodius" University
Skopje, Republic of Macedonia
simonas@item.ntnu.no

Danilo Gligoroski
Department of Telematics, NTNU
Trondheim, Norway
danilog@item.ntnu.no

***Abstract*—We propose a new general framework for the security of Multivariate Quadratic (MQ) public key schemes with respect to attacks that exploit the existence of linear subspaces. We adopt linearity measures that have been used traditionally to estimate the security of symmetric cryptographic primitives, namely, the nonlinearity measure for vectorial functions introduced by Nyberg, and the *(s, t)*–linearity measure introduced recently by Boura and Canteaut. We redefine some properties of MQ cryptosystems in terms of these known symmetric cryptography notions, and show that our new framework is a compact generalization of several known attacks in MQ cryptography against single field schemes. We use the framework to explain various pitfalls regarding the successfulness of these attacks. Finally, we argue that linearity can be used as a solid measure for the susceptibility of MQ schemes to these attacks, and also as a necessary tool for prudent design practice in MQ cryptography.**

Keywords–Strong (s, t)–linearity; (s, t)–linearity; MinRank; good keys; separation keys.

I. INTRODUCTION

In the past two decades, as a result of the advancement in quantum algorithms, the crypto community showed increasing interest in algorithms that would be potentially secure in the post quantum world. One of the possible alternatives are Multivariate Quadratic ($\mathcal{MQ}$) public key cryptosystems based on the NP-hard problem of solving quadratic polynomial systems of equations over finite fields.

Many different $\mathcal{MQ}$ schemes emerged over the years, most of which fall into two main categories - single field schemes, including UOV (Unbalanced Oil and Vinegar) [1], Rainbow [2], TTM (Tame Transformation Method) [3], STS (Stepwise Triangular System) [4], MQQ-SIG (Multivariate Quadratic Quasigroups - Signature scheme) [5], TTS (Tame Transformation Signatures) [6], EnTTS (Enhanced TTS) [7] and mixed field schemes including C* [8], SFLASH [9], HFE (Hidden Field Equation) [10], MultiHFE [11][12], QUARTZ [13]. Unfortunately, most of them have been successfully cryptanalysed [4][14][15][16][17]. Three major types of attacks have proven devastating for $\mathcal{MQ}$ cryptosystems:

i. MinRank attacks – based on the problem of finding a low rank linear combination of matrices, known as MinRank [18]. Although NP-hard, the instances of MinRank arising from $\mathcal{MQ}$ schemes are often easy, and provide a powerful tool against single field schemes [4][14].
ii. Equivalent Keys attacks – based on finding an equivalent key for the respective scheme. The concept was introduced by Wolf and Preneel [19], and later further developed by Thomae and Wolf [16] to the generalization of good keys. The attacks on TTM [14], STS [4][16], HFE and MultiHFE [15][17] can all be seen from this perspective.
iii. Differential attacks – based on specific invariants of the differential of a given public key, such as the dimension of the kernel, or some special symmetry. It was introduced by Fouque *et al.* in [20] to break the perturbed version of the C^* scheme PMI [21], and later also used in [22][23][24][25].

Interestingly, the history of $\mathcal{MQ}$ cryptography has witnessed cases where, despite the attempt to inoculate a scheme against some attack, the enhanced variant has fallen victim to the same type of attacks. Probably, the most famous example is the SFLASH [9] signature scheme, that was build using the minus modifier on the already broken C^* [26], and selected by the NESSIE European Consortium [27] as one of the three recommended public key signature schemes. It was later broken by Dubois *et al.* in [24][25] using a similar differential technique as in the original attack on C^*. Another example is the case of Enhanced STS [28], which was designed to be resistant to rank attacks, that broke its predecessor STS. Even the authors themselves soon realized that this was not the case, and the enhanced variant is vulnerable to a HighRank attack.

Such examples indicate that the traditional "break and patch" practice in $\mathcal{MQ}$ cryptography should be replaced by a universal security framework. Indeed, in the last few years, several proposals have emerged that try to accomplish this [29][30][31]. Notably, the last two particularly concentrate on the properties of the differential of the used functions, a well known cryptanalytic technique from symmetric cryptography. We will show here that another well known measure from symmetric cryptography, namely linearity, is fundamental for the understanding of the security of $\mathcal{MQ}$ schemes.

A. Our Contribution

We propose a new general framework for the security of $\mathcal{MQ}$ schemes with respect to attacks that exploit the existence of linear subspaces. Our framework is based on two linearity measures that we borrow from symmetric cryptography, used to measure the resistance of symmetric primitives to linear cryptanalysis (cf. Matsui's attack on the DES cipher [32]). To our knowledge, this is the first time that the notion of linearity has been used to analyse the security of $\mathcal{MQ}$ schemes.

 ISBN: 978-1-61208-376-6

In particular, we take the linearity measure for vectorial functions introduced by Nyberg [33] already in 1992, and the (s,t)–linearity measure introduced recently by Boura and Canteaut [34] at FSE'13, and adopt them suitably in the context of $\mathcal{MQ}$ cryptography. We extend the first to a new notion of strong (s,t)–linearity in order to include an additional important parameter of the size of the vector subspace of the components of the function that have common linear space. We show that strong (s,t)–linearity and (s,t)–linearity are intrinsically connected to the security of $\mathcal{MQ}$ schemes, and can be used to explain almost all attacks on single field schemes, such as rank attacks, good keys attacks and attacks on oil and vinegar schemes. Indeed this is possible, since all these attacks share a common characteristic: They try to recover a subspace with respect to which the public key of an $\mathcal{MQ}$ scheme is linear. Therefore they can all be considered as linear attacks on $\mathcal{MQ}$ schemes.

We devise two generic attacks that separate the linear subspaces, and that are a generalization of the aforementioned known attacks. We present one of the possible modellings of the attacks using system solving techniques, although other techniques are possible as well. Using the properties of strong (s,t)–linearity and (s,t)–linearity, we show what are the best strategies for the attacks. Notably, the obtained systems of equations are equivalent to those that can be obtained using good keys [16], a technique based on equivalent keys and missing cross terms. By this we show that our new framework provides a different, elegant perspective on why good keys exist, and why they are so powerful in cryptanalysis.

Moreover, we use our framework to explain various pitfalls regarding design choices of $\mathcal{MQ}$ schemes and the successfulness of the linear attacks against them. Finally, we argue that linearity can be used as a solid measure for the susceptibility of $\mathcal{MQ}$ schemes to linear attacks, and also as a necessary tool for prudent design practice in $\mathcal{MQ}$ cryptography.

B. Organization of the Paper

The paper is organized as follows. In Section II, we briefly introduce the design principles of $\mathcal{MQ}$ schemes and also recall the well known measure of nonlinearity of functions. In the next Section III, we introduce the notion of strong (s,t)–linearity, which is basically an extension of the standard linearity measure and review the recently introduced (s,t)–linearity measure. In Sections IV and V, we show how the two linearity measures fit in the context of $\mathcal{MQ}$ cryptography. Some discussion on the matter proceeds in Section VI, and the conclusions are presented in Section VII.

II. Preliminaries

Throughout the text, $\mathbb{F}_q$ will denote the finite field of q elements, where $q = 2^d$, and $a = (a_1, \ldots, a_n)^\intercal$ will denote a vector from $\mathbb{F}_q^n$.

A. Vectorial Functions and Quadratic Forms

Definition 1: Let n, m be two positive integers. The functions from $\mathbb{F}_q^n$ to $\mathbb{F}_q^m$ are called (n,m) functions or vectorial functions. For an (n,m) function $f = (f_1, \ldots, f_m)$, f_i are called the coordinate functions of f.

Classically, a quadratic form

$$f(x_1, \ldots, x_n) = \sum_{1 \le i \le j \le n} \gamma_{ij} x_i x_j : \mathbb{F}_q^n \to \mathbb{F}_q$$

can be written as $x^\intercal \mathfrak{F} x$ using its matrix representation $\mathfrak{F}$. This matrix is constructed differently depending on the parity of the field characteristic. In odd characteristic, $\mathfrak{F}$ is chosen to be a symmetric matrix, where $\mathfrak{F}_{ij} = \gamma_{ij}/2$ for $i \neq j$ and $\mathfrak{F}_{ij} = \gamma_{ij}$ for $i = j$. Over fields $\mathbb{F}_q$ of even characteristic $\mathfrak{F}$ can not be chosen in this manner, since $(\gamma_{ij} + \gamma_{ji}) x_i x_j = 0$ for $i \neq j$. Instead, let $\widetilde{\mathfrak{F}}$ be the uniquely defined upper-triangular representation of f, *i.e.,* $\widetilde{\mathfrak{F}}_{ij} = \gamma_{ij}$ for $i \le j$. Now, we obtain a symmetric form by $\mathfrak{F} := \widetilde{\mathfrak{F}} + \widetilde{\mathfrak{F}}^\intercal$. Note that, in this case *only* the upper-triangular part represents the according polynomial and $\mathfrak{F}$ is always of even rank.

B. $\mathcal{MQ}$ Cryptosystems

The public key of a $\mathcal{MQ}$ cryptosystem is usually given by an (n,m) function $\mathcal{P}(x) = (p_1(x), \ldots, p_m(x)) : \mathbb{F}_q^n \to \mathbb{F}_q^m$, where

$$p_s(x) = \sum_{1 \le i \le j \le n} \widetilde{\gamma}_{ij}^{(s)} x_i x_j + \sum_{i=1}^{n} \widetilde{\beta}_i^{(s)} x_i + \widetilde{\alpha}^{(s)}$$

for every $1 \leqslant s \leqslant m$, and where $x = (x_1, \ldots, x_n)^\intercal$.

The public key $\mathcal{P}$ is obtained by masking a structured central (n,m) function $\mathcal{F} = (f_1, \ldots, f_m)$ using two secret linear transformations $S, T \in \mathrm{GL}_n(\mathbb{F}_q)$ and defined as $\mathcal{P} = T \circ \mathcal{F} \circ S$. We denote by $\mathfrak{P}^{(s)}$ and $\mathfrak{F}^{(s)}$ the $(n \times n)$ matrices describing the homogeneous quadratic part of p_s and f_s, respectively.

Example 1:

i. The internal map of UOV [1] is defined as $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, with central polynomials

$$f_s(x) = \sum_{i \in V, j \in V} \gamma_{ij}^{(s)} x_i x_j + \sum_{i \in V, j \in O} \gamma_{ij}^{(s)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(s)} x_i + \alpha^{(s)}, \tag{1}$$

for every $s = 1 \ldots m$, where $n = v + m$, $V = \{1, \ldots, v\}$ and $O = \{v+1, \ldots, n\}$ denote the index sets of the vinegar and oil variables, respectively. The public map $\mathcal{P}$ is obtained by $\mathcal{P} = \mathcal{F} \circ \mathcal{S}$, since the affine $\mathcal{T}$ is not needed (Indeed, any component $w^\intercal \cdot \mathcal{F}$ has again the form (1)).

ii. The internal map $\mathcal{F} : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ of C^* [8] is defined by

$$\mathcal{F}(x) = x^{2^\ell + 1}, \text{where } \gcd(2^\ell + 1, 2^n - 1) = 1.$$

This condition ensures that $\mathcal{F}$ is bijective.

iii. The representatives of the family of Stepwise Triangular Systems (STS) [4] have an internal map $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ defined as follows. Let L be the number of layers, and let $r_i, 0 \le i \le L$ be integers such that $0 = r_0 < r_1 < \cdots < r_L = n$. The central polynomials in the k-th layer are defined by

$$f_i(x_1, \ldots, x_n) = f_i(x_1, \ldots, x_{r_k}), \quad r_{k-1} + 1 \le i \le r_k.$$

We describe briefly two important cryptanalytic tools in $\mathcal{MQ}$ cryptography, that are of particular interest for us.

1) The MinRank Problem: The problem of finding a low rank linear combination of matrices is a known NP-hard linear algebra problem [35] known as MinRank in cryptography [18]. It has been shown that it underlies the security of several $\mathcal{MQ}$ cryptographic schemes [4][14][15]. It is defined as follows.

MinRank $MR(n, r, k, M_1, \ldots, M_k)$
Input: $n, r, k \in \mathbb{N}$, where $M_1, \ldots, M_k \in \mathcal{M}_{n \times n}(\mathbb{F}_q)$.

Question: Find – if any – a k-tuple $(\lambda_1, \ldots, \lambda_k) \in \mathbb{F}_q^k \setminus \{(0, 0, \ldots, 0)\}$ such that:

$$\mathrm{Rank}\left(\sum_{i=1}^{k} \lambda_i M_i\right) \leqslant r.$$

2) Good Keys: The concept of equivalent keys formally introduced by Wolf and Preneel in [36] is fundamentally connected to the security of $\mathcal{MQ}$ schemes. In essence, any key that preserves the structure of the secret map is an equivalent key. This natural notion was later generalized by Thomae and Wolf [16] to the concept of *good keys* that only preserve some of the structure of the secret map. Good keys improve the understanding of the level of applicability of MinRank against $\mathcal{MQ}$ schemes, and are a powerful tool for cryptanalysis. Good keys are defined as follows.

Let $k, 1 \leq k \leq m$ and $\mathcal{F} = \{f_1, \ldots, f_m\}$ be a set of polynomials of $\mathbb{F}_q[x_1, \ldots, x_n]$. Let $I^{(k)} \subseteq \{x_i x_j \,|\, 1 \leq i \leq j \leq n\}$ be a subset of the degree-2 monomials, and let $\mathcal{F}\big|_I = \{f_1\big|_{I^{(1)}}, \ldots, f_m\big|_{I^{(m)}}\}$ where $f_k\big|_{I^{(k)}} := \sum_{x_i x_j \in I^{(k)}} \gamma_{ij}^{(k)} x_i x_j$.

Definition 2 ([16]): Let $(\mathcal{F}, S, T), (\mathcal{F}', S', T') \in \mathbb{F}_q[x_1, \ldots, x_n]^m \times \mathrm{GL}_n(\mathbb{F}_q) \times \mathrm{GL}_m(\mathbb{F}_q)$ Let also $J^{(k)} \subsetneq I^{(k)}$ for all $k, 1 \leq k \leq m$ with at least one $J^{(k)} \neq \emptyset$. We call $(\mathcal{F}', S', T') \in \mathbb{F}_q[x_1, \ldots, x_n]^m \times \mathrm{GL}_n(\mathbb{F}_q) \times \mathrm{GL}_m(\mathbb{F}_q)$ a *good key* of $(\mathcal{F}, S, T)$ if and only if:

$$\left(T \circ \mathcal{F} \circ S = T' \circ \mathcal{F}' \circ S'\right) \wedge \left(\mathcal{F}\big|_J = \mathcal{F}'\big|_J\right).$$

C. Linearity of Vectorial Functions

Linearity is one the most important measures for the strength of an (n, m) function for use in symmetric cryptoprimitives. We provide here some well known results about this notion.

Definition 3 ([33]): The linearity of an (n, m) function f is measured using its Walsh transform, and is given by

$$\mathcal{L}(f) = \max_{w \in \mathbb{F}_q^m \setminus \{0\}, u \in \mathbb{F}_q^n} \left| \sum_{x \in \mathbb{F}_q^n} (-1)^{w^\intercal \cdot f(x) + u^\intercal \cdot x} \right|$$

The nonlinearity of an (n, m) function f is the Hamming distance between the set of nontrivial components $\left\{w^\intercal \cdot f | w \in \mathbb{F}_q^m \setminus \{0\}\right\}$ of f and the set of all affine functions. It is given by

$$\mathcal{N}(f) = (q-1)(q^{n-1} - \frac{1}{q}\mathcal{L}(f)).$$

Definition 4: A vector $w \in \mathbb{F}_q^n$ is called a linear structure of an (n, m) function f if the derivative $D_w f(x) = f(x + w) - f(x)$ is constant, *i.e.,* if

$$f(x + w) - f(x) = f(w) - f(0)$$

for all $x \in \mathbb{F}_q^n$. The space generated by the linear structures of f is called the linear space of f.

Nyberg [33] proved the following results.

Proposition 1 ([33]): The dimension of the linear space of an (n, m) function is invariant under bijective linear transformations of the input space and of the coordinates of the function.

Proposition 2 ([33]): Let $x^\intercal \mathfrak{F} x$ be the matrix representation of a quadratic form f. Then, the linear structures of f form the linear subspace $\mathrm{Ker}(\mathfrak{F})$.

The linear structures can provide a measure for the distance of the quadratic forms from the set of linear forms. Indeed the link is given by the following theorem.

Theorem 1 ([33]): 1) Let $x^\intercal \mathfrak{F} x$ be the matrix representation of a quadratic form f, and let $\mathrm{Rank}(\mathfrak{F}) = r$. Then the linearity of f is $\mathcal{L}(f) = q^{n - \frac{r}{2}}$.

2) Let f be a quadratic (n, m) function, and let $x^\intercal \mathfrak{F}_w x$ denote the matrix representation of a component $w^\intercal \cdot f$. Then the linearity of f is $\mathcal{L}(f) = q^{n - \frac{r}{2}}$, where $r = \min\{\mathrm{Rank}(\mathfrak{F}_w) | w \in \mathbb{F}_q^m\}$.

It is well-known that the linearity of an (n, m) function is bounded from below by the value $\mathcal{L}(f) \geqslant q^{\frac{n}{2}}$, known as the covering radius bound. It is tight for every even n, and functions that reach the bound are known as *bent* functions. It is also known [37] that bent functions exist only for $m \leqslant n/2$. A class of quadratic bent functions that has been extensively studied in the literature is the class of Maiorana-McFarland bent functions [38]. In general, an (n, m) function from the Maiorana-McFarland class has the form $f = (f_1, f_2, \ldots, f_m) : \mathbb{F}_{2^{n/2}} \times \mathbb{F}_{2^{n/2}} \to \mathbb{F}_2^m$ where each of the components f_i is

$$f_i(x, y) = L(\pi_i(x)y) + g_i(x), \tag{2}$$

where π_i are functions on $\mathbb{F}_{2^{n/2}}$, L is a linear function onto $\mathbb{F}_2^m$ and g_i are arbitrary $(n/2, m)$ functions. Nyberg [37] showed that f is an (n, m)-bent function if every nonzero linear combination of the functions $\pi_i, i \in \{1, \ldots, m\}$ is a permutation on $\mathbb{F}_{2^{n/2}}$.

Since the minimum linearity (maximum nonlinearity) is achieved only for $m \leqslant n/2$, permutations can not reach the covering radius bound. But, they can reach the Sidelnikov-Chabaud-Vaudenay (SCV) bound [39], valid for $m \geqslant n - 1$, which for $m = n$ odd, can be stated as: $\mathcal{L}(f) \geqslant q^{\frac{n+1}{2}}$. (n, n) functions, where n is odd, that reach the SCV bound with equality, are called Almost Bent (AB) functions.

As a direct consequence of Theorem 1 and the aforementioned bounds we have that quadratic (n, m) functions are

i. bent if and only if $\mathrm{Rank}(\mathfrak{F}_w) = n$ for every $w^\intercal \cdot f$,
ii. almost bent if and only if $\mathrm{Rank}(\mathfrak{F}_w) = n - 1$ for every $w^\intercal \cdot f$.

III. Strong (s, t)–linearity and (s, t)–linearity

We will show in the next sections that linearity plays a significant role for the security of $\mathcal{MQ}$ cryptosystems. However, in order to better frame it for use in $\mathcal{MQ}$ cryptography, we introduce the following notion of strong (s, t)–linearity. The motivation for this more precise measure comes from the recently introduced notion of (s, t)–linearity [34], that will also be discussed here in the context of $\mathcal{MQ}$ cryptography.

Definition 5: Let f be an (n, m) function. Then, f is said to be strongly (s, t)–linear if there exist two linear subspaces $V \subset \mathbb{F}_q^n$, $W \subset \mathbb{F}_q^m$ with $\mathrm{Dim}(V) = s$, $\mathrm{Dim}(W) = t$ such that for all $w \in W$, V is a subspace of the linear space of $w^\intercal \cdot f$.

Compared to the standard measure for linearity given in Definition 3, that actually measures the size of the vector space V, strong (s, t)–linearity also measures the size of the vector space W. We will see that this is particularly important in the case of $\mathcal{MQ}$ cryptosystems. We next provide some basic properties about strong (s, t)–linearity.

Proposition 3: If a function is strongly (s, t)–linear, then it is also strongly $(s-1, t)$–linear, and strongly $(s, t-1)$–linear.

Proof: Let f be strongly (s, t)–linear. Then there exists spaces V, W of $\mathrm{Dim}(V) = s$ and $\mathrm{Dim}(W) = t$, such that

every $a \in V$ is a linear structure of all the components $w^\intercal f$, where w is a basis vector of W. From here, a is also a linear structure of any subspace of W of dimension $t-1$. Therefore, f is also strongly $(s, t-1)$–linear. Similarly, the elements of any subspace of V of dimension $s-1$ are linear structures of all the components $w^\intercal f$, and thus, f is also strongly $(s-1, t)$–linear. ■

Proposition 4: Let f be an quadratic (n, m) function and $V \subset \mathbb{F}_q^n$ and $W \subset \mathbb{F}_q^m$ with $\mathrm{Dim}(V) = s$, $\mathrm{Dim}(W) = t$ be two linear spaces. Then f is strongly (s, t)–linear with respect to V, W if and only if the function f_W corresponding to all components $w^\intercal \cdot f$, $w \in W$ can be written as

$$f_W(x, y) = g_W(x) + L_W(y) \tag{3}$$

where $\mathbb{F}_q^n$ is a direct sum of U and V, $g_W : U \to \mathbb{F}_q^t$ is a quadratic function and $L_W : V \to \mathbb{F}_q^t$ is a linear function.

Proof: From Definition 5, f is strongly (s, t)–linear with respect to V, W if and only if V is a subspace of the linear space of $w^\intercal \cdot f$, for all $w \in W$. Now, for w a basis vector of W, $w^\intercal \cdot f$ can be written as $w^\intercal \cdot f(x, y) = g_w(x) + L_w(y)$ where $y \in V$ belongs to the linear space of $w^\intercal \cdot f$. Combining all the components for a basis of W we obtain the form (3). ■

Proposition 5: Let f be a quadratic (n, m) function. Then f is strongly (s, t)–linear with respect to V, W if and only if the function f_W corresponding to all components $w^\intercal \cdot f$, $w \in W$ is such that all its derivatives $D_a w^\intercal \cdot f$, with $a \in V$ are constant.

Recently, Boura and Canteaut [34] introduced a new measure for the propagation of linear relations through S-boxes, called (s, t)-linearity.

Definition 6 ([34]): Let f be an (n, m) function. Then, f is said to be (s, t)–linear if there exist two linear subspaces $V \subset \mathbb{F}_q^n$, $W \subset \mathbb{F}_q^m$ with $\mathrm{Dim}(V) = s$, $\mathrm{Dim}(W) = t$ such that for all $w \in W$, $w^\intercal \cdot f$ has degree at most 1 on all cosets of V.

Similarly as for strong (s, t)–linearity, it is true that

Proposition 6 ([34]): If a function is (s, t)–linear, then it is also $(s-1, t)$–linear, and $(s, t-1)$–linear.

Boura and Canteaut [34] proved that any (s, t)–linear function "contains" a function of the Maiorana-McFarland class, in the following sense.

Proposition 7 ([34]): Let f be an (n, m) function and $V \subseteq \mathbb{F}_q^n$ and $W \subseteq \mathbb{F}_q^m$ with $\mathrm{Dim}(V) = s$, $\mathrm{Dim}(W) = t$ be two linear spaces. Then f is (s, t)–linear with respect to V, W if and only if the function f_W corresponding to all components $w^\intercal \cdot f$, $w \in W$ can be written as

$$f_W = M(x) \cdot y + G(x)$$

where $\mathbb{F}_q^n$ is the direct sum of U and V, G is a function from U to $\mathbb{F}_q^t$ and $M(x)$ is a $t \times s$ matrix whose coefficients are functions defined on U.

A useful characterization of (s, t)–linearity, resulting from the properties of the Maiorana-McFarland class is through second order derivatives defined by $D_{a,b} f = D_a D_b f = D_b D_a f$.

Proposition 8 ([34]): Let f be an (n, m) function. Then f is (s, t)–linear with respect to V, W if and only if the function f_W corresponding to all components $w^\intercal \cdot f$, $w \in W$ is such that all its second order derivatives $D_{a,b} w^\intercal \cdot f$, with $a, b \in V$ vanish.

The two measures of linearity, even though they measure different linear subspaces are also interconnected. The following two propositions illustrate this connection.

Proposition 9: If a function is strongly (s, t)–linear, then it is also (s, t)–linear.

Proposition 10: If a quadratic (n, m) function f is $(\lceil \frac{n}{2} \rceil + s, 1)$–linear than it is strongly $(2s, 1)$–linear.

Proof: From Proposition3 [34] we have the fact that a $(s, 1)$–linear function has linearity $\mathcal{L}(f) \geqslant q^s$ (This comes from the fact that the linearity of a function is lower bounded by the linearity of any of its components.) Thus, if a quadratic (n, m) function is $(\lceil \frac{n}{2} \rceil + s, 1)$–linear, then $\mathcal{L}(f) \geqslant q^{\lceil \frac{n}{2} \rceil + s}$. From Theorem1 $\mathcal{L}(f) = q^{n - \frac{r}{2}}$, where $r = \min\{\mathrm{Rank}(\mathfrak{F}_w) | w \in \mathbb{F}_q^m\}$. From here $n - \frac{r}{2} \geqslant \lceil \frac{n}{2} \rceil + s$ and further $n - 2s \geqslant r$. Hence f is strongly $(2s, 1)$–linear. ■

In the next two sections, we will provide a general framework for the security of $\mathcal{MQ}$ schemes against linear cryptanalysis using the notions of strong (s, t)–linearity and (s, t)–linearity.

IV. The strong (s, t)–linearity measure for $\mathcal{MQ}$ systems

In this section, we show that strong (s, t)–linearity is fundamentally connected to the susceptibility of an $\mathcal{MQ}$ scheme to MinRank attacks and good keys attacks.

From Proposition 2 we have the following theorem.

Theorem 2: Let $f = (f_1, f_2, \ldots, f_m)$ be a quadratic (n, m) function, and let $\mathfrak{F}_1, \mathfrak{F}_2, \ldots, \mathfrak{F}_m$ be the matrix representations of the coordinates of f. Then, the MinRank problem $MR(n, r, m, \mathfrak{F}_1, \mathfrak{F}_2, \ldots, \mathfrak{F}_m)$ has a solution if and only if f is strongly $(n-r, 1)$–linear.

Proof: We see that $v = (v_1, \ldots, v_n) \in \mathbb{F}_q^n \setminus \{0\}$ is a solution to the MinRank problem $MR(n, r, m, \mathfrak{F}_1, \mathfrak{F}_2, \ldots, \mathfrak{F}_m)$ if and only if $\mathrm{Rank}\left(\sum_{i=1}^{n} v_i \mathfrak{F}_i\right) \leqslant r$, that is, if and only if $\mathrm{Dim}\left(\mathrm{Ker}\left(\sum_{i=1}^{n} v_i \mathfrak{F}_i\right)\right) \geqslant n - r$, *i.e.,* from Proposition 2, if and only if $v^\intercal \cdot f$ has at least $n-r$ linearly independent linear structures. Taking W to be the space generated by the vector v and V to be the linear space of $v^\intercal \cdot f$, from Definition 5 the last is equivalent to f being strongly $(n-r, 1)$–linear. ■

Example 2: From Theorem 2, it is clear that bent functions are resistant to MinRank attacks, since no linear combination of the components of the function has smaller rank than n. Thus, regarding MinRank attacks, bent functions are optimal for use as a secret map in $\mathcal{MQ}$ cryptosystems.

Example 3: Regarding encryption $\mathcal{MQ}$ schemes, a natural conclusion would be that AB permutations are the most suitable for use. One of the most scrutinized AB permutations are the Gold functions defined over $\mathbb{F}_{q^n}$ for odd n by:

$$f(x) = x^{q^\ell + 1}, \ \gcd(q^\ell + 1, q^n - 1) = 1, \ \gcd(\ell, n) = 1$$

where the first condition guarantees balancedness, and the second AB-ness. Notably, one of the most famous $\mathcal{MQ}$ schemes, the C^* scheme, uses an AB function, although there are variants that do not meet the second condition [21].

As mentioned before, AB functions have $\mathrm{Rank}(\mathfrak{F}_v) = n - 1$ for any component $v^\intercal \cdot f$. This means that each of the components have a linear space of dimension 1, and no two components share a linear space, *i.e.,* AB functions are only strongly $(1, 1)$–linear. Hence, MinRank for $r = n - 1$ is trivially satisfied and does not reveal anything more about the structure of the map.

 ISBN: 978-1-61208-376-6

The example of Gold functions from Example 3 implies that although MinRank on its own can be a good indicator of a weakness in a scheme, it does not provide a sufficient condition for mounting a successful attack. A better framework for the applicability of MinRank is provided by the concept of good keys (cf. Section II-B2). It should be emphasized that the definition of good keys (Definition 2), does not explicitly state the structure that is being preserved, thus, providing a framework even for structures not yet discovered. On the other hand, the motivation for good keys comes from the Rainbow band separation attack [40], that exploits (among others) a particular weakness connected to the presence of linear structures in the secret map. Moreover, known attacks that use MinRank, as well as other applications of good keys, again take advantage of the same property. Hence, we give a new definition for the special type of keys that separate the space of linear structures. This definition comes as a direct consequence of strong (s,t)–linearity. Later, we will also take a look at another weakness that the Rainbow band separation attack and its generalizations take advantage of, and we will also define the corresponding keys. We will call both types of keys *separation keys*.

Let V be a subspace of $\mathbb{F}_q^n$ of dimension $k \leq n$, and let S_V be an invertible matrix such that k of its rows form a basis of V. We note that the rest of the columns of the matrix can be arbitrary, as long as the matrix is invertible.

Definition 7: Let $(\mathcal{F}, S, T), (\mathcal{F}', S', T') \in \mathbb{F}_q[x_1, \ldots, x_n]^m \times \mathrm{GL}_n(\mathbb{F}_q) \times \mathrm{GL}_m(\mathbb{F}_q)$ and let $\mathcal{P} = T \circ \mathcal{F} \circ S = T' \circ \mathcal{F}' \circ S'$. We call $(\mathcal{F}', S', T')$ a *strong* (s,t) *separation key* for $\mathcal{P}$ if $\mathcal{P}$ is strongly (s,t)–linear with respect to two spaces V and W, $\mathrm{Dim}(V) = s$, $\mathrm{Dim}(W) = t$ and

$$S' = S_V^{\intercal}, \quad T' = T_W.$$

A strong (s,t) separation key separates the components of the public key that have a non empty common linear space. As a direct consequence of Definition 7 we have that:

Proposition 11: If $(\mathcal{F}', S', T')$ is a strong (s,t) separation key for $\mathcal{P}$, then it is also a good key for $\mathcal{P}$.

Many $\mathcal{MQ}$ cryptosystems, proposed so far have strong separation keys. As mentioned before, Rainbow [2] is one of the examples, but also all STS cryptosystems ([3], [4]), and all $\mathcal{MQ}$ cryptosystem that combine a layered structure with other types of design principles, including among others Branched C^* [41], MQQ-SIG [5], TTS [6], EnTTS [7], MFE [42]. Table I summarizes the different strong separation keys for some of these schemes.

TABLE I. EXAMPLES OF STRONG (s,t) SEPARATION KEYS FOR SOME $\mathcal{MQ}$ CRYPTOSYSTEMS

scheme	parameters	strong (s,t) separation keys
Branch.C^*	$(n_1, \ldots, n_b)$	$(\sum_i n_i, n - \sum_i n_i)$
STS	$(r_1, \ldots, r_L)$	$(n - r_k, r_k)$, $k = 1, \ldots, L-1$
Rainbow	$(v_1, o_1, o_2) = (18, 12, 12)$	$(12, 12)$
MQQ-SIG	$(q, d, n, r) = (2, 8, 160, 80)$	$(k, 80-k)$, $k = 1, \ldots, 79$
MFE	$(q^k, n, m) =$	$(2k, 10k)$, $(4k, 4k)$,
	$((2^{256})^k, 12, 15)$	$(6k, 2k)$, $(8k, k)$
EnTTS	$(n, m) = (32, 24)$	$(10, 14)$,$(14, 10)$

The known attacks on these systems, can all be considered as separation key attacks involving different techniques and optimizations. The framework of strong (s,t) linearity provides a unified way of looking at these attacks, and a *single* measure that can be used as criteria for the parameters of schemes that have strong separation keys. The next two theorems explain in detail how to mount a generic strong separation key attack, what is the complexity of the attack, and what is the best strategy for attack when the existence of a strong separation key is known. We decided to present the attack by representing the conditions for strong (s,t) linearity as systems of equations. This way we obtain completely equivalent systems to the ones that can be obtained using good keys, thus, offering another elegant point of view on why good keys exist. Note that this is not the only technique that can be used to recover strong (s,t) separation keys (for example we can use probabilistic approach). However, it provides a clear picture of the cases when the existence of a particular strong separation key is devastating for the security of $\mathcal{MQ}$ schemes.

Theorem 3: Let it be known that a strong (s,t) separation key exists for a given $\mathcal{MQ}$ public key $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ with matrix representations $\mathfrak{P}_w$ of a component $w^{\intercal} \cdot \mathcal{P}$.

i. The task of finding a strong (s,t) separation key $(S_V^{\intercal}, T_W)$ is equivalent to solving the system of bilinear equations

$$\mathfrak{P}_{w^{(i)}} \cdot a^{(j)} = 0, \ i \in \{1, \ldots, t\}, \ j \in \{1, \ldots, s\}, \quad (4)$$

in the unknown basis vectors $w^{(i)}$ of the space W, and the unknown basis vectors $a^{(j)}$ of the space V.

ii. The complexity of recovering the strong (s,t) separation key through solving the system (4) is

$$\mathcal{O}\left(t \cdot s \cdot n \cdot \binom{(n-s)s + (m-t)t + d_{reg}}{d_{reg}}^{\omega}\right) \quad (5)$$

where $d_{reg} = \min\{(n-s)s + (m-t)t\} + 1$, and $2 \leqslant \omega \leqslant 3$ is the linear algebra constant.

Proof: i. From Definition7 the existence of a strong (s,t) separation key $(S_V^{\intercal}, T_W)$ means that $\mathcal{P}$ is strongly (s,t)–linear with respect to two spaces V, W of dimension $\mathrm{Dim}(V) = s, \mathrm{Dim}(W) = t$. So the task is to recover these two spaces, *i.e.,* to recover some bases $\{a^{(1)}, \ldots, a^{(s)}\}$ and $\{w^{(1)}, \ldots, w^{(t)}\}$ of V and W, respectively. From Definition 5 and Proposition 2, $w \in W$ and $a \in V$ if and only if a is in the kernel of $\mathfrak{P}_w$, *i.e.,* if and only if $\mathfrak{P}_w \cdot a = 0$. Let the coordinates of the basis vectors $\{a^{(1)}, \ldots, a^{(s)}\}$ and $\{w^{(1)}, \ldots, w^{(t)}\}$ be unknowns. In order to insure that they are linearly independent, we fix the last s coordinates of $a^{(j)}$ to 0 except the $(n-j+1)$-th coordinate that we fix to 1, and similarly we fix the first t coordinates of $w^{(i)}$ to 0 except the i-th coordinate that we fix to 1. This way we can form the bilinear system (4). The solution of the system will yield the unknown bases of U and W. Note that if we get more than one solution, any of the obtained solutions will suffice. However, it can also happen that the system has no solutions. This is due to the fixed coordinates in the basis vectors, which can be done in the particular manner with probability of approximately $(1 - \frac{1}{q-1})^2$. Still, if no solutions, we can randomize the function $\mathcal{P}$ by applying linear transformation to the input space and the coordinates of the function, since from Proposition 1, this preserves the strong (s,t)–linearity of $\mathcal{P}$.

ii. From i., the system (4) consists of $t \cdot s \cdot n$ bilinear equations in two sets of variables of sizes $\nu_1 = (n-s)s$ and $\nu_2 = (m-t)t$, bilinear with respect to each other. The best known estimate of the complexity of solving a random system of bilinear equations is due to Faugere *et al.* [43], which says

that for the grevlex ordering, the degree of regularity of a generic affine bilinear zero-dimensional system over a finite field is upper bounded by

$$d_{reg} \leq \min(\nu_1, \nu_2) + 1. \quad (6)$$

Now, we use the F_5 algorithm for computing a grevlex Gröbner basis of a polynomial system [44][45], that has a complexity of

$$\mathcal{O}\left(\mu \cdot \binom{\nu_1 + \nu_2 + d_{reg}}{d_{reg}}^{\omega}\right), \quad (7)$$

for solving a system of $\nu_1 + \nu_2$ variables and μ equations ($2 \leqslant \omega \leqslant 3$ is the linear algebra constant). Using (6) and (7), we obtain the complexity given in (5). ∎

The complexity given in (5) is clearly not polynomial, since d_{reg} depends on n. However, it can be substantially improved using the properties of strong (s,t)–linearity from Proposition 3. This is shown in the next theorem.

Theorem 4: Let it be known that a strong (s,t) separation key exists for a given $\mathcal{MQ}$ public key $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ with matrix representations $\mathfrak{P}_w$ of a component $w^\intercal \cdot \mathcal{P}$.

i. The task of finding a strong (s,t) separation key can be reduced to

1. Solving the system of bilinear equations

$$\mathfrak{P}_w^{(i)} \cdot a^{(j)} = 0,\ i \in \{1, \ldots, c_1\},\ j \in \{1, \ldots, c_2\}, \quad (8)$$

in the unknown basis vectors $w^{(i)}$ of the space W, and the unknown basis vectors $a^{(j)}$ of the space V, where c_1, c_2 are small integers chosen appropriately.

2. Solving the system of linear equations

$$\mathfrak{P}_w^{(i)} \cdot a^{(j)} = 0,\ i \in \{c_1{+}1, ..., t\},\ j \in \{1, ..., c_2\},$$
$$\mathfrak{P}_w^{(i)} \cdot a^{(j)} = 0,\ i \in \{1, ..., c_1\},\ j \in \{c_2{+}1, ..., s\}, \quad (9)$$

in the unknown basis vectors $w^{(i)}$, $i \in \{c_1 + 1, \ldots, t\}$ of the space W, and the unknown basis vectors $a^{(j)}$, $j \in \{c_2 + 1, \ldots, s\}$ of the space V.

ii. The complexity of recovering the strong (s,t) separation key using the procedure from i. is

$$\mathcal{O}\left(\binom{(n-s)c_2 + (m-t)c_1 + d_{reg}}{d_{reg}}^{\omega}\right) \quad (10)$$

where $d_{reg} = \min\{(n-s)c_2, (m-t)c_1\}$.

Proof: i. The crucial observation that enables us to prove this part, is a consequence of Proposition 3. Recall that it states that strong (s,t)–linearity implies strong $(s-1,t)$ and strong $(s,t-1)$–linearity. Even more, if $\mathcal{P}$ is strongly (s,t)–linear, with respect to $V = Span\{a^{(1)}, \ldots, a^{(s)}\}$, $W = Span\{w^{(1)}, \ldots, w^{(t)}\}$, then it is strongly $(s-1,t)$–linear with respect to V_1, W, where $V_1 \subset V$, and strongly $(s,t-1)$–linear with respect to V, W_1, where $W_1 \subset W$. Hence, there exist two arrays of subspaces $V \supset V_1 \supset \cdots \supset V_{s-1}$ and $W \supset W_1 \supset \cdots \supset W_{t-1}$, such that $\mathcal{P}$ is strongly $(s-i,t-j)$–linear with respect to $V_i = Span\{a^{(1)}, \ldots, a^{(s-i)}\}$, $W_j = Span\{w^{(1)}, \ldots, w^{(t-j)}\}$. Thus, we can first recover the bases of some spaces V_{s-c_2}, W_{t-c_1}, and then extend them to the bases of V, W. Again, similarly, as in the proof of Theorem 3, we take the coordinates of the basis vectors $\{a^{(1)}, \ldots, a^{(s)}\}$ and $\{w^{(1)}, \ldots, w^{(t)}\}$ of V and W to be the unknowns, and again fix the last s coordinates of $a^{(j)}$ to 0 except the $(n-j+1)$-th coordinate that we fix to 1, and fix the first t coordinates of $w^{(i)}$ to 0 except the i-th coordinate that we fix to 1. Next, we pick two small constants c_1 and c_2, and form the bilinear system (8). Once the solution of this system is known, we can recover the rest of the bases vectors, by solving the linear system (9).

ii. The main complexity for the recovery of the key is in solving the system (8). Thus, proof for the complexity (10) is the same as for ii. Theorem 3. What is left, is to explain how the constants c_1 and c_2 are chosen. First of all, the system (8) consists of $c_1 \cdot c_2 \cdot n$ equations in $(n-s)c_2 + (m-t)c_1$ variables. We choose the constants c_1 and c_2 such that $c_1 \cdot c_2 \cdot n > (n-s)c_2 + (m-t)c_1$. Second, since the complexity is mainly determined by the value $d_{reg} = \min\{(n-s)c_2, (m-t)c_1\}$, these constants have to be chosen such that this value is minimized. Note that in practice, for actual $\mathcal{MQ}$ schemes, we can usually pick $c_1, c_2 \in \{1, 2\}$. ∎

The most important implication of the last theorem is that when $n - s$ or $m - t$ is constant we have a polynomial time algorithm for recovering a strong (s,t) separation key. This immediately implies that for any $\mathcal{MQ}$ scheme with this property we can recover in polynomial time a subspace on which the public key is linear.

Another implication is that it provides the best strategy of attacking an $\mathcal{MQ}$ scheme that possesses some strong (s,t) separation key. Indeed, since we need to minimize d_{reg}, we simply look for the minimal $m-t$ or minimal $n-s$ s.t. there exists a strong (s,t) separation key.

Example 4: Consider a (n,n) public key function from the family of STS systems (cf. Example 1.iii). From Table I, for the parameter set $(r_1, \ldots, r_L)$ we see that the scheme has a strong $(n-r_1, r_1)$ separation key and also a strong $(n-r_{L-1}, r_{L-1})$ separation key. For the first key, $n - s = r_1$ is small, so we can choose $c_2 = 1$ and c_1 such that $c_1 n > r_1 + (n-r_1)c_1$, *i.e.,* we can choose $c_1 = 2$. For the second key, $n - t = n - r_{L-1}$ is small so we can choose $c_1 = 1$ and c_2 such that $c_2 n > r_{L-1}c_2 + (n - r_{L-1})$, *i.e.,* we can choose $c_2 = 2$. Note that for small q it is perfectly fine to choose $c_1 = c_2 = 1$ in both cases, since then at most q solutions for the strong keys will need to be tried out.

The level of nonlinearity of a given function can be used as sufficient condition for the nonexistence of a strong (s,t) separation key.

Theorem 5: An (n,m) function f of linearity $\mathcal{L}(f) \leqslant q^{n-\frac{r}{2}}$ does not posses a strong (s,t) separation key for $s > n - r$.

Proof: From the linearity given, f does not have any component whose linear space has dimension bigger than $n-r$. Thus, f is not strongly (s,t)–linear for $s > n - r$, and does not have a corresponding strong (s,t) separation key. ∎

As a direct consequence, we have the following:

Corollary 1:

1) If $(\mathcal{F}', S', T')$ is a strong (s,t) separation key for C^*, then $s \leqslant 1$ and $t \leqslant 1$.
2) UOV using Maiorana-McFarland bent function does not posses a strong (s,t) separation key for any $s > 0$.

V. THE (s,t)-LINEARITY MEASURE FOR $\mathcal{MQ}$ SCHEMES

The size of the linear space of the components of an (n,m) quadratic function clearly provides a measure for the applicability of the function in $\mathcal{MQ}$ systems. Still, the notion of strong (s,t)–linearity can not provide a measure for the

existence of all the linear subspaces on which the restriction of an (n, m) function is linear.

For example, the secret map of UOV is linear on the oil space, regardless of its nonlinearity, and even when it is of maximum nonlinearity *i.e.,* when it is bent. The existence of this space enabled Kipnis and Shamir to recover it in cases when it is large enough, as in the original Oil and Vinegar scheme. Furthermore, the existence of such spaces improves the attack against Rainbow, compared to an attack that only considers linear spaces of the components.

We will show next that (s, t)–linearity provides a characterization for such subspaces, and thus, provides an improved measure for the security of $\mathcal{MQ}$ schemes.

Example 5: Let $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ be a UOV public mapping. In Section IV, we saw that the secret map of an UOV scheme belongs to the Maiorana-McFarland class. Thus, immediately, from Proposition 7, we conclude that $\mathcal{P}$ is (m, m)–linear, *i.e.,* $\mathcal{P}$ is linear on the oil space.

Now, similarly as in the previous section, we can define a special type of separation key, that separates the spaces with respect to which a function is (s, t)–linear.

Definition 8: Let $(\mathcal{F}, S, T), (\mathcal{F}', S', T') \in \mathbb{F}_q[x_1, \ldots, x_n]^m \times \mathrm{GL}_n(\mathbb{F}_q) \times \mathrm{GL}_m(\mathbb{F}_q)$ and let $\mathcal{P} = T \circ \mathcal{F} \circ S = T' \circ \mathcal{F}' \circ S'$. We call $(\mathcal{F}', S', T')$ an (s, t) *separation key* for $\mathcal{P}$ if $\mathcal{P}$ is (s, t)–linear with respect to two spaces V and W, $\mathrm{Dim}(V) = s$, $\mathrm{Dim}(W) = t$ and

$$S' = S_V^{\intercal}, \quad T' = T_W.$$

Conclusively, any public mapping that was created using an oil and vinegar mixing has a (s, t) separation key. Table II gives the (s, t) separation keys for some of the $\mathcal{MQ}$ schemes that combine a layered structure with oil and vinegar mixing.

TABLE II. EXAMPLES OF (s, t) SEPARATION KEYS FOR SOME $\mathcal{MQ}$ CRYPTOSYSTEMS

scheme	parameters	(s, t) separation keys
UOV	(q, v, o)	(o, o)
Rainbow	$(q, v, o_1, o_2) = (2^8, 18, 12, 12)$	$(12, 24)$, $(24, 12)$
MQQ-SIG	$(q, d, n, r) = (2, 8, 160, 80)$	$(8+8i, 80-8i), i \in \{0, ..., 9\}$
MFE	$(q^k, n, m) = ((2^{256})^k, 12, 15)$	$(2k, 2k)$,$(3k, 2k)$,$(4k, 4k)$
ℓIC	$(q^k, \ell) = (2^k, 3)$	$(2k, 2k)$,$(k, 2k)$
EnTTS	$(n, m) = (32, 24)$	$(10, 24)$,$(14, 14)$,$(24, 10)$

An interesting case regarding (s, t)–linearity is the C^* scheme for which we have the following result.

Proposition 12: Let $\mathcal{F} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be the secret map of C^* (cf. Example 1ii) and let $\gcd(\ell, n) = d$. Then, there exists a (d, n) separation key for these parameters of C^*.

Proof: First, let us consider the equation $D_{a,x}(f) = 0$ for a nonzero a. A little computation shows that it is equivalent to

$$ax(a^{2^\ell - 1} + x^{2^\ell - 1}) = 0,$$

and since we are interested in nonzero solutions we can restrict our attention to

$$a^{2^\ell - 1} + x^{2^\ell - 1} = 0.$$

This equation has $\gcd(2^\ell - 1, 2^n - 1) = 2^d - 1$ independent roots (see for example [46]). Thus, there exists a space V of dimension $\mathrm{Dim}(V) = d$ s.t. $D_{a,b}(f) = 0$, for all $a, b \in V$. This implies that $D_{a,b}(w^{\intercal} \cdot f) = 0$, for any $w \in \mathbb{F}_2^n$. Further, from Proposition 8 and Definition 8 it follows that there exists a (d, n) separation key for the given parameters. ■

Hence, the best choice for parameters of the C^* scheme is when $d = 1$, because in this case, the dimension of the space V is the smallest, and it is hardest to separate it. Note that this is analogous to the case of the UOV scheme, where also it is desirable to have smaller space V. The use of $d > 1$ was exactly the property that was exploited by Dubois *et al.* in [25] to break a modified version of the signature scheme SFLASH with $d > 1$ before the more secure version with $d = 1$ was broken due to the possibility to decompose the second order derivative into linear functions [24]. Even then, the authors of [25] noted that the condition $d = 1$ should be included in the requirements of the scheme, a fact that was overseen by the NESSIE consortium.

Note further that Proposition 12 implies that the dimension of the space V is invariant under restrictions of the public map (minus modifier). Thus, the SFLASH signature scheme also possesses a (d, k) separation key, where $k \leqslant n$ is the number of coordinates of the public key of SFLASH, and can equivalently be used to attack the modified version.

Similarly as for the case of strong (s, t) separation keys, (cf. Theorem 3 and Theorem 4), we can construct a generic algorithm that finds (s, t) separation keys. This part will be covered in the extended version of the paper. Here we focus our interest on a special type of separation keys, namely, (s, m) separation keys where the space W is the entire image space of the function. Indeed, schemes including UOV, Rainbow, Enhanced TTS, all posses exactly such keys. We will also show how the properties of (s, m)–linearity provide the best strategy for attacking schemes that posses (s, m) separation keys. Unfortunately, in this case it is more difficult to estimate the complexity of the attacks, since the obtained equations are of mixed nature. Therefore, we leave the complexity estimate for future work. Still, it is notable that we again arrive to equivalent systems of equation as in the case of good keys.

Theorem 6: Let it be known that an (s, m) separation key exists for a given $\mathcal{MQ}$ public key $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ with matrix representations $\mathfrak{P}_i := \widetilde{\mathfrak{P}}_i + \widetilde{\mathfrak{P}}_i^{\intercal}$ of the coordinate functions p_i.

i. The task of finding an (s, m) separation key $(S_V^{\intercal}, T_{\mathbb{F}_q^m})$ is equivalent to solving the following system of equations

$$\begin{aligned} a^{(j)}\mathfrak{P}_i a^{(k)} &= 0, \ i \in \{1, ..., m\}, \ j, k \in \{1, ..., s\}, j < k \\ a^{(k)}\widetilde{\mathfrak{P}}_i a^{(k)} &= 0, \ i \in \{1, ..., m\}, \ k \in \{1, ..., s\}, \end{aligned} \tag{11}$$

in the unknown basis vectors $a^{(j)}$ of the space V.

ii. The key can equivalently be found by

1. First solving the system of equations

$$\begin{aligned} a^{(j)}\mathfrak{P}_i a^{(k)} &= 0, \ i \in \{1, ..., m\}, \ j, k \in \{1, ..., c\}, j < k \\ a^{(k)}\widetilde{\mathfrak{P}}_i a^{(k)} &= 0, \ i \in \{1, ..., m\}, \ k \in \{1, ..., c\}, \end{aligned} \tag{12}$$

in the unknown basis vectors $a^{(k)}$, $k \in \{1, \ldots, c\}$ of the space V, for an appropriately chosen integer c.

2. And then, solving the system of linear equations

$$\begin{aligned} a^{(j)}\mathfrak{P}_i a^{(k)} &= 0, \\ i \in \{1, ..., m\}, j \in \{1, ..., c\}, k \in \{c+1, ..., s\}, j < k \end{aligned} \tag{13}$$

in the unknown basis vectors $a^{(k)}$, $k \in \{c + 1, ..., s\}$ of the space V.

Proof: i. From Definition 8, $\mathcal{P}$ is (s, m)–linear with respect to $V, \mathbb{F}_q^m$ where $\mathrm{Dim}(V) = s$. So we need to recover only some basis $\{a^{(1)}, \ldots, a^{(s)}\}$ of V. From Definition 6

and Proposition 8, the condition for (s,t)–linearity can be written as $D_{a^{(j)},a^{(k)}}f = 0$ for all $a^{(j)}, a^{(k)} \in V$, *i.e.,* as $a^{(j)}\mathfrak{P}_i a^{(k)} = 0$. Since $D_{a,a}f = 0$ for any a, we must write this condition as $a^{(k)}\widetilde{\mathfrak{P}}_i a^{(k)} = 0$. We ensure the linear independence of the unknown basis vectors $\{a^{(1)}, \ldots, a^{(s)}\}$ by fixing the last s coordinates of $a^{(j)}$ to 0 except the $(n-j+1)$-th coordinate that we fix to 1. The probability that we can fix the coordinates of the basis vectors in this way is approximately $1-\frac{1}{q-1}$. If the system does not yield a solution we randomize $\mathcal{P}$. In this way we form the system (11). It consists of $m\binom{s+1}{2}$ equations in $s(n-s)$ variables.

ii. From Proposition 6, we have that if $\mathcal{P}$ is (s,m)–linear, with respect to $V = Span\{a^{(1)}, \ldots, a^{(s)}\}$, $\mathbb{F}_q^m$, then it is $(s-1,m)$–linear with respect to $V_1, \mathbb{F}_q^m$, where $V_1 \subset V$. Hence, there exists an array of subspaces $V \supset V_1 \supset \cdots \supset V_{s-1}$, such that $\mathcal{P}$ is $(s-i,m)$–linear with respect to $V_i = Span\{a^{(1)}, \ldots, a^{(s-i)}\}$. Thus, we can first recover the basis of some space V_{s-c} and then extend it to the bases of V. That is, we first solve (12), and then we are left with the linear system (13). What is left is how we choose the constant c. The system (12) consists of $m\binom{c+1}{2}$ equations in $(n-s)c$ variables. It is enough to choose c such that $m\binom{c+1}{2} > (n-s)c$, in order to get a unique solution for the basis vectors. ■

Remark 1: Conditions for (s,t)–linearity have been used in other attacks not involving good keys or system solving. For example, the analysis of UOV in [1] uses exactly the conditions of Proposition 8 in order to test whether a subspace is contained in the oil space. An equivalent condition is also used in [47] again for analysis of UOV, and the authors' approach here is a purely heuristic one.

We conclude this part with an interesting result on the (s,m)–linearity of a random quadratic (n,m)-function.

Proposition 13: Let f be a randomly generated (n,m)-function over $\mathbb{F}_q$. Then, we can expect that there exist $q^{\frac{2(n-s)}{m(s+1)}}$ different subspaces V, such that f is (s,m)–linear with respect to $V, \mathbb{F}_q^m$.

Proof: Let the (n,m)-function f be given. Then f is (s,m) linear with respect to some space V if and only if there exist s linearly independent vectors $a^{(1)}, \ldots, a^{(s)} \in \mathbb{F}_q^n$ such that $V = Span\{a^{(1)}, \ldots, a^{(s)}\}$ and f is linear on every coset of V. Without loss of generality, we can fix s coordinates in each of the $a^{(k)}$ to ensure linear independence. In this manner, from the conditions of linearity from Theorem 6 we obtain a quadratic system of $m\binom{s+1}{2}$ equations in $s(n-s)$ variables. We can expect that such a system, on average has around $q^{\frac{s(n-s)}{m\binom{s+1}{2}}} = q^{\frac{2(n-s)}{m(s+1)}}$ solutions. For simplicity, we assume that the coordinates can be fixed in the particular manner. (In general, this is possible with a probability of $1-\frac{1}{q-1}$.) Note that all of these solutions span different subspaces. Indeed, suppose $(a_1^{(1)}, \ldots, a_1^{(s)})$ and $(a_2^{(1)}, \ldots, a_2^{(s)})$ are two different solutions. Then there exists i such that $a_1^{(i)} \neq a_2^{(i)}$. Then $a_2^{(i)}$ is not in the span of $a_1^{(1)}, \ldots, a_1^{(s)}$ because the fixed coordinates ensure linear independence. Thus, all the solutions generate different subspaces. ■

Proposition 13 implies that random quadratic (n,m) functions most probably have an $(\lfloor\frac{2n-m}{m+2}\rfloor, m)$ separation key. For the case of $n = m$, this means that there are no nontrivial (s,m) separation keys, but for the case of $n = 2m$, we can expect that there is a $(2,m)$ separation key, and for $n = 2m+4$, even a $(3,m)$ separation key.

Note that Proposition 13 further implies, that for $n \approx m^2$, a random quadratic (n,m) function is likely to have a (m,m) separation key. This is exactly the case identified by Kipnis *et al.* [1] as an insecure parameter set; see [1] for an efficient algorithm for recovering this space.

A. On the Reconciliation Attack on UOV

Recall the shape of the internal map of UOV from Example 1i. From Proposition 7 and Proposition 6, it follows that $\mathcal{P}$ is (i,m)–linear for any $1 \leqslant i \leqslant m$. In order to break the scheme, it is necessary to find a vector space V, such that $\mathcal{P}$ is (m,m)–linear with respect to $(V, \mathbb{F}_q^m)$. We will call any such space V an oil space. Ding *et al.* in [40] propose an algorithm that sequentially performs a change of basis that reveals gradually the space V. They call the algorithm *Reconciliation Attack on UOV*. In Figure 1, we present an equivalent version of the attack interpreted in terms of (s,t)–linearity (cf. Algorithm 2 [40]).

Input: UOV public key $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$.
$V_0 \leftarrow$ the zero-dimensional vector space
for $k := 1$ to m **do**
Find $a^{(k)} = (a_1^{(k)}, ..., a_v^{(k)}, 0, ..., 0, 1_{n-k+1}, 0, ..., 0) \in \mathbb{F}_q^n$, where 1_{n-k+1} denotes that the $(n-k+1)$-th coordinate is 1, by solving

$$\begin{aligned} a^{(j)}\mathfrak{P}_i a^{(k)} &= 0,\ i \in \{1, \ldots, m\}, j < k \\ a^{(k)}\widetilde{\mathfrak{P}}_i a^{(k)} &= 0,\ i \in \{1, \ldots, m\}, \end{aligned}$$

Construct a space $V_k \subset \mathbb{F}_q^n$ with $\mathrm{Dim}(V_k) = k$, s.t.

- $V_k = V_{k-1} \oplus Span\left\{a^{(k)}\right\}$, and
- $\mathcal{P}$ is (k,m)–linear with respect to $(V_k, \mathbb{F}_q^m)$.

end for
Output: An oil space $V = V_m$ of dimension m.

Figure 1. Reconciliation Attack on UOV in terms of (s,t)–linearity

It can be noticed that the Reconciliation attack is exactly an (s,m) separation key attack, where the constant c in Theorem 6 is chosen to be $c = 1$. However, we will show that the choice of $c = 1$ is justified only for the (approximately) balanced version of UOV, and not for any parameter set.

For example, consider the UOV parameter set $m = 28$ and $v = 56$. The public key in this case has a $(28,28)$ separation key. Using the reconciliation attack (equivalently if we take $c = 1$ in Theorem 6) in order to find a solution for $a^{(1)}$ one needs to solve a system of 28 quadratic equations in 56 variables. On average we can expect $q^{v-m} = q^{28}$ solutions. From the description of the reconciliation attack it seems that any of the solutions is a "good one", *i.e.,* it leads eventually to the recovery of the space V. This means that we can simply fix $v - m = 28$ variables and on average get a single solution by solving a system of 28 equations in 28 variables. In other words, this approach seems to work equally well for the balanced version of the scheme (when $m = v$) and for the unbalanced version.

Now, consider a UOV public key $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$. By definition it is (m,m)–linear, and also (s,m)–linear for every $s \leqslant m$. We can use Theorem 6 ii. to find the (m,m) separation key by choosing c such that $m\binom{c+1}{2} > (n-m)c$, *i.e.,* $c > 2(n/m - 2)$. We suppose that we have fixed $n - m$

coordinates of the vectors $a^{(1)}, \ldots, a^{(m)} \in \mathbb{F}_q^n$ to ensure linear independence. Suppose instead that we have chosen $c < 2(n/m - 2)$. Then Step 1 of Theorem 6 ii. will give on average $q^{2(n-m)/m(c+1)}$ solutions for the basis vectors, and all the solutions span a different space of dimension c such that $\mathcal{P}$ is (c, m) linear with respect to it (cf. Proposition 13). From the choice of the basis vectors, only one of these spaces is a subspace of the oil space V we are trying to recover. Thus, if $q^{2(n-m)/m(c+1)}$ is relatively big, it is infeasible to find the correct subspace. If we choose a wrong space, after several steps (depending on n, m, c), we will not be able to find any new linearly independent vectors. The reason is that from Proposition 13 it is expected that even in the random case such subspaces exist, but their dimension is much smaller than that of the actual oil space. Hence, we must choose at least $c \approx 2(n/m - 2)$. For example, $c = 1$ is suitable only for balanced versions where $n \approx 2m$, $c = 2$ can be used for n upto $\approx 3m$, and for the practically used parameters of $3m < n < 4m$ c should be 4 or even 5.

Remark 2: In [48], Thomae analyses the efficiency of the Reconciliation attack on UOV, and concludes that solving the equations from the first step of the attack is quite inefficient. He proposes instead to recover several columns from the good key at once and introduces some optimal parameter k for the number of columns, that corresponds to our parameter c in Theorem 6. However, the author does not discuss why the parameter is necessary, how to choose it, and what does it mean with regards to different parameters of UOV. The discussion above answers these questions.

B. Combining strong (s, t)–linearity and (s, t)–linearity

A number of existing $\mathcal{MQ}$ schemes combine several paradigms in their design. For example, Rainbow [2] or EnTTS [7] have a secret map with both layered and UOV structure. In other words, these schemes posses both types of separation keys. (Note that we do not talk about the trivial implication of a (s, t) separation key when a strong (s, t) separation key exists.) For example, Rainbow, with parameters (v, o_1, o_2), where $n = v + o_1 + o_2$, $m = o_1 + o_2$, has a $(o_2, o_1 + o_2)$ separation key with respect to $V, \mathbb{F}_q^m$, but also a strong (o_2, o_1) separation key with respect to the same subspace V and some $W \subset \mathbb{F}_q^m$. We can certainly focus on only one of the keys, and for example use either Theorem 4 or Theorem 6 to recover it. But since they share the same V the best strategy would be to combine the conditions for both strong linearity and linearity, *i.e.,* combine both theorems. A little computation shows that in this way, we can take both $c_1 = c_2 = 1$ in Theorem 4 and $c = 1$ in Theorem 6, *i.e.,* indeed we arrive to the most efficient case for recovery of V, W.

A similar argument applies to any $\mathcal{MQ}$ cryptosystem that encompasses layered and UOV structure. Notably, the possibility to use the aforementioned combination is exactly why the Rainbow band separation attack is much more efficient than the reconciliation attack.

VI. Prudent Design Practice for $\mathcal{MQ}$ schemes

In the previous sections, we saw that strong (s, t)–linearity and (s, t)–linearity provide a reasonable measure for the security of $\mathcal{MQ}$ cryptosystems. Certainly, in some schemes, the internal structure is clear from the construction, and such characterization may seem redundant. However, many schemes contain a hidden structure, that is invariant under linear transformations, (and thus, present in the public map) and that became obvious only after the scheme was broken. Furthermore, sometimes the constructions of the internal map lack essential conditions as in the case of SFLASH, where the specification was missing a condition on the $\gcd(\ell, n)$. We give another example concerning the MQQ-SIG scheme.

Example 6: The designers of the MQQ-SIG signature scheme in the construction of the internal map use a special type of quadratic $(2d, d)$ function $f = (f_1, \ldots, f_d)$ that is balanced when the first d arguments are fixed. They classify such functions depending on how many of f_i are linear, and as a security measure require that all should be quadratic. They further impose the restrictions that the rank of the matrix of f_i, $i = 1, \ldots, d$ should be high. While these are completely reasonable requirements, they do not properly reflect the linearity of the function, and are, thus, not at all sufficient to avoid instances of high linearity. Instead, a better requirement would be to impose a restriction on the rank of any of the components $v^\intercal \cdot f$, or equivalently to bound from above the linearity $\mathcal{L}(f)$.

Thus, it seems that a good practice is to include conditions about the linearity of the used functions. A nice concise criteria is the behaviour of the derivatives $D_a(f)$ and $D_{a,b}(f)$ of a function f (cf. Proposition 5 and 8) and the nonlinearity measure. As already mentioned, bent functions have the highest possible nonlinearity. However, since all quadratic bent functions over characteristic 2, are from the Maiorana-McFarland class [49], their relatively high (s, t)-linearity can be considered as a drawback. Conclusively, other functions that have low linearity in both senses (strong (s, t) and (s, t)) should be considered. AB functions have such properties. Unfortunately, Gold functions (cf. C^*) can not be used because of the presence of symmetry invariants, but it seems as a good idea to investigate other AB functions (or close to AB) for applicability in $\mathcal{MQ}$ cryptosystems.

VII. Conclusion

High nonlinearity of vectorial functions is nowadays widely accepted criterion in symmetric cryptography. As it turns out, it is also crucial for the security of $\mathcal{MQ}$ cryptosystems and thus can be used as a relevant security measure in their design. Indeed, in this paper, we provided a general framework based on linearity measures that encompasses *any* attack that takes advantage of the existence of linear spaces, and thus can be considered as a generalization of all such attacks. That is why, we believe that other notions from symmetric cryptography including resiliency and differential uniformity can successfully be adapted in the $\mathcal{MQ}$ context, and benefit further to the understanding of the security of $\mathcal{MQ}$ cryptosystems.

Acknowledgement

The first author of the paper is partially supported by FCSE, UKIM, R. Macedonia.

References

[1] A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced oil and vinegar signature schemes," in Advances in Cryptology – EUROCRYPT '99. Springer, 1999, pp. 206–222.

[2] J. Ding and D. Schmidt, "Rainbow, a new multivar. polynomial signature scheme." in ACNS, ser. LNCS, vol. 3531, 2005, pp. 164–175.

[3] T.-T. Moh, "A public key system with signature and master key functions," Comm. in Algebra, vol. 27, no. 5, 1999, pp. 2207–2222.

 ISBN: 978-1-61208-376-6

[4] C. Wolf, A. Braeken, and B. Preneel, "On the security of stepwise triangular systems," Designs, Codes and Cryptography, vol. 40, no. 3, 2006, pp. 285–302.

[5] D. Gligoroski et al., "MQQ-SIG - An Ultra-Fast and Provably CMA Resistant Digital Signature Scheme," in INTRUST, ser. LNCS, vol. 7222. Springer, 2011, pp. 184–203.

[6] B.-Y. Yang, J.-M. Chen, and Y.-H. Chen, "Tts: High-speed signatures on a low-cost smart card," in CHES, ser. LNCS, vol. 3156. Springer, 2004, pp. 371–385.

[7] B.-Y. Yang and J.-M. Chen, "Building secure tame-like multivariate public-key cryptosystems: The new tts." in ACISP '05, ser. LNCS, vol. 3574. Springer, 2005, pp. 518–531.

[8] H. Imai and T. Matsumoto, "Algebraic methods for constructing asymmetric cryptosystems." in AAECC, ser. LNCS, vol. 229. Springer, 1985, pp. 108–119.

[9] N. Courtois, L. Goubin, and J. Patarin, "Sflash, a fast asymmetric signature scheme for low-cost smartcards - primitive specification and supporting documentation." [Online]. Available: www.minrank.org/sflash-b-v2.pdf [Retrieved: September 2014].

[10] J. Patarin, "Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of asymmetric algorithms," in Advances in Cryptology – EUROCRYPT '96, ser. LNCS, vol. 1070. Springer, 1996, pp. 33–48.

[11] O. Billet, J. Patarin, and Y. Seurin, "Analysis of intermediate field systems," Cryptology ePrint Archive, Report 2009/542, 2009.

[12] C.-H. O. Chen, M.-S. Chen, J. Ding, F. Werner, and B.-Y. Yang, "Odd-char multivariate hidden field equations," Cryptology ePrint Archive, Report 2008/543, 2008.

[13] J. Patarin, N. Courtois, and L. Goubin, "Quartz, 128-bit long digital signatures." in CT-RSA, ser. LNCS, vol. 2020. Springer, 2001, pp. 282–297.

[14] N. Courtois and L. Goubin, "Cryptanalysis of the TTM cryptosystem," in Advances in Cryptology – ASIACRYPT '00, ser. LNCS, vol. 1976. Springer, 2000, pp. 44–57.

[15] A. Kipnis and A. Shamir, "Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization," in Advances in Cryptology – CRYPTO '99, ser. LNCS, vol. 1666. Springer, 1999, pp. 19–30.

[16] E. Thomae and C. Wolf, "Cryptanalysis of Enhanced TTS, STS and all its Variants, or: Why Cross-Terms are Important," in Progress in Cryptology – AFRICACRYPT '12, ser. LNCS, vol. 7374. Springer, 2012, pp. 188–202.

[17] L. Bettale, J.-C. Faugre, and L. Perret, "Cryptanalysis of hfe, multi-hfe and variants for odd and even characteristic," Designs, Codes and Cryptography, vol. 69, no. 1, 2013, pp. 1–52.

[18] N. T. Courtois, "Efficient zero-knowledge authentication based on a linear algebra problem MinRank," in Advances in Cryptology – ASIACRYPT '01, ser. LNCS, vol. 2248. Springer, 2001, pp. 402–421.

[19] C. Wolf and B. Preneel, "Large Superfluous Keys in Multivariate Quadratic Asymmetric Systems," in Public Key Cryptography, ser. LNCS, vol. 3386. Springer, 2005, pp. 275–287.

[20] P.-A. Fouque, L. Granboulan, and J. Stern, "Differential cryptanalysis for multivariate schemes," in Advances in Cryptology - EUROCRYPT '05, ser. LNCS, vol. 3494. Springer, 2005, pp. 341–353.

[21] J. Ding, "A new variant of the Matsumoto-Imai cryptosystem through perturbation." in PKC, 2004, pp. 305–318.

[22] V. Dubois, L. Granboulan, and J. Stern, "An efficient provable distinguisher for hfe," in ICALP (2), ser. LNCS, vol. 4052. Springer, 2006, pp. 156–167.

[23] ——, "Cryptanalysis of hfe with internal perturbation," in Public Key Cryptography, ser. LNCS, vol. 4450. Springer, 2007, pp. 249–265.

[24] V. Dubois, P.-A. Fouque, A. Shamir, and J. Stern, "Practical cryptanalysis of sflash." in Advances in Cryptology – CRYPTO '07, ser. LNCS, A. Menezes, Ed., vol. 4622. Springer, 2007, pp. 1–12.

[25] V. Dubois, P.-A. Fouque, and J. Stern, "Cryptanalysis of sflash with slightly modified parameters." in Advances in Cryptology – EUROCRYPT '07, ser. LNCS, M. Naor, Ed., vol. 4515. Springer, 2007, pp. 264–275.

[26] J. Patarin, "Cryptoanalysis of the Matsumoto and Imai public key scheme of EUROCRYPT '88," in Advances in Cryptology – CRYPTO '95, 1995, pp. 248–261.

[27] "Nessie: New european schemes for signatures, integrity, and encryption," 2003. [Online]. Available: https://www.cosic.esat.kuleuven.be/nessie/ [Retrieved: September 2014].

[28] S. Tsujii, M. Gotaishi, K. Tadaki, and R. Fujita, "Proposal of a signature scheme based on sts trapdoor," in Post-Quantum Cryptography, ser. LNCS. Springer, 2010, vol. 6061, pp. 201–217.

[29] K. Sakumoto, T. Shirai, and H. Hiwatari, "On provable security of uov and hfe signature schemes against chosen-message attack," in Post-Quantum Cryptography, ser. LNCS, 2011, vol. 7071, pp. 68–82.

[30] D. Smith-Tone, "On the differential security of multivariate public key cryptosystems," in Post-Quantum Cryptography, ser. LNCS. Springer, 2011, vol. 7071, pp. 130–142.

[31] R. Perlner and D. Smith-Tone, "A classification of differential invariants for multivariate post-quantum cryptosystems," in Post-Quantum Cryptography, ser. LNCS. Springer, 2013, vol. 7932, pp. 165–173.

[32] M. Matsui, "Linear cryptanalysis method for des cipher," in Advances in Cryptology - EUROCRYPT '93, ser. LNCS, T. Helleseth, Ed. Springer Berlin Heidelberg, 1994, vol. 765, pp. 386–397.

[33] K. Nyberg, "On the construction of highly nonlinear permutations," in Advances in Cryptology – EUROCRYPT '92, ser. LNCS, vol. 658. Springer, 1992, pp. 92–98.

[34] C. Boura and A. Canteaut, "A new criterion for avoiding the propagation of linear relations through an Sbox," in FSE 2013 - Fast Software Encryption, ser. LNCS. Singapore: Springer, 2014.

[35] W. Buss, G. Frandsen, and J. Shallit, "The computational complexity of some problems of linear algebra." J. Comput. System Sci., 1999.

[36] C. Wolf and B. Preneel, "Equivalent Keys in Multivariate Quadratic Public Key Systems," Journal of Mathematical Cryptology, vol. 4, April 2011, pp. 375–415.

[37] K. Nyberg, "Perfect nonlinear s-boxes," in Advances in Cryptology – EUROCRYPT '91, ser. LNCS, D. W. Davies, Ed., vol. 547. Springer, 1991, pp. 378–386.

[38] J. F. Dillon, "Elementary hadamard difference sets," Ph.D. dissertation, University of Maryland, 1974.

[39] F. Chabaud and S. Vaudenay, "Links between differential and linear cryptoanalysis." in Advances in Cryptology – EUROCRYPT '94, ser. LNCS, vol. 950. Springer, 1994, pp. 356–365.

[40] J. Ding, B.-Y. Yang, C.-H. O. Chen, M.-S. Chen, and C.-M. Cheng, "New differential-algebraic attacks and reparametrization of rainbow." in ACNS, ser. LNCS, vol. 5037, 2008, pp. 242–257.

[41] J. Patarin and L. Goubin, "Asymmetric cryptography with s-boxes." in ICICS, ser. LNCS, vol. 1334. Springer, 1997, pp. 369–380.

[42] J. Ding, L. Hu, X. Nie, J. Li, and J. Wagner, "High order linearization equation (hole) attack on multivariate public key cryptosystems." in Public Key Cryptography '07, ser. LNCS, vol. 4450, 2007, pp. 233–248.

[43] J.-C. Faugère, M. S. E. Din, and P.-J. Spaenlehauer, "Gröbner bases of bihomogeneous ideals generated by polynomials of bidegree (1, 1): Algorithms and complexity," J. Symb. Comput., vol. 46, no. 4, 2011, pp. 406–437.

[44] M. Bardet, J.-C. Faugère, and B. Salvy, "On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations," in ICPSS, 2004, pp. 71–75.

[45] M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang, "Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems," in MEGA '05, 2005.

[46] R. Lidl and H. Niederreiter, Finite Fields. Cambridge UP, 1997.

[47] A. Braeken, C. Wolf, and B. Preneel, "A study of the security of unbalanced oil and vinegar signature schemes." in CT-RSA, ser. LNCS, A. Menezes, Ed., vol. 3376. Springer, 2005, pp. 29–43.

[48] E. Thomae, "About the Security of Multivariate Quadratic Public Key Schemes," Ph.D. dissertation, Ruhr-University Bochum, 2013.

[49] L. Budaghyan, C. Carlet, T. Helleseth, and A. Kholosha, "Generalized bent functions and their relation to maiorana-mcfarland class." in ISIT '12. IEEE, 2012, pp. 1212–1215.

 ISBN: 978-1-61208-376-6

Managed Certificate Whitelisting – A Basis for Internet of Things Security in Industrial Automation Applications

Rainer Falk and Steffen Fries
Siemens AG
Corporate Technology
Munich, Germany
Email: {rainer.falk|steffen.fries}@siemens.com

***Abstract*—Device authentication is a basic security feature for automation systems and for the future Internet of Things. The design, setup and operation of a practically usable security infrastructure for the management of required device credentials – as cryptographic device keys and device certificates – is a huge challenge. Also, access permissions defining authorized communication peers have to be configured on devices.**

The set-up and operation of a public key infrastructure PKI with registration authority (RA) and certification authority (CA), as well as the management of device permissions has shown to be burdensome for industrial application domains.

A recent approach is based on certificate whitelisting. It is currently standardized for field device communication within energy automation systems by IEC 62351 in alignment with ITU-T X.509. This new approach changes the way how digital certificates are used and managed significantly.

After describing the new approach of managed certificate whitelisting and giving a summary of ongoing standardization activities, an example for the application in a real-world application domain is described. Needs for further technical work are derived, and solution options are presented.

Keywords*—*Digital certificate, certificate whitelisting, credential management, PKI, device authentication, Internet of Things.

I. Introduction

Industrial automation systems, e. g., for energy automation, railway automation or process automation, use open communication protocols as Ethernet, wireless local area network (WLAN) IEEE 802.11 [1], transmission control protocol (TCP), user datagram protocol (UDP), and hypertext transfer protocol (HTTP) [2]. The communication can be protected using standard security protocols like IEEE 802.1X/MACsec [3], Internet key exchange (IKE) [4] with Internet protocol security (IPsec) [5], secure shell (ssh) [6], secure sockets layer (SSL) [7], and transport layer security (TLS) [8]. Often, asymmetric cryptographic keys and corresponding device certificates are used. Symmetric keys would not not scale well for the huge number of involved devices.

In a common realization of a public key infrastructure PKI, digital certificates are issued by a trusted certification authority (CA). This allows to authenticate devices. Additionally, access permissions are defined for authorized communication peers. While this technology could be the basis for a global, uniform secure communication, in reality, the deployment and adoption of PKIs is often limited to HTTP server authentication. A reason for that is the significant effort required to set-up, maintain, and use a PKI.

The problem addressed in this paper is the practical management of device certificates for field-level automation devices. A certificate infrastructure is required that is suitable for an operational automation environment. Main considerations are the demand for extremely high system availability, requiring that the automation system can continue to operate in an autonomous island mode, and the fact that many automation systems are set-up as separate network segments that have no or only limited connectivity with general office networks or even the public Internet. Moreover, the fact that these systems are typically engineered, e.g., that the communication relations are known up front, can be leveraged for certificate and access management.

A self-contained certificate management tool (command line tool, or with GUI) can be well suited for a small number of devices, but it does not scale well to scenarios with a larger number of devices. A full-blown PKI infrastructure could be efficient for an extremely huge number of devices, but these go beyond the scale of a common single automation systems.

The problem can be summarized that a solution is needed that can be set-up and operated autonomously within a certain automation environment without relying on a globally accepted certification authority, and that scales well for "mid-size" automation environments, for which a self-contained certificate tool is too small, and a full PKI solution would be too complex and costly. It may be also advantageous to avoid the need for deploying a separate identity and access management infrastructure.

The remainder of this paper if structured as follows: After summarizing background work in Section II, Section III describes certificate whitelists as a new paradigm for using digital certificates. The management of certificate whitelists is described generically in Section IV, and a specific adaption into energy automation systems is outlined in Section V. An outlook to possible future extensions is given in Section VI.

II. Background and Previous Work

Secure communication protocols, digital certificates, and public key infrastructure PKI [9], [10] have been dealt with intensively for years. An introduction is given in common text books on IT security [11]. The remainder of this section summarizes shortly major aspects that are relevant to managed certificate whitelists.

 ISBN: 978-1-61208-376-6

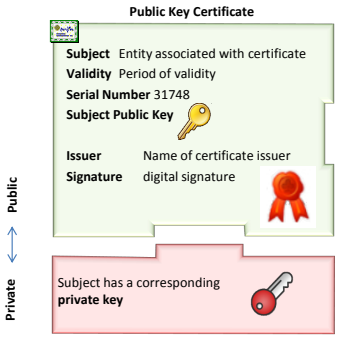


Fig. 1. Digital Certificate (X.509)

A. Device Communication Security Technologies

Digital device certificates are the basis for device communication security as used in industrial automation systems, and in the future Internet of Things (IoT). Major communication security protocols are available for the different layers of the communication protocol stack that support digital device certificates for authentication:

- Link layer: The standard 802.1X [3] provides Network Access Control to restrict access to a network only for authenticated devices. It is also possible to encrypt the communication link using the MACsec of 802.1X.
- Network layer: The communication can be protected with IPsec [5] on the network layer. The required security associations can be established by the IKE [4] protocol.
- Transport layer: With TLS [8], the successor of the SSL protocol [7], communication can be protected on the transport layer.
- Application layer: SSH or WS-Sec are available to protect application layer protocols as HTTP, SOA (REST, SOAP), CoAP, XMPP, or MQTT.

B. Digital Certificates

The main purpose of a digital certificate is to reliably assign information about the subject, i. e., the owner, of a public key. The owner may be identified by its name or email address in case of a person, or by its network name (DNS name) or IP address of a server. Additional information encodes usage information about the public key respectively the digital certificate, as validity period, and allowed key usages as user authentication or email encryption. For device certificates, it is possible to encode the device manufacturer, the device model, and the serial number within a device certificate.

The most commonly used certificate format is ISO X.509 [9]. Figure 1 shows the format and some examplary fields. The main purpose of a digital certificate is to bind a public key (`Subject Public Key Info`) of an entity to the name of the entity (`Subject`). Additional information as the validity period, the issuer, and usage restrictions can be included as well.

When a digital certificate of a subject is validated by a communication peer, it is verified that the certificate has a valid digital signature of a trusted certification authority. It is furthermore verified that the entries of the certificate match the intended usage. It may also be verified whether the certificate has been revoked. A revocation check may verify whether a given certificate is included in a certificate revocation list (CRL), or an online revocation status check may be performed using the open certificate status protocol (OCSP) [12]. In either case, at least partial online access to a PKI entity that is issuing certificates and providing revocation information is needed at least from one component in an automation network or cell. This component may further distribute the information within the automation cell.

C. Certificate Root Key

A digital certificate has to be validated before it is accepted. This includes a check whether the digital signature protecting the certificate is trusted. The standard approach is to use a set of trusted root certificates for certification authorities CA. A certificate is accepted if its signature chain can be verified back to a trusted root certificate. The root certificate may belong to a globally recognized CA, or to a local CA that is accepted only within an administrative domain, e. g., within a single operator network. If no PKI with CA is available, it is also possible to use self-signed certificates. This means that each certificate is signed with the private key associated with the public key contained in the certificate. Such certificates have to be configured as trusted in the same way as trusted root certificates, i. e., the (self-signed) certificates of trusted peers have to be configured explicitly. This requires to store the trusted peer information (root CA, or self signed certificates) in a secure manner, as this information is crucial for system security.

D. Certificate Whitelisting

The basic concept of certificate whitelists is well-known. The underlying idea is to enumerate explicitly all authorized certificates. A certificate is validated successfully only if it is contained in the certificate whitelist. The whitelist may contain the certificates directly, or reference the certificates by their serial number and issuer, by the certificate fingerprint, or by the public key. The latter avoids issuing a new whitelist, when a certificate is updated.

Such a certificate whitelist can be considered and used also as an access control list that contains the certificates of all authorized subjects. Without using specific certificate extensions, the different operations cannot be distinguished, however. The configuration of the set of trusted root certificates is also a form of certificate whitelists. It is known to check whether the certificate of a communication peer is included in a certificate whitelist [13]. Also, the Microsoft Digital Rights Management License Protocol is using a certificate whitelists [14].

As these certificate whitelists have been used as a proprietary means for configuring a list of trusted certificates, or to be more precise a *set* of trusted certificates, the approach has been rather limited as general means for certificate management.

 ISBN: 978-1-61208-376-6

III. Certificate Management and Validation Using Certificate Whitelists

The set-up and operation of a public key infrastructure has shown to require significant effort and costs. This has been a limiting factor for the practical usage of public key cryptography. Ongoing standardization activities define the technological basis for simpler usage of public key cryptography for industrial automation environments and the future Internet of Things.

While a certificate whitelist has been used so far as proprietary means for configuring some digital certificates as trusted, a certificate whitelists format is currently standardized for the smart energy grid environment. It has been acknowledged that the application of certificate whitelists in restricted environments supports the long term administration of security parameters. Hence, standardizing the format is the next consequent step to ensure interoperability of different vendor products.

A certificate whitelist is a data structure containing respectively referencing a set of trusted digital certificates. A certificate can be referenced by its serial number and issuer, or by a fingerprint of the certificate (hash value). The certificate whitelist is signed using a whitelist root key of trust (WROT).

A certificate is validated successfully if it is contained in a corresponding certificate whitelist. Further checks on the contents of the certificate as the name of the subject, the certificate extensions, and the certificate signature are performed in the usual way.

Certificate whitelists can be used with certificates issued by a CA, or with self-signed certificates. A common technological basis is provided for smaller environments using self-signed certificates as well as environments using a PKI for issuing certificates. So, a smooth migration from self-signed certificates to a local PKI and even towards global PKI is provided.

A certificate can be revoked easily by not including it anymore in the certificate whitelists. However, it is also possible to check the certification revocation status using certificate revocation lists [9] or using the online certificate status protocol OCSP [12].

1) Standardization Activities: Currently ongoing standardization activities performed by ISO/IEC 62351 [15] in alignment with ITU-T X.509 [9] define the usage of certificate whitelists for energy automation systems. Currently, a format is defined for a certificate whitelist. Figure 2 shows a recent proposal for a certificate whitelist. It is based on the format of a certificate revocation list CRL, but its assigned type (`CertificateWhiteList`) distinguishes it from a CRL. Also, the intended scope of a certificate whitelist is defined by a specific attribute `scope`. It allows a client to verify whether a certain certificate whitelist has in fact been intended for a specific purpose. For example, the IP addresses or DNS names of devices for which the whitelist is intended to be used can be included.

The target scope of a certificate whitelist can be explicitly encoded in a certificate whitelist. Therefore, a certificate whitelist cannot be used unintentionally for a different purpose as the intended purpose at time of compilation. Certificate whitelists can be compiled once during as part of engineering. Alternatively, end devices can pull a certificate whitelist from a whitelist certificate server in defined time intervals. The CWL can also be pushed to the field devices.

A digital certificate may be intended to be used only within a certificate whitelisting environment. To ensure that a certificate is in fact validated successfully only together with a corresponding whitelist, it is possible to include a corresponding extension in the certificate. The extension marks it explicitly to be accepted only if it is included in a certificate whitelist. A corresponding certificate extension is currently defined by ISO/IEC 62351 [15].

The validation of a certificate depends on whether it contains a certificate whitelist extension. Figure 3 shows the relevant checks. If a certificate includes the whitelisting extension, it is required that the corresponding whitelist is available and that the certificate is in fact included in the whitelist.

```
CertificateWhiteList  ::=  SEQUENCE  {
     tbsCertWhiteList     TBSCertWhiteList,
     signatureAlgorithm   AlgorithmIdentifier,
     signatureValue       BIT STRING
}
TBSCertWhiteList  ::=  SEQUENCE  {
   version      Version OPTIONAL,
                -- if present must be v1
   signature    AlgorithmIdentifier,
   issuer       Name,
   thisUpdate       Time,
   nextUpdate       Time OPTIONAL,
   scopedList       SEQUENCE OF SEQUENCE {
      scope         ScopeConstraints,
                    -- geographic,organizational
      authorizedCertificates    SEQUENCE OF SEQUENCE  {
         fingerprint     AlgorithmIdentifier, -- for FP creation
         certIdentifier::== CHOICE {
         serialCert      [0] CertificateSerialNumber,
         fingerprintCert [1] OCTET STRING -- FP of certificate
         fingerprintPK   [2] OCTET STRING -- FP of public key
   }
   certificateIssuer     Name OPTIONAL,
   cwlEntryRestriction   [0] EXPLICIT Extension OPTIONAL
                         -- further restrictions of cert. usage
   }
   }
   cwlExtensions [0] EXPLICIT Extensions OPTIONAL
                 {- for future use
}
```

Fig. 2. Certificate Whitelist Format [15]

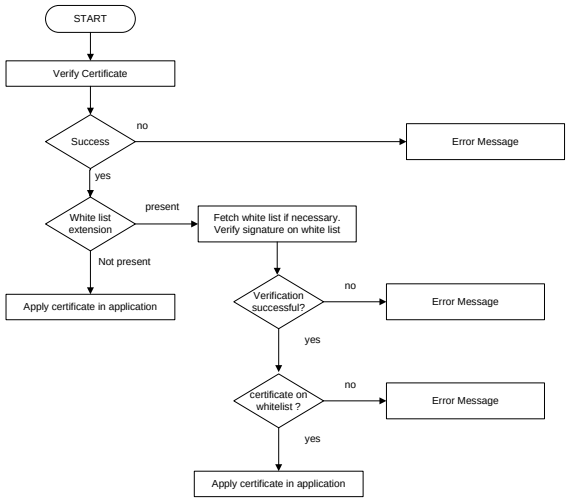


Fig. 3. Validation of a Certificate with Certificate

IV. Managed Certificate Whitelists

The introduction of certificate whitelisting implies the need for a management system for certificate whitelists. Managed certificate whitelists are a new approach for using public key cryptography in a practical, efficient and effective way. It is particularly suited for systems with well-known set of devices and their communication relationships, as it is common for networked automation systems. As the management of whitelists can be fully automated, it scales well to larger number of devices, although due to the increasing size of whitelists the targeted application environment is characterized by a number of devices within a range up to some 100 to some 1000 devices. It integrates well within existing industrial workflows for installing or exchanging devices, as device configuration databases are kept up-to-date within automation systems. So, the information that is required to generate updated certificate whitelists is already available. Once certificate whitelists have been generated and installed on the target devices, the target devices can operate autonomously even if the security infrastructure is not available. This is an important property for automation environments with high availability requirements to ensure that the automation system can continue to operate even if backend systems are temporarily unavailable.

A. Whitelist Generation and Distribution

The basic concept for automatic whitelist management is rather straightforward. Using information which is available in common automation systems about the devices and their communication relationships within a networked automation system, several purpose-specific – and also device-specific if needed – certificate whitelists are generated automatically. The whitelists are distributed to the target devices using remote configuration protocols. For example, secure copy scp [6], HTTPS [16], or OPC-UA [17] can be used to distribute configuration files securely to the target devices.

Figure 4 shows the main components involved in the automatic management of certificate whitelists. A central device management component accesses a device database including all registered devices of a networked automation system and their associated device certificates. Using automation system configuration data, the communication relationships are determined. Based on this information, certificate whitelists can be compiled for the different communication purposes as automation control communication, supervisory control communication, remote service access and diagnostic access. Depending on policy, device-specific certificate whitelists can be compiled, or certificate whitelists for defined purposes and target device classes. The certificate whitelists are created and provided to a device management system that configures the relevant certificate whitelists on the target devices. As important difference to a certification revocation list CRL, a certificate whitelist will usually be provided and be signed by the operator, not by the certification authority (CA). This has the advantage that an automation system operator can use managed certificate whitelists easily with certificates issued by different CAs, and even with self-signed certificates.

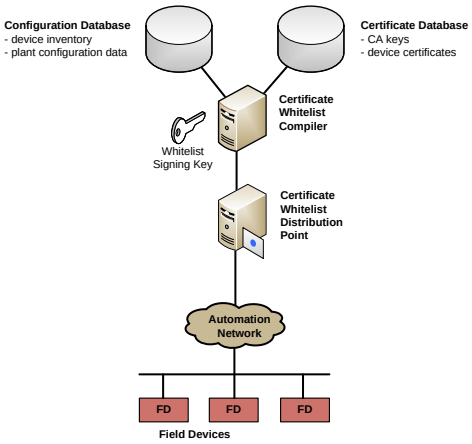


Fig. 4. Certificate Whitelist Management System

For networked automation systems with a typical size of some 100 to some 1000 devices, such a certificate management system based on whitelisting provides several advantages for the application in real-world industrial usage scenarios: A local PKI or even self-signed certificates can be used, so that a deployment with a very limited security infrastructure is possible. For the operation of the automation system, no continous reachability or availability of the whitelisting security infrastructure is required. So, the availability of the automation system availability does not depend on the availability of the security infrastructure. A commonly availably device management infrastructure can be extended easily for automatically creating and distributing certificate whitelists. It is possible to use a certificate whitelist only for authentication. Authorization checks would then be performed in addition, e. g., by checking an access control list. However, a certificate whitelist can be used directly as access control list as well. Different certificate whitelists would be configured for the different types of access (e. g., control communication, service access, diagnosis). The current proposal for a CWL structure considers this by supporting the encoding of a list of lists. Moreover, within the CWL, further certificate usage restrictions may be encoded. One example is the definition of dedicated applications or communication protocols which are allowed to utilize a dedicated certificate. Using this approach, the communication peer could refuse to accept a certificate included on the CWL if it is not associated within the CWL with the currently used communication protocol.

This has the advantage that no separate identity and access management infrastructure is needed, and that access control decisions can be performed by a field device when the backend systems are not available. These properties make certificate whitelisting a very interesting approach for managing digital certificates in typical industrial automation systems.

B. Example Usage Scenarios

Typical workflows in industrial automation systems are the initial installation, the replacement, and removal of devices. As

 ISBN: 978-1-61208-376-6

device configuration databases are already maintained as part of these workflows, the information for updating certificate whitelists is available without any extra effort required from the service personnel. As changes in the configuration are detected by the certificate whitelisting system, the generation of updated certificate whitelists is started and the deployment to affected target devices is triggered.

V. Application Within Energy Automation Systems

The general approach of using managed certificate whitelists as described in the previous section can be applied for energy automation systems (smart grid). Figure 5 shows a substation automation system. A substation typically transforms voltage levels, and includes power monitoring and protection functions. Figure 5 shows separate network zones of the substation communication network. The field devices that perform the actual field level functionality of monitoring and acting on the electric power are called intelligent energy devices (IED). They are monitored and controlled by a substation controller, realizing a realtime automation system. Energy automation protocols are defined by the standard IEC61850 [18] which defined the Generic Object Oriented Substation Events (GOOSE) protocol. Additional network zones are available for local and remote service access, for integrating intelligent field devices with serial interfaces, and for support functions (file server, historian server for logging, remote access server, terminal server). A substation is connected to the utility communication network providing backend services like supervisory control and data acquisition (SCADA). Firewalls are used to control the traffic flow between zones.

A hierarchical creation and distribution of certificate whitelists to a substation may be realized in the following way: A utility operator creates a substation-specific certificate whitelist (substation cert whitelist) based on the engineering information for this substation and distributes it to the substation controller. The specific substation is encoded in the CWL by the scope restriction. Using engineering information that is available at the substation controller, the substation controller creates device-specific certificate whitelists for the field devices, i. e., intelligent energy devices (IED), of the substation. The device-specific certificate whitelists are configured by the substation controller on the differend IEDs.

An alternative approach would be to compile a CWL for a substation, and to distribute this CWL to all components in the substation via the substation controller. Through the engineering information, each IED will only communicate with other IEDs by means of the engineering data and the CWL. This means that the access control decision is made by an IED by checking both the CWL and the engineering information. This saves the additional effort for creating device specific CWLs, but has the disadvantage that each IED needs to search a larger CWL, and has to check two pieces of configuration information separately. It is a validation perfomance decision which approach is more appropriate in a target environment. The generic definition of CWLs allows for both approaches.

A further usage scenario for certificate whitelisting within energy automation systems would be integration of decentralized energy resources. Here, a smart grid operator could realize a (managed) certificate pinning by using certificate whitelists. A smart grid operator would define which certificates are acceptable by including these certificates in a whitelist. Thereby, the smart grid operator would use certificate whitelists to restrict the set of certificates issued by a larger PKI. The possibility to misuse broken certificates or CAs is reduced as the set of accepted certificates is limited.

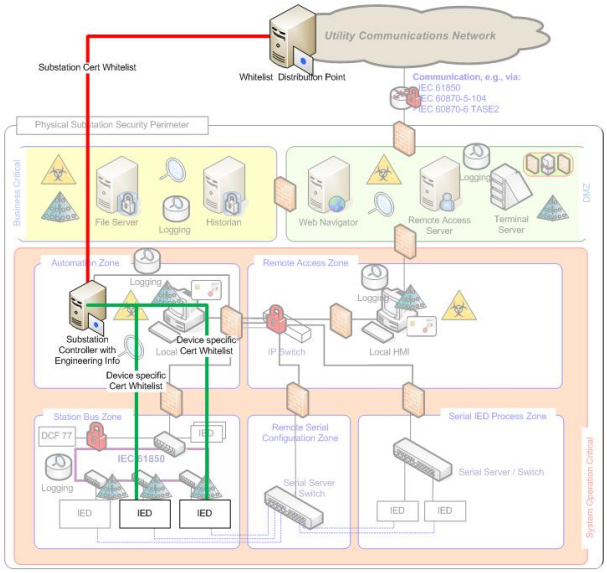


Fig. 5. Managed Certificate Whitelisting in Energy Automation Substations

VI. Conclusion and Outlook

Explicitly designating trusted certificates in certificate whitelists has been recently put forward within standardization for industrial energy automation communication [15]. It promises to provide a cost-efficient, easily deployable, and operable approach for digital device certificates even if self-signed certificates are used. It is intended for mid-sized industrial automation domains, while providing a migration path to more flexible PKI and access management structures. It allows in particular to avoid the usage of simple manually configured pre-shared secrets, that would be difficult to migrate to more complex and managed security infrastructures that are expected to be advantageous for large scale deployments.

The usage of certificate whitelisting can be supported with automatic whitelist generation and distribution. A format for certificate whitelists is currently being standardized to provide an interoperable format. Specific extensions can mark a certificate explicitly for being used only in combination with a certificate whitelist. Several additional extensions may be introduced. It may be possible to indicate usage restrictions within a certificate whitelist associated with a certain certificate entry. This could be used to limit the authorized usage of a certificate on a certificate-by-certificate basis. Certificate

 ISBN: 978-1-61208-376-6

whitelists may be encoded efficiently by including matching criteria of included certificates. Alternatively to the explicit enumeration of certificates, a filter can be included in a certificate whitelist that defines matching criteria of included certificates, i. e., that defines required properties of certificate fields. A Bloom filter [19] may be used, combined with a check on false match. Bloom filters are a probabilistic data structure for membership queries which allow for an efficient encoding, but for which a wrong positive match may occur. As the set of all issued certificates is known in typical usage scerarios, a checking for a false match is easily possible. Also, certificates can be designated within a whitelist. Also, a PKI gateway can be deployed for secure interworking with external network domains using a standard public key infrastructures.

Also, the logical combination of multiple certificate whitelists is possible in general. A combination of certificate whitelists may be advantageous for instance in an inter-substation communication scenario. Here, a first certificate whitelist may be provided for the substation internal communication, and a second one for inter-substation communication. The final certificate whitelist for each purpose may be defined by a logical combination of whitelists to ease the certificate whitelist administration and the handling for the field device. This might be done by logical OR, AND, or XOR combinations of the certificate whitelists. This logical combination can be realized in different ways: The field devices themselves can check against multiple certificate whitelists. A logical expression is configured that defines the logical combination of the certificate whitelists to be applied. As the defined certificate whitelist structure shown in Fig. 2 allows the encapsulation of multiple certificate whitelists within a single data structure, an enhancement of this data structure could indicate the logical combination of the whitelist entries using the extension option. A further alternative would be the preparation of device specific certificate whitelists by a centralized infrastructure component that determines the result of the logical combination of different certificate whitelists before distributing the actual certificate whitelist to the end points. This puts more effort on the centralized component, but keeps the effort low for the field device. The assumption here is that the certificate whitelist for a single endpoint is rather short compared to substation wide certificate whitelists containing all allowed (engineered) combinations of communication associations. The structure defined in Fig.2 also allows to use different matching criteria for the certificate. While the serial number and issuer or the fingerprint are straight forward, the utilizatin of the public key fingerprint provides another degree of freedom. This approach allows even for updating certificates (assumed the public key stays the same) without changing the CWL. This decouples the certificate life cycle management from the access security policy management of certificates in automation environments.

REFERENCES

[1] IEEE 802.11, "IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems, Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications." [Online]. Available: http://standards.ieee.org/about/get/802/802.11.html [accessed: 2014-09-01]

[2] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Barners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," 1999, Internet Request for Comments RFC2696. [Online]. Available: https://tools.ietf.org/html/rfc2696 [accessed: 2014-09-01]

[3] IEEE 802.1X-2010, "IEEE Standard for Local and metropolitan area networks–Port-Based Network Access Control," . [Online]. Available: http://standards.ieee.org/findstds/standard/802.1X-2010.html [accessed: 2014-09-01]

[4] C. Kaufmann, P. Hoffman, Y. Nir, and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)," Sep. 2010, Internet Request for Comments RFC5996. [Online]. Available: https://tools.ietf.org/html/rfc5996 [accessed: 2014-09-01

[5] S. Kent, and K. Seo, "Security Architecture for the Internet Protocol," Dec. 2005, Internet Request for Comments RFC4301. [Online]. Available: https://tools.ietf.org/html/rfc4301 [accessed: 2014-09-01]

[6] T. Ylonen, and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture," Jan. 2006, Internet Request for Comments RFC4251. [Online]. Available: https://tools.ietf.org/html/rfc4251 [accessed: 2014-09-01]

[7] Netscape, "SSL 3.0 specification," Nov. 1996. [Online]. Available: http://web.archive.org/web/20080208141212/http://wp.netscape.com/eng/ssl3/ [accessed: 2014-09-01]

[8] T. Dierks, and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," Aug. 2008, Internet Request for Comments RFC5246. [Online]. Available: https://tools.ietf.org/html/rfc5246 [accessed: 2014-09-01]

[9] ITU-T X.509, "X.509 Information technology – Open Systems Interconnection -– The Directory: Public-key and attribute certificate frameworks," 2012, version 3 corrigendum 3. [Online]. Available: http://www.itu.int/rec/T-REC-X.509-201210-S!Cor3/en [accessed: 2014-09-01]

[10] D. Cooper, S. Santesson, S. Farrel, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," May 2008, Internet Request for Comments RFC5280. [Online]. Available: https://tools.ietf.org/html/rfc5280 [accessed: 2014-09-01

[11] J. Buchmann, E. Karatsiolis, and A. Wiesmaier, "Introduction to Public Key Infrastructures," 2013.

[12] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," Jan. 2013, Internet Request for Comments RFC6960. [Online]. Available: https://tools.ietf.org/html/rfc6960 [accessed: 2014-09-01]

[13] eTutorials.org, "C/C++ Secure Programming – Chapter 10.9 Using a Whitelist to Verify Certificates," 2014, eTutorials.org. [Online]. Available: http://etutorials.org/Programming/secure+programming/ [accessed: 2014-09-01]

[14] Microsoft, "Digital Rights Management License Protocol – Retrieving Revocation Data from the Enrollment Server," 2014. [Online]. Available: http://msdn.microsoft.com/en-us/library/dd644914.aspx [accessed: 2014-09-01]

[15] ISO/IEC 62351, "Power systems management and associated information exchange Data and communication security," 2014, IEC TC57. [Online]. Available: http://tc57.iec.ch/index-tc57.html [accessed: 2014-09-01]

[16] E. Rescorla, "HTTP Over TLS," 2000, Internet Request for Comments RFC2818. [Online]. Available: https://tools.ietf.org/html/rfc2818 [accessed: 2014-09-01]

[17] OPC Foundation, "OPC Unified Architecture Specification Part 1: Overview and Concepts, Release 1.02," Jul. 2012. [Online]. Available: http://www.opcfoundation.org/ua/ [accessed: 2014-09-01]

[18] ISO/IEC 61850, "IED Communications and Associated Data Models in Power Systems," 2014, IEC TC57. [Online]. Available: http://tc57.iec.ch/index-tc57.html [accessed: 2014-09-01]

[19] Wikipedia, "Bloom Filter." [Online]. Available: http://en.wikipedia.org/wiki/Bloom_filter [accessed: 2014-09-01]

 ISBN: 978-1-61208-376-6

Challenges for Evolving Large-Scale Security Architectures

Geir M. Køien

Institute of ICT
Faculty of Engineering and Science
University of Agder, Norway
Email: geir.koien@uia.no

***Abstract*—In this paper, we conduct an informal analysis of challenges that face evolving large-scale security architectures. The 3rd generation partner project (3GPP) mobile systems is our example case and we shall investigate how these systems have evolved and how the security architecture has evolved with the system(s). The 3GPP systems not only represent a truly long-lived system family, but are also a massively successful system family, serving billions of subscribers. What once was an auxiliary voice-based infrastructure has evolved to become a main (and thereby critical) information and communications technology (ICT) infrastructure for billions of people. The 25+ years of system evolution has not all been a linearly planned progression and the overall system is now clearly also a product of its history. Our ultimate goal is to capture some of the essence of security architecture evolution for critical ICT system.**

Keywords–Evolving Security; System Security; Security Architecture; Long-term security planning.

I. INTRODUCTION

In this paper, we carry out a case-study analysis of some of the challenges that evolving large-scale security architectures must meet. The object of our study, the 3GPP systems, has gradually become important, all-encompassing and pervasive on a global scale. The systems have emerged to become a critical ICT infrastructure and this makes the system robustness and security a concern for society-at-large.

A. The 3GPP System Context

The first 3GPP system is the second generation (2G) Global System for Mobile communications (GSM), developed in the mid/late 1980ies. Originally, GSM only featured circuit-switched (CS) services, but was later adapted to also include packet-switched (PS) services through the General Packet Radio Service (GPRS) extension. With the new millennium came the third generation (3G) Universal Mobile Telecommunications System (UMTS), which natively features both CS and PS services. From around 2010 we also have the fourth generation (4G) Long-Term Evolution (LTE) system, which is a broadband PS-only system. LTE is further developed into LTE-Advanced (LTE-A).

1) Principal Parties: From a subscriber perspective, the system can be described with three types of principal parties.

- The Home Public Land Mobile Network (HPLMN)
- The Visited Public Land Mobile Network (VPLMN)
- The subscriber/user (USER)

These parties are legal entities, and the relationships are determined by contractual agreements. A national telecom regulator will also be involved, in addition to external service providers. One may also add intruders to the list. The external service providers usually have little influence on how the networks operate and so we exclude those for further discussion. Likewise, in this context, we do not see a need for including virtual mobile network operators (VMNOs).

2) System Development: The 3GPP system specifications are developed by the 3GPP, but ratification is done by the organizational partners (formal standardization bodies). As with other such groups, the 3GPP is contribution driven. This has an important impact on what is actually being done. The impact is noticeable when it comes to priorities and efforts spent. Early on, when GSM/GPRS was specified, the operators took considerable responsibility and led many of the efforts. Subsequently, the vendors have taken over more and more of this work. The impetus to carry out work is clearly related to the business potential the work has. Unfortunately, investments in security functions seldom look like a good business proposition prior to an incident.

The 3GPP differentiates between *mandatory for implementation* and *mandatory for use*. That is, a feature may be mandatory to be implemented by the vendors if they want compliance with a system release. At the same time, the operators may freely disregard the feature if they want. Other functions may be mandatory both to develop and deploy.

3) License to Operation and Regulatory Requirements: Cellular systems operate in licensed bands and are subject to regulatory requirements. These requirements include support for lawful interception (LI) and emergency call (EC). The last decade we have also had anti-terrorist measures such the EU Data Retention Directive (DRD) [1].

B. Brief Introduction to 3GPP Systems

1) 2G – GSM and GPRS: The GSM and GPRS systems are the 2G systems. It is common to see monikers like 2.5G used for GPRS, and 2.9G used for GPRS with Enhanced Data rates for Global Evolution (EDGE). The main GSM features are mobility, speech and text messaging. GPRS is an overlay system to GSM. It features two additional core network nodes and provides PS support. With EDGE (new codecs) it provides up to 236 kbps data-rate. There is also an "Evolved EDGE" extension on the horizon, with yet higher data-rates. The 2G-based radio access network is called GSM EDGE Radio Access Network (GERAN).

 ISBN: 978-1-61208-376-6

2) 3G – UMTS (incl. High-Speed Packet Access (HSPA)): The UMTS system was finalized in late 1999 and is a combined CS/PS system. It can readily achieve >10 Mbps data-rates (w/max. rates >100 Mbps downlink). The system is a mix of GSM/GPRS technology and protocols and, increasingly, IP-based protocols and technology. The radio access network is called the Universal Terrestrial Radio Access Network (UTRAN).

3) 4G – LTE and LTE-A: The LTE systems are designed as all-IP networks (AIPN) and features true mobile broadband. The core network is fully IP based and there are no CS components to be found. The radio system is highly advanced and provides true broadband services. The radio base-stations, called eNB, are logically mesh connected. There are no longer any controllers in the access network (E-UTRAN). The VPLMN mobility functions are carried out by the mobility management entity (MME) server.

C. Paper Layout

In Section II, we briefly outline the security of the 3GPP systems. In Section III, we attempt to capture some of the triggers for changing the security architecture. Then we proceed in Section IV, with observations regarding successful systems, and for security and cryptography in those systems. We also include observations regarding the typical intruders. In Section V, we try to learn from the lessons and provide some advice. Finally, we sum up our effort and provide some concluding remarks in Section VI.

II. Security in the 3GPP Systems

In this Section, we provide a (necessarily) short description of the main features of the 3GPP security provisions.

A. 2G Security

There is no well-defined security architecture per se in the 2G systems. The main security specification was technical specification (TS) 03.20 "Security-related network functions", which subsequently has been transposed into TS 43.020 [2]. It defines the identity- and location privacy scheme, the entity authentication protocol and the smart-card based security functions. It also outlines the over-the-air cipher function.

1) Background and Requirements: In the voice-only 1G systems one had experienced charging fraud and impersonation fraud. Two distinct types of attacks quickly came into focus: **a)** Eavesdropping was a big problem as the analogue voice channel was unprotected and easy to listen-in on. **b)** Faking the call setup signaling, which was digital, was quite easy and could in principle be done by simply recording a setup sequence and then later replay it. The main priority for a fully digital system a la GSM was therefore to **a)** protect the over-the-air channel against eavesdropping, such that it would no longer be the weakest link, and **b)** provide credible subscriber authentication to avoid impersonation attacks.

2) The 2G Security Architecture: GSM security is based on a physical subscriber identity module (SIM). For portability reasons it was decided to use a smart-card. The SIM comprises both hardware and software functionality, and it contains the authentication and key agreement (AKA) functions (symmetric crypto). The SIM also contains the security credentials, like the permanent subscriber identity (IMSI) and the corresponding 128-bit authentication secret, called K_I in the 2G SIM. Figure 1 outlines the GSM security procedures.

The AKA protocol used is called GSM AKA, and it is a single-pass challenge-response protocol with a signed response (SRES). The challenge is a pseudo-random 128-bit RAND bit-field and the response is the 32-bit SRES element. The challenge-response part is dependent on an "authentication set" forwarding stage, in which the HPLMN forwards the authentication credentials to the VPLMN network. The protocol is run between the SIM and the visited network. This scheme is efficient and allows for fast and simple authentication of the subscriber as well as deriving a session key (the 64-bit K_C). The SIM features the A3 and A8 AKA interfaces, which are only found in the SIM and the home subscriber database (HLR). The original example implementation, called COMP128, is cryptographically broken [3], but still seems to be in use in many markets.

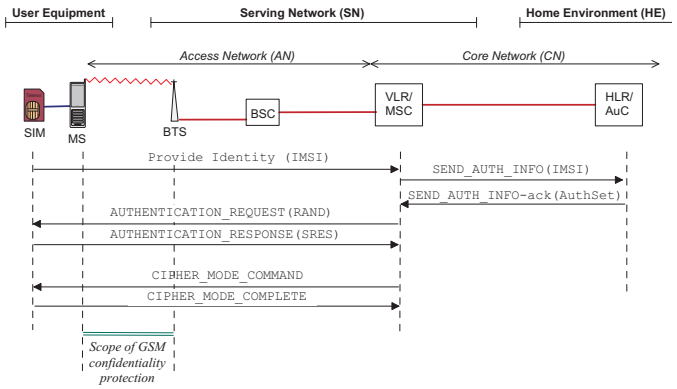


Figure 1: GSM security overview

Over-the-air encryption is by means of the A5 stream cipher family, which is located in the mobile phone and the base tranceiver station (BTS). There are several A5 versions available, but the original A5/1 is still the default and mandatory-to-deploy algorithm. It can easily be broken today by a dedicated attacker [4]. The A5/2 algorithm, which was explicitly designed to be weak (CoCom regulations), is officially deprecated. The A5/3 algorithm, which is based on the 3G KASUMI design, is the current best option for GSM, but rainbow table attacks still work since the algorithm is limited to 64-bit [5]. The A5 family is based around a 64-bit key, expect the new (and not deployed) A5/4 cipher, which is a 128-bit design based on the KASUMI algorithm. In GPRS one uses the GSM AKA protocol as-is, but here one uses the GPRS Encryption Algorithm (GEA) ciphers to protect the asynchronous packet transfers.

3) Omissions and Shortcomings: There are many obvious omissions and shortcomings to GSM security. This is not strange as the 2G systems do not have a security architecture as such; it is more akin to a collections or measures put together without well-defined requirements. The following list (derived in [6]) identifies some of the flaws. Even with all these flaws, the GSM/GPRS system has been a remarkably secure system. However, some 25 years down the line and the shortcomings have become serious liabilities. There are also a number of

 ISBN: 978-1-61208-376-6

implementations issues [7]. The list is not fair with regard to the threats found early on, but it is certainly valid now.

- One-way authentication is utterly inadequate
- Delegated authentication is naive trust-wise
- No inter-operator authentication
- No way to authenticate system nodes
- No uniqueness/freshness to challenges
- Unauthenticated plain-text transfer of security credentials
- Unprotected key transfer
- Missing key binding and too short keys
- Key refresh dependent of re-authentication
- Missing expiry condition on security context
- Weak A3/A8 functions and no key-deriving key structure
- Short A5 key stream cycle and key stream re-use
- Redundant and structured input to A5 (expand-then-encrypt)
- Highly redundant input to A5 (in signaling message)
- Protection coverage/range too short (only MS – BTS)
- Missing integrity protection
- Weak/inadequate identity/location privacy
- No core network control plane (signaling) security features
- No core network user plane protection
- No IP protection (GPRS)
- No mobile phone (MS) platform security

B. 3G Security

1) Background and Requirements: Security in the UMTS system is described briefly in [6, 8] and in considerable depth in [9]. The main security specification is TS 33.102 [10]. One also provided a "Security Objectives and Principles" [11] background document, as well as conducting a threats and requirements analysis [12]. One also introduced Network Domain Security (NDS), which includes IPsec profiles for use with 3GPP systems [13] and a standard set of public-key infrastructure (PKI) protocols and methods [14].

2) The 3G Security Architecture: The UMTS security architecture, depicted in Figure 2, is an important overhaul of the GSM security, yet the underlying system model remains much the same. Amongst the features are:

- New subscriber card (UICC) with security module (USIM)
- Introduction of 128-bit crypto primitives
- Improved two-way AKA algorithm (UMTS AKA)
- Introduction of core network protection (IP protocols)

Sadly, backwards compatibility concerns also dictated that the GSM SIM could still be used, which re-introduces many if not most of the 2G weaknesses.

3) The IP Multimedia Subsystem (IMS): IMS came with UMTS (Rel.5). We do not include IMS in our discussions as it is an optional service-level feature.

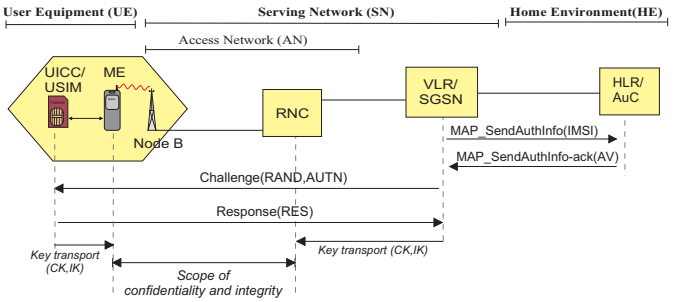


Figure 2: UMTS security

4) Omissions and Shortcomings: The 3G security is substantially better and more future proof than the 2G security, and one really has a security architecture. The architecture is by no means perfect or complete, but it does at least capture the main risks/threats and defines what one wants to protect. Completeness will always be an issue, but in the 3G systems we also have that there sometimes is a considerable mismatch between stated goal and what the mechanisms achieve. A case in point would be the identity/location privacy requirements, which does capture the problem well, but the mechanisms that should provide the necessary services are woefully inadequate. They are however *a)* exactly the same as for the 2G systems and *b)* they are intimately tied to the identity presentation scheme defined in the basic mobility management (MM) protocol machinery (discussed in [6, 15]). Making changes here would have been a major undertaking, and since there was considerable time pressure to complete the 3G standard, improvements to identity/location privacy simply did not happen (there were efforts investigating the possibilities during the Rel.99 design).

Many of the items on the 2G list of omissions and shortcomings are mitigated and resolved, but suffice to say that many of the 2G weaknesses were inherited or permitted through backwards compatibility requirements. Another main problem with 3G security is the limited scope.

C. 4G Security

1) Background and Requirements: The book "LTE Security" [16] is good and thorough introduction. The main security standard for LTE is TS 33.401 [17]. LTE and LTE-A are very similar with respect to the security architecture, which for historical reasons is called the "System Architecture Evolution (SAE)" security architecture. The term Evolved Packet System (EPS) is also used.

The radio access architecture changed significantly with LTE and this triggered large-scale changes to the whole system, including the security architecture. The security requirements were retained more or less as-is. For compatibility reasons and due to time constraints during the design phase, the UMTS AKA protocol was retained as a component of the EPS AKA protocol.

2) The 4G Security Architecture: The LTE security architecture has a lot in common with 3G security, but with some important changes. Amongst the LTE features are:

- UICC/USIM is retained and required

 ISBN: 978-1-61208-376-6

- Introduction of full key-deriving key hierarchy
- Session keys not dependent on re-authentication
- Auth. master key (K_{ASME}) bounded to VPLMN id.
- New session keys for every handover
- Separation of user plane and control plane protection
- Introduction of improved AKA algorithm (EPS AKA)

A welcome change is that backwards compatibility with GSM SIM is prohibited for access to E-UTRAN. UMTS AKA derived security contexts can be used (mapped) to LTE contexts. Figure 3 depicts the EPS key hierarchy, which is very different from the 2G/3G schemes. The new key derivations take place exclusively outside the UICC/USIM. This makes for a significant departure from previous practices.

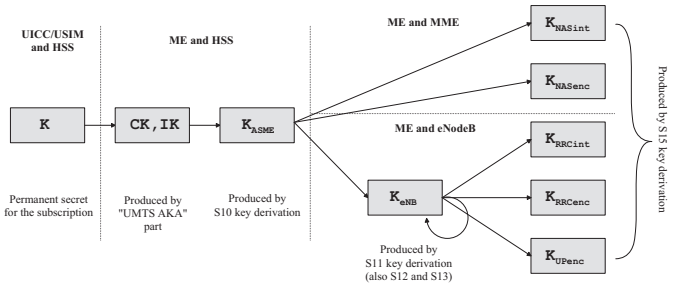


Figure 3: The EPS key hierarchy

3) Omissions and Shortcomings: The list of omissions and shortcoming is shorter for LTE, but there are also new threats. In a world of smart phones, it is obvious that 128-bit crypto on the access link may count for nothing if the mobile phone is infested with malicious Apps. Likewise, the networks are often hybrid systems, and it is common to have base stations that are 2G/3G/4G compliant. With different security levels and common hardware/software, it is clear that strong 4G protection may easily be offset with weak 2G/3G protection. For 4G this is quite important, as the mesh architecture means that all eNBs will be able to reach all other eNBs. Thus, one compromised eNB can reach all other eNBs in the network segment (which may span the entire operator network). It is also clear that many of the nodes, including the base station (BTS/NB/eNB) may be running commodity operating systems (OS). The chosen OS, likely a Linux variant, may be reasonably secure, but even a high-security OS will have weaknesses and must be properly managed to remain secure. Also, introduction of firewalls and intrusion detection systems will be required for these systems now. Server hardening is a must, and even so it is clear that not all attacks can be prevented. This means that prevention alone cannot be a viable future strategy.

The EPS security architecture does require the eNB to be secure, but the specification is not very specific [17]. It also has recommendations on use of firewalls, but the specification is quite vague on this subject too. For a greenfield 4G system, the security may be quite good at what the system provides, but the standard system does not do all it needs to do. Also, it is obvious that even though the user equipment (UE) normally is not owned or controlled by the network operator, the mobile devices must have a minimal level of protection. This is not only to protect the user, which a HPLMN should be interested in anyhow, but also to protect the network as a population of broadband devices could disrupt the access network. Distributed Denial-of-Service (DDoS) attacks would be but one possibility.

D. Architectural Oddities

One puzzling aspect of the 3GPP security architectures is that while identity presentation and entity authentication is fully standardized, there is no authorization mechanisms present. There are of course mechanisms to discriminate subscriber based on the type of subscription, but these schemes are not a feature of the security architecture.

Another aspect to be noted is that the subscriber identity that actually is authenticated, the IMSI, is basically a link layer identifier. Since there is only basic connectivity present at the link layer it may help explain why there never was any built-in authorization scheme in the 3GPP security architecture.

III. Evolving Security Architecture

A. Why Change the Security Architecture?

The short answer is that we need to change the security architecture because some of the premises for the original security architecture have changed. A slightly longer answer would revolve around the following aspects.

B. High-level change triggers

There are many high-level change triggers, amongst others:

- *Changes to the assets of the system*
  This could include changes to the value of the existing assets, inclusion of new assets or removal of assets.
- *Changes in the threats towards the assets*
  This includes assets exposure, new intruders, new intruder capabilities. For new assets it could also include missing or mismatched protection.
- *Changes to the system context*
  The system may initially have played a limited role, but may have evolved into something more.

C. Evolution aspects

Large-scale long-lived systems cannot remain as static objects for long. Instead, they must be dynamic and adapt to changing environments.

- *Evolving Target System*
  If the target system changes, then this will likely affect the security architecture. Still, the nature of the change may be such that it does not trigger a need for updating the security architecture.
- *Evolving Security Architecture - Externally triggered*
  The security architecture may need updates and modifications due to external circumstances, or even completion of planned features that were not initially fully specified. Changes in the threats towards the assets, the exposure of the assets, and the number of users will

 ISBN: 978-1-61208-376-6

also affect the system. It could also involve changing trust-relationships and changes to value of the assets.

- *Evolving Security Architecture - Internally triggered*
  Change in use. The internal circumstances would encompass altered or increased use, which would include changes to the assets of the system.
- *Security Evolution History*
  An evolving system is obviously a product of its history. Decisions taken during the design of GSM still have an impact on LTE. For instance, the basic identity presentation scheme essentially remains the same for LTE as for GSM [18, 19].
- *Societal Impact*
  When a system reaches certain thresholds it will take on a new role. It enters a state of criticality to society and will become an object of regulatory interest. The critical infrastructure (CI) requirements, will focus on system survival and service availability rather than security and privacy for the individual.
- *Privacy*
  Privacy requirements may not have mattered too much for a small system with few users back in the early 1990ties. Today privacy requirements are often mandated by laws and regulations.

IV. Assumptions Regarding Systems, Security and Cryptographic Characteristics

The following set of assumptions not all be true for all systems, but we advocate assuming that they are true.

A. Assumptions about Successful Systems

We assume that when people start to design a system they intend it to be successful. Thus, they must therefore take the above into account in their design. Our high-level assumptions about a successful system:

1) It will outlive its intended lifetime (and design)
2) It will have many more users then originally intended
3) It will need to scale its services cost-effectively
4) It will become highly valuable (many/valuable assets)
5) It will outlive its base technologies
6) It may become a critical system (company, organization)
7) It may become a critical infrastructure (society-at-large)
8) It will spawn unsuccessful branches/features
9) It will have to deal with multi-vendor cases
10) It will need to operate with multiple releases in place
11) It must encompass all of operations & maintenance too
12) It will be subject to regulatory interventions

B. Assumptions about System Security

Our assumptions about a long-lived security architecture:

1) The assets will change (value/number/types)
2) The principal parties will change and multiply
3) The threats will change
4) Trust models will fail (and/or become outdated)
5) Trust will be betrayed
6) Risk evaluations will be outdated
7) The weaknesses, vulnerabilities and exposure will change
8) The intruders will become more powerful and proliferate
9) Attacks will only be better over time
10) There will be security incidents
11) Scalability in security mechanisms will be decisive
12) No single security scheme or approach will be sufficient
13) Effective and efficient defense-in-depth will be needed
14) Pro-active security protection will not be sufficient
15) Re-active security will be very important (detect & respond)
16) Ability to handle large incidents will be required
17) Mitigation and recovery must be supported
18) Pervasive resilience and robustness is required
19) Autonomous sub-system response will become important
20) There will be security architecture omissions
21) There will be security compatibility issues (multi-vendor)
22) There will be security compatibility issues (multi-release)
23) Fixing minor security wholes can take a very long time
24) Fixing the security architecture take years (next generation)
25) Security management will be crucial
26) Security configuration management is crucial
27) Security migration methods should be built-in
28) Privacy will become ever more important

C. Assumptions about Cryptographic Solutions

Our assumptions related to cryptographic solutions:

1) The cryptographic base functions must be future-proof
2) Cryptographic primitives will be broken (or too weak)
3) Key sizes will be changed
4) Security protocols will be broken (or too weak)
5) Cryptographic parameters will need to be negotiated (securely)
6) Cryptographic primitives will need to be revoked
7) Implementations will contain weaknesses
8) Management of cryptographic elements will be crucial

It is clear that the basic boot-strapping fundament must be very solid. This minimal base is what you will depend on if you need to boot-strap new security solution and new cryptographic primitives in the rest of the security architecture. It needs to contain enough to support boot-strapping and it needs to be future-proof. Efficiency is *not* a main priority here.

D. The Scalability War

The classical Dolev-Yao Intruder (DYI) is not the most realistic intruder [20]. Real intruder will use any available means (subversion, physical intrusion, tricking the principals), ultimately being as powerful as a DYI. There is a reasonably body of papers detailing various intruder model, but suffice to say that a modern CI system must be able to handle **all** types of intruders. And many of them! This essentially means that the system *must* have efficient as well as effective protection, and that mechanisms that do not scale well, compared to intruder capabilities, will be doomed to fail in the long run.

Our assumptions related to scalability and efficiency:

1) Security scalability will be a major concern
2) Efficiency is highly important
3) Effectiveness is imperative for core mechanism
4) Auxiliary defense-in-depth solution are needed
5) Avoid specific-attack measures if at all possible
6) Security management must scale well

Assumption three and four are apparently somewhat at odds, but in the end assumption three can be supported given that these means are complementary and cost-effective. See

 ISBN: 978-1-61208-376-6

also considerations about the economy of attacks and defenses outlined in [21], This indicates that for broad sweeping attacks, even quite weak mechanisms may successfully thwart the attacks. Measures that are only effective for one specific attack should be avoided.

E. Other Concerns

1) Passive Regulatory Authorities: One main concern is that the regulatory authorities generally are quite passive with regard to security requirements. This is apparent for the cellular system and regulations concerning the operators. The 3GPP standards are by no means perfect or complete, but it is still the case that many of the standardized and recommended security mechanisms are not deployed in the networks. The regulatory authorities are generally more reactive than proactive, unless they have a clear political mandate to be stringent. One should also be concerned about regulations just subsequent to a major public incident, since it is likely that the urge to "do something" is strong while it is also likely that one focuses narrowly on details. One may end up with *security theater*, as coined by Schneier [22].

Part of this problem is a that one sometimes ends up with a lot of attention to correct and strengthen unimportant features. To do something right is not enough, one must also do the right thing.

2) False Security: Security theater may over time develop into the more elaborate *cargo cult security* type of deception. Then the main functions and mechanisms may all be there (or mimicked closely), but with some vital part missing or done completely wrong. Cargo cultism is defined by "perfect form", but it simply does not work as intended. Feynman has an amusing description of "cargo cult science" that nicely illustrates the principles [23]. Since security can be very difficult to get right and to verify, cargo cult security may look like the real deal.

3) Security Testing and Security Configuration: In [7] the authors clearly also demonstrate that not only is not all security options exercises, but that, unsurprisingly, there are implementation weaknesses and vulnerabilities. The ASMONIA project provides many more examples of weakness, vulnerabilities and risks facing a mobile system [24]. The ASMONIA project published a lot of useful documents for operators wanting to improve their security level. The documents also include advice and methods for how to test the security. The EU body ENISA provides a lot of useful security-related input, but generally have no mandate to impose security [25]. When it comes to IP network security and server security there is a large body of standards and methods for how to design and test security hardening [26–29]. There are also various checklists available [30].

V. Lessons Learned

A. Verify Assumptions

One must verify assumption about the system and the security periodically or when there are substantial changes to the system. That is, an audit is called for to verify assumptions about the assets, the principal entities, trust relationships etc. Security policies will be affected by changes to these assumptions. This is a process oriented task that must take place both for the design phase and for the deployed system(s).

B. Rock Solid Bootstrapping Security

There needs to be a rock solid fundament that will be secure for the foreseeable future. The smart-card has served this purpose in the 3GPP systems on the subscriber side. The smart-card is not tamper-proof, but it has successfully served as a high-trust platform.

C. Planned Deprecations

A scalable and evolving system must be able to handle deprecation of almost all cryptographic algorithm, security protocols and security services. The deprecation, needless to say, must be conducted in a secure manner. Backwards compatibility requirements and fallback solutions must be handled in a secure way.

D. Negotiable and Adaptable

Given that one must plan for deprecation of security features/services, one must also plan how to negotiate new features/services. This feature must be built-in and have high assurance. Adaptation may be necessary to account for local requirements, but is vital that adaptations must be fully compliant with a well-defined security policy.

E. Proactive & Reactive Security

Basic security functionality to identify and authenticate principals and entities is necessary, but not sufficient. Adding authorization, protected storage and protect communication is also necessary, but still not sufficient. More may be added, but in the end it is impossible to fully secure the system. This means that one must handle and deal with incidents. There is therefore a clear need for intrusion detection and response systems, to deploy firewalls, anti-virus protection, secure backups, secure audit trails etc. The reactive measures must be included in the overall system security plans and subject to revisions as need be.

F. Stability, Resilience and Recovery

System integrity is imperative to ensure a stable and resilient system. System integrity is a system-level characteristic and does not preclude partial or local failures. What is imperative is to prevent the failures to scale. Failures, whether man-made intentional or unintentional, cannot entirely be prevented. Procedures that support mitigation and recovery must be an integral part of the overall system security plan.

G. Configuration Management

Proper planned configuration management, which must include security functionality, is an absolute necessity.

H. Privacy Matters

Privacy is one feature that must be accounted for in all systems that include human users or any kind of data pertaining to humans. This must be planned for from the design phase and handled in all phases of system deployment.

 ISBN: 978-1-61208-376-6

VI. Concluding Remarks

The results in this paper cannot be said to be fully supported by the evidence provided in this paper (or in the referenced papers). They are neither rigorous nor complete. This is to be expected for such a complex issue. Thus, while the results may be valid and true, they will hardly be complete and not always necessary either. That is, the usual "necessary and sufficient" conditions are not really there. Still, experience and empirical evidence should not be discounted, and we advocate that the lessons learned are taken into account, not as mathematical axioms, but inputs to be considered. We therefore recommend that scalable evolving security architectures should be designed with these assumption as background.

In this paper, we have outlined the 3GPP security architecture as it has evolved over more than 25 years. From being an auxiliary service for the few, it has grown to literally cater to billions of subscribers, and the number and types of services provided has changed dramatically over the years. The use-patterns of these systems has changed as well. All in all, there has been a complete transformation of almost all aspects of these systems. During this process, the security architecture has evolved with the system and the changing system context, though not without some noticeable failures and a growing number of security problems.

We have argued that to achieve scalable security architectures that are able to evolve over time, one needs to take into account the fact that almost all assumption one initially had will become false or moot. This means that adaptability and ability to support changes is crucial. This is important in a world where the internet-of-things (IoT) landslide is about to happen and where the systems will be ever more important.

In the wake of the Snowdon revelations, it is also clear that cyber-security is under constant pressure, and while we do not want to over-state the Snowdon case per se, it should be clear that the cyber-war methods will (over time) become available to many organizations and individuals. So we need to learn how to cope with this and do so fast.

References

[1] European Parliament/European Council, "Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC," EU, Directive 24/EC, 2006.

[2] 3GPP, TS 43.020, "Security related network functions," 3GPP, France, TS 43.020 (2G), 2014.

[3] J. R. Rao, P. Rohatgi, H. Scherzer, and S. Tinguely, "Partitioning attacks: or how to rapidly clone some gsm cards," in Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on. IEEE, 2002, pp. 31–41.

[4] M. Kalenderi, D. Pnevmatikatos, I. Papaefstathiou, and C. Manifavas, "Breaking the gsm a5/1 cryptography algorithm with rainbow tables and high-end fpgas," in Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on. IEEE, 2012, pp. 747–753.

[5] P. Papantonakis, D. Pnevmatikatos, I. Papaefstathiou, and C. Manifavas, "Fast, fpga-based rainbow table creation for attacking encrypted mobile communications," in Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on. IEEE, 2013, pp. 1–6.

[6] G. M. Køien, Entity authentication and personal privacy in future cellular systems. River Publishers, 2009, vol. 2.

[7] F. van den Broek, B. Hond, and A. Cedillo Torres, "Security Testing of GSM Implementations," in Engineering Secure Software and Systems, ser. Lecture Notes in Computer Science, J. Jürjens, F. Piessens, and N. Bielova, Eds. Springer International Publishing, 2014, vol. 8364, pp. 179–195.

[8] G. M. Køien, "An introduction to access security in UMTS," Wireless Communications, IEEE, vol. 11, no. 1, Feb 2004, pp. 8–18.

[9] V. Niemi and K. Nyberg, UMTS Security. John Wiley & Sons, 2003.

[10] 3GPP, TS 33.102, "3G Security; Security architecture," 3GPP, France, TS 33.102 (3G), 2014.

[11] 3GPP, TS 33.120, "Security Objectives and Principles," 3GPP, France, TS 33.120 (3G), 2001.

[12] 3GPP, TS 21.133, "3G security; Security threats and requirements," 3GPP, France, TS 21.133 (3G), 2001.

[13] 3GPP, TS 33.210, "3G security; Network Domain Security (NDS); IP network layer security," 3GPP, France, TS 33.210 (NDS/IP), 2012.

[14] 3GPP, TS 33.310, "Network Domain Security (NDS); Authentication Framework (AF)," 3GPP, France, TS 33.310 (NDS/AF), 2014.

[15] G. M. Køien, "Privacy enhanced cellular access security," in Proceedings of the 4th ACM workshop on Wireless security. ACM, 2005, pp. 57–66.

[16] D. Forsberg, G. Horn, W.-D. Moeller, and V. Niemi, LTE security. John Wiley & Sons, 2012, vol. 1.

[17] 3GPP, TS 33.401, "3GPP System Architecture Evolution (SAE); Security architecture," 3GPP, France, TS 33.401 (3G), 2014.

[18] G. M. Køien, "Privacy enhanced mutual authentication in LTE," in Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on. IEEE, 2013, pp. 614–621.

[19] G. Køien, "Mutual entity authentication for LTE," in Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International. IEEE, 2011, pp. 689–694.

[20] D. Dolev and A. C. Yao, "On the Security of Public-Key Protocols," IEEE Transactions on Information Theory, vol. 29, no. 2, 3 1983, pp. 198–208.

[21] D. Florêncio and C. Herley, "Where do all the attacks go?" in Economics of Information Security and Privacy III. Springer, 2013, pp. 13–33.

[22] B. Schneier, "Beyond fear," Copernicus Book, New York, 2003.

[23] R. P. Feynman, "Cargo cult science," in Surely You're Joking, Mr. Feynman, 1st ed. W. W. Norton, 1985, Originally a 1974 Caltech commencement address.

[24] "The ASMONIA project," See www.asmonia.de, 2014.

[25] "ENISA - European Union Agency for Network and Information Security," See www.enisa.europa.eu/, 2014.

[26] K. Scarfone, W. Jansen, and M. Tracy, "Guide to General Server Security," NIST, Gaithersburg, MD 20899-8930, Special Publication 800-123, 2008.

[27] Z. Anwar, M. Montanari, A. Gutierrez, and R. H. Campbell, "Budget constrained optimal security hardening of control networks for critical cyber-infrastructures," International Journal of Critical Infrastructure Protection, vol. 2, no. 1, 2009, pp. 13–25.

[28] R. Dewri, I. Ray, N. Poolsappasit, and D. Whitley, "Optimal security hardening on attack tree models of networks: a cost-benefit analysis," International Journal of Information Security, vol. 11, no. 3, 2012, pp. 167–188.

[29] R. Dewri, N. Poolsappasit, I. Ray, and D. Whitley, "Optimal security hardening using multi-objective optimization on attack tree models of networks," in Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 204–213.

[30] NIST, "Security configuration checklists program," See http://csrc.nist.gov/groups/SNS/checklists/, 2014.

 ISBN: 978-1-61208-376-6

A Backtracking Symbolic Execution Engine with Sound Path Merging

Andreas Ibing

Chair for IT Security
TU München, Germany
Email: andreas.ibing@tum.de

***Abstract*—Software vulnerabilities are a major security threat and can often be exploited by an attacker to intrude into systems. One approach to mitigation is to automatically analyze software source code in order to find and remove software bugs before release. A method for context-sensitive static bug detection is symbolic execution. If applied with approximate path coverage, it faces the state explosion problem. The number of paths in the program execution tree grows exponentially with the number of decision nodes in the program for which both branches are satisfiable. In combination with the standard approach using the worklist algorithm with state cloning, this also leads to exponential memory consumption during analysis. This paper considers a source-level symbolic execution engine which uses backtracking of symbolic states instead of state cloning, and extends it with a sound method for merging redundant program paths, based on live variable analysis. An implementation as plug-in extension of the Eclipse C/C++ development tools (CDT) is described. The resulting analysis speedup through path merging is evaluated on the buffer overflow test cases from the Juliet test suite for static analyzers on which the original engine had been evaluated.**

Keywords–Static analysis; Symbolic execution.

I. INTRODUCTION

Software vulnerabilities like, e.g., buffer overflows can in many cases be exploited by an attacker for remote code execution. Automated bug detection during software development and for releases are a main component of application security assurance.

Symbolic execution [1] is a static program analysis method, where software input is regarded as variables (symbolic values). It is used to automatically explore different paths through software, and to compute path constraints as logical equations (from the operations with the symbolic input). An automatic theorem prover (constraint solver) is then used to check program paths for satisfiability and to check error conditions for satisfiability. The current state of automatic theorem provers are Satisfiability Modulo Theories (SMT) solvers [2], the standard interface is the SMTlib [3]. An example state-of-the art solver is [4].

Automatic analysis tools which rely on symbolic execution have been developed for the source-code level, intermediate code and binaries (machine code). Available tools mostly analyze intermediate code, which exploits a small instruction set and certain independence of programming language and target processor. Examples are [5] and [6], which analyzes LLVM code [7]. An overview of available tools is given in [8][9][10]. Symbolic execution on the source-code level is also interesting for several reasons. An intermediate representation loses source information by discarding high-level types and the compiler lowers language constructs and makes assumptions about the evaluation order. However, rich source and type information is needed to explain discovered bugs to the user [11] or to generate quick-fix proposals. An example of a source-level symbolic execution engine for C/C++ is [12], which uses the parser and control flow graph (CFG) builder from Eclipse CDT [13].

During symbolic execution, the engine builds and analyzes satisfiable paths through programs, where paths are lists of CFG nodes. Always restarting symbolic execution from the program entry point for different, partly overlapping program paths (path replay) is obviously inefficient. The standard approach is therefore the worklist algorithm [14]. Symbolic program states of frontier nodes (unexplored nodes) of the program execution tree are kept in memory, and at program branches the respective states are cloned. The reuse of intermediate analysis results with state cloning has the downside of being memory-intensive. [5] uses state cloning with a recursive data structure to store only state differences. Another approach for engine implementation is symbolic state backtracking [12]. It keeps only the symbolic program states along the currently analyzed program path in memory (stored incrementally with single assignments) and avoids the inefficiency of path replay as well as the exponential memory consumption of state cloning.

The program execution tree grows exponentially with the number of decisions in the program for which both branches are satisfiable. Straight-forward application of symbolic execution with approximate path coverage (where the number of unrolled loop iterations is bounded) is therefore not scalable. This is often called the path explosion problem. In [15] it is noted that program paths can be merged when the path constraints differ only in dead variables, because further path extension would have the same consequences for the paths. It presents an implementation which extends [5]. This implementation uses a cache of observed symbolic program states and introduces a type of live variables analysis which it calls read-write-set (RWSet) analysis.

Interesting properties of bug detection algorithms are soundness (no false negative detections) and completeness (no false positives). Because a bug checker cannot be sound and complete and have bounded runtime, in practice bug checkers are evalutated with measurement of false positive and false negative detections and corresponding runtimes on a sufficiently large bug test suite. The currently most comprehensive

 ISBN: 978-1-61208-376-6

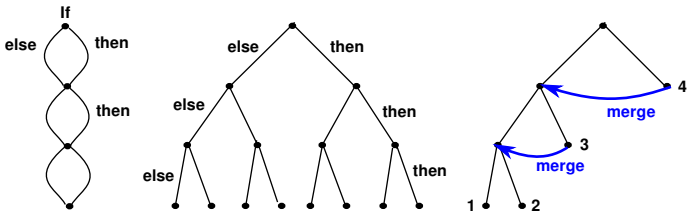


Figure 1. Sequence of three decisions and corresponding branches (left); the execution tree under the assumption that all branches are satisfiable splits into $2^3 = 8$ leafs (middle); path merging folds the execution tree (right).

C/C++ bug test suite for static analyzers is the Juliet suite [16]. Among other common software weaknesses [17] it contains buffer overflow test cases. In order to systematically measure false positives and false negatives, it contains both 'good' and 'bad' functions, where 'bad' functions contain a bug. It further combines 'baseline' bugs with different data and control flow variants to cover the languages grammar constructs and to test the context depth of the analysis. The maximum context depth spanned by a flow variant is five functions in five different source files.

This paper develops and evaluates a sound path merging method in a source-level backtracking symbolic execution engine. The implementation extends [12]. The remainder of this paper is organized as follows. Section II describes the design decisions. Section III gives an overview of the implementation in Eclipse CDT. Section IV presents results of experiments with buffer overflow test cases from the Juliet suite. Section V discusses related work and section VI then discusses the presented approach based on the results.

II. Merge Points and Context Cache

A. Dead and live variables

Paths can be merged without any loss in bug detection accuracy when the path constraints differ only in dead variables. The detection of such merge possibilities requires a context cache at potential merge points. Also required is a way to detect dead variables and to filter them from the path constraint. Potentially interesting merge points are therefore program locations where the sets of dead and live variables change. Such points are function start and function exit and after scope blocks like `if` / `else` or `switch` statements and loops.

B. Design decisions

The idea of merging program paths during symbolic execution is illustrated in Figure 1. The left of the figure shows a control flow with a sequence of three decisions and corresponding branches. For the assumption that all branches are satisfiable, the middle of the figure shows the execution tree which splits into $2^3 = 8$ leafs. The right of the figure illustrates how path merging potentially folds the execution tree together again. In this work, path merges are performed at function exit. Merges are possible because stack frame variables die at function exit. A path constraint at function exit is treated as concatenation of the function's call context and the local context. The approach misses possibilities to merge paths earlier after scope blocks inside one function. On the other hand it does not require more complex live variable analysis

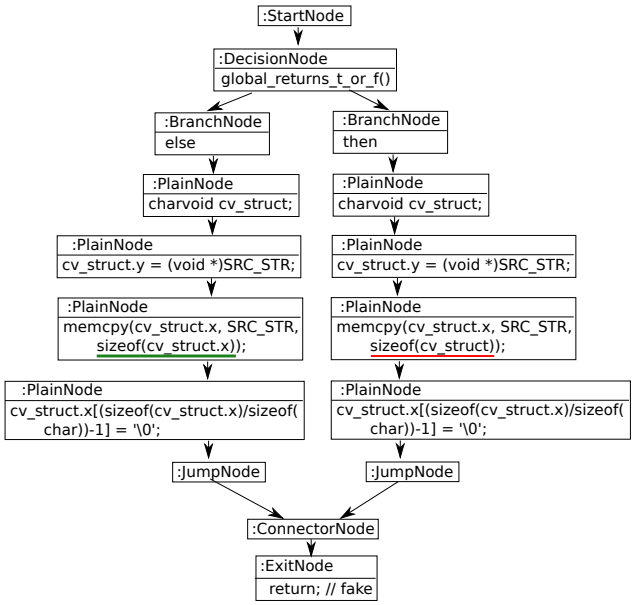


Figure 2. Control flow graph for example function from Figure 3. with buffer overflow in the `then` branch.

at intermediate points. The approach merges paths which have split inside the same function, possibly with other function calls in between. It needs to know the set of variables which have been written since the merge paths have split. This is overapproximated by the set of variables written since entering the function which is left at the program location in question. A set of potentially read variables along path extensions is not computed. From the set of variables which have been written as local context (i.e., since function entry), global variables, the return value and all variables which have been written through pointers (pointer escape, potential write to other stack frame etc.) are assumed as live. The remaining written local variables are soundly assumed as dead. The local context is then reduced by removing the dead variables. A context cache is used to lookup observed reduced local contexts from pairs of a function's exit node (in the function's control flow graph) and call context. During symbolic execution, at each exit node the context cache is queried for a merge possibility. Then, the current path is pruned (merged) if possible, otherwise the local reduced context is added as new entry to the context cache.

III. Implementation in Eclipse CDT

A. Symbolic execution with symbolic state backtracking

This subsection shortly reviews [12] which is extended by the paper at hand with path merging functionality. The backtracking symbolic execution engine [12] uses Eclipse CDT's C/C++ parser to construct abstract syntax trees (AST) from source code files. Control flow graphs (CFG) are then constructed for function definitions rooted in AST subtrees. CFG construction uses the `ControlFlowGraphBuilder` class from CDT's code analysis framework (Codan [13]). The symbolic interpretation is implemented according to the tree-based interpretation pattern from [18], the translator class extends CDT's `ASTVisitor` (visitor pattern [19]). The interpretation is symbolic, i.e., variable values are logic formulas. Satisfiability queries to the SMT solver use the SMTLIB

 ISBN: 978-1-61208-376-6

```
typedef struct _charvoid
{
    char x[16];
    void * y;
    void * z;
} charvoid;

void CWE121_memcpy_12_bad_simplified() {
  if(global_returns_t_or_f()) {
    charvoid cv_struct;
    cv_struct.y = (void *)SRC_STR;
    /* FLAW: Use the sizeof(cv_struct) which
       will overwrite the pointer y */
    memcpy(cv_struct.x, SRC_STR,
       sizeof(cv_struct));
    /* null terminate the string */
    cv_struct.x[(sizeof(cv_struct.x)/sizeof(
       char))-1] = '\0';
  }
  else {
    charvoid cv_struct;
    cv_struct.y = (void *)SRC_STR;
    /* FIX: Use sizeof(cv_struct.x) to avoid
       overwriting the pointer y */
    memcpy(cv_struct.x, SRC_STR,
       sizeof(cv_struct.x));
    /* null terminate the string */
    cv_struct.x[(sizeof(cv_struct.x)/sizeof(
       char))-1] = '\0';
  }
}
```

Figure 3. Simplified example function from [16], contains a buffer overflow in the `then` branch. Corresponding CFG in Figure 2.

sublogic of arrays, uninterpreted functions and nonlinear integer and real arithmetic (AUFNIRA). Backtracking is enabled by a class `ActionLog` which records certain semantic actions performed for CFG nodes on the current path (e.g., variable creation or hiding). If for example a function exit is backtracked, the function's stack frame with contained variables must be made visible again. Dead variables are therefore not garbage-colled, because this would impede backtracking. The engine further allows to record and visualize explored parts of a program execution tree. The engine was evaluated in [12] by measuring detection accuracy (false positives and false negatives) and run-times for the detection of buffer overflows in Juliet test programs.

B. Path merging

In this implementation, paths are merged at function exit. The method can merge paths which have split since entering the same function, with the possibility that several other functions are called between entering and leaving the function. Path merging needs knowledge about the sets of written variables since path split. The implementation uses the class `ActionLog` from [12] to derive this information. It contains all writes to variables, including writes to globals and writes through pointers (potentially to other stack frames). The action log is looked through backwards up to the current function's CFG start node, and the reduced local context is built from the variable declaration actions. The reduced local context is yielded by removing all writes to variables if the variables don't have global scope, are not written through pointers and are not the current function's return value. This approach does not necessitate a comparably more complex dead/live variable analysis. Path merge possibilities are detected using a class `ContextCache`, which is a `HashSet`. The keys are exit nodes with function call context, the values are the observed reduced local contexts. The context cache is queried at each function exit (CFG exit node). Comparing the reduced local contexts does not necessitate expensive calls to the SMT solver.

An example function is shown as listing in Figure 3. It is a simplified version of a 'bad' function from one of the buffer overflow test cases of the Juliet suite. The control flow graph of this function is shown in Figure 2. The function contains a decision node corresponding to an `if`/`else` statement, for which both branches are satisfiable. The error location is marked by red underlining in the branch on the right of Figure 2. and by a comment in the listing. For both branches, the function only writes to stack variables, and the reduced local context at function exit is the empty set. Merging the two paths at function exit which have split at the decision node is therefore clearly possible without missing any bug.

Path merging applies in the same way to branches which belong to loops, when the loop iteration number depends on program input (otherwise there would be only one satisfiable sub-path through the loop). Symbolic execution is currently applied with loop unrolling up to a maximum loop depth bound. A path through a loop can therefore split into a maximum number of paths equal to the loop unrolling bound. Branch nodes in the CFG belonging to loop statements are treated by symbolic execution just as branch nodes belonging to `if`/`else` statements. The branch nodes also have the same labels, `'then'` for the loop body and `'else'` to skip the loop. The only difference is that loops have a connector node with two incoming branches, which closes the loop before the decision node. This however has no influence on the merging of unrolled paths.

IV. Experiments

Path merging is evaluated on the same buffer overflow test programs from the Juliet suite as [12]. These programs contain buffer overflows with the `memcpy` (18 programs) and `fgets` (36 programs) standard library functions, and cover the Juliet control and data flow variants for C (e.g., multipath loops and fuction pointers). A screenshot for error reporting with the CDT GUI is shown in Figure 7. The tests are run as JUnit plug-in tests with Eclipse 4.3 on 64bit Linux kernel 3.2.0 and an i7-4770 CPU. The same bug detection accuracy with and without path merging is validated, there are no false positive or false negative bug detections on the test set.

Figures 4. and 5. illustrate the merging of paths, which corresponds to folding the execution tree. Figure 4. shows the execution tree for a `memcpy` buffer overflow with flow variant 12. This test program contains a 'good' and a 'bad' function, where both functions contain a decision node with two satisfiable branches. The bad function is given in a simplified version in Figure 3. The tree shows only decision nodes and branch nodes. Figure 5. shows the same tree when path merging is applied. Paths are merged at two points which are indicated in the tree (the two function exits), and the traversal of two subtrees is skipped.

 ISBN: 978-1-61208-376-6

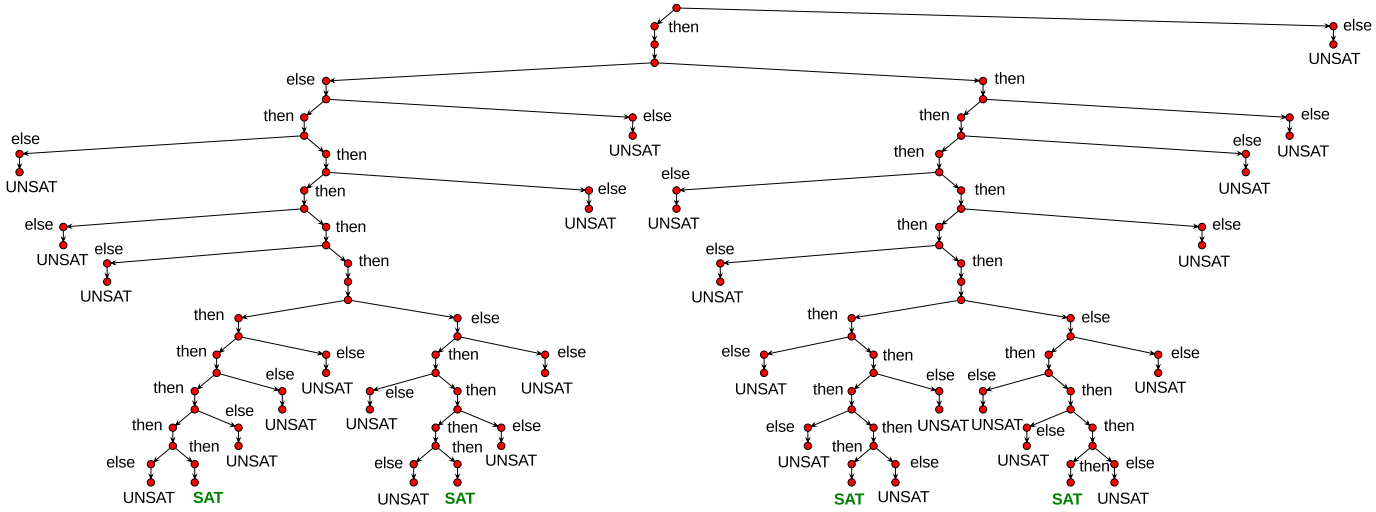


Figure 4. Execution tree for test program `CWE121_Stack_Based_Buffer_Overflow__char_type_overrun_memcpy_12` from [16], showing only decision and branch nodes.

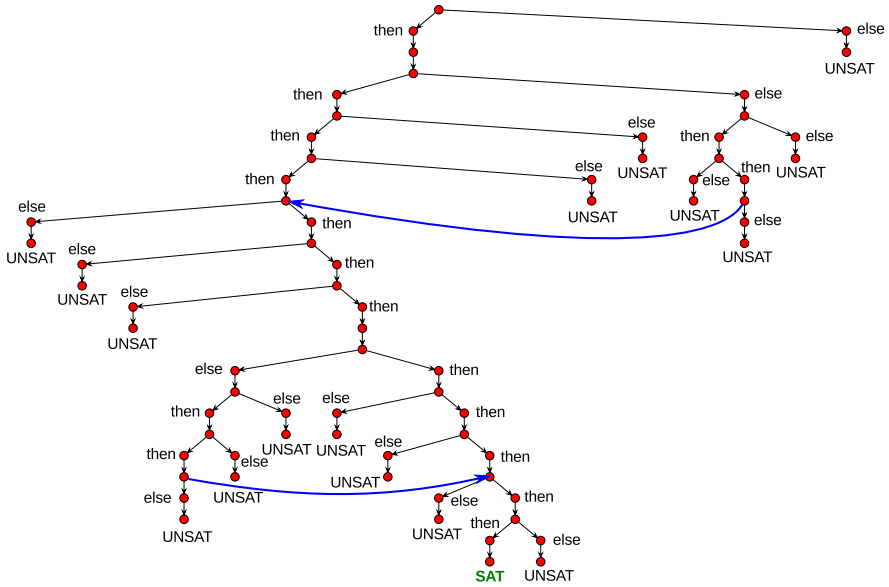


Figure 5. Effect of path merging for the test program of Figure 4. The execution tree is folded at two locations. The number of traversed satisfiable paths is reduced from four to one.

TABLE I. Analysis runtime sums for the two test sets, with and without path merging.

	CWE121_memcpy (18 test programs)	**CWE121_CWE129_fgets** (36 test programs)	**Sum** (54 test programs)
backtracking according to [12]	14,7 s	80,7 s	95,4 s
backtracking and path merging	15,3 s	34,4 s	49,7 s

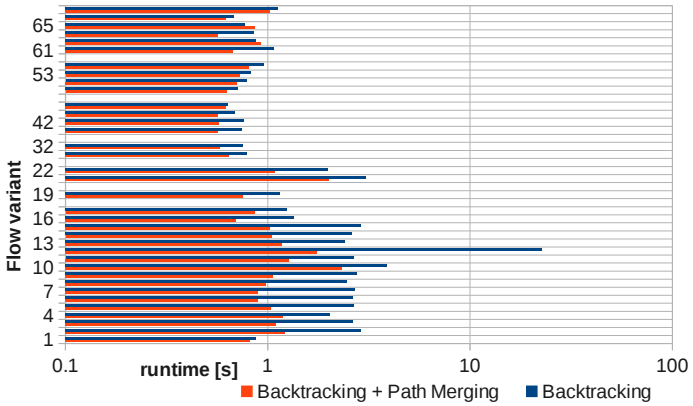


Figure 6. Analysis runtimes with and without path merging for 54 buffer overflow test programs from [16], with corresponding control/data flow variant numbers.

Figure 6. shows the analysis runtimes for the set of buffer overflows with `fgets`, for the backtracking engine and for backtracking with path merging. The figure uses a logarithmic scale and contains values for 36 flow variants. Flow variants in Juliet are not numbered consecutively, to leave room for later insertions. Since path merging folds complete subtrees of a program's execution tree, it has an exponential effect on runtimes. This is exemplified by flow variant 12. While merging paths for the `memcpy` buffer overflow with variant 12 reduces the runtime from 1.1 s to 0.8 s, the runtime for the `fgets` buffer overflow is reduced from 22.8 s (longest analysis time for any tested program) to 1.7 s. This is because the `fgets` program contains several other decision nodes with two satisfiable branches.

The sum analysis runtimes for the two test sets are given in table I. For the `memcpy` overflows path merging increases the runtime a little bit due to the overhead of computing and comparing reduced local contexts. Most of the `memcpy` programs do not contain a single decision for which both branches are satisfiable, and therefore no merge possibilities. The `fgets` test programs all contain such decisions, and the sum runtime is reduced by path merging from 80.7 s to 34.4 s. The sum runtime for the 54 programs without merging is 94 s, while path merging reduces it to 50s. The overall speedup with path merging on the test set is therefore about two, which is considerable for the tiny Juliet programs.

V. Related Work

There is a large body of work on symbolic execution available which spans over 30 years [10]. Dynamic symbolic execution for test case generation for x86 binaries is presented in [20]. To reduce complexity, only variables are modelled as symbolic which directly depend on program input, in order to find exploitable bugs. Most tools perform symbolic execution on an intermediate code representation. Apart from [6], where LLVM intermediate code is analyzed using a worklist algorithm, prominent symbolic execution engines are presented in [21] and [22]. In [21], dynamic symbolic execution of the Common Intermediate Language (MSIL/CIL) is performed for test case generation. The engine described in [22] analyzes Java bytecode. Sound path merging based on dead path differences is presented in [15], the implementation extends [6]. Merging of paths with live differences is investigated in [23]. Path disjunctions are used in the corresponding logic formulation passed to the solver. Heuristics for path merging are presented, which aim at balancing computational effort between the symbolic execution frontend and the SMT solver backend. The implementation extends [6].

VI. Conclusion

This paper described the extension of a source-level backtracking symbolic execution engine for C/C++ with path merging functionality and its implementation in Eclipse CDT. The evaluation with tiny test programs from the Juliet suite already showed a significant speedup. For larger programs path merging has an exponential effect on analysis runtimes (exponential in the number of decision nodes with more than one satisfiable branch). Future work might include extensions in different directions. One is to investigate the effect of additional merge points, for example at connector nodes after `if`/`else` and `switch` statements and loops, A memory-efficient implementation of the context cache might exploit redundant information due to shared sub-paths. The very simple live variable analysis implementation can be improved to find more merge possibilities. Inter-procedural live variable analysis could find merge possibilities, e.g., in certain flow variants with dead global variables. Another direction is the extension to support path merging in the analysis of multi-threaded code, in a straight-forward combination with [24]. A way to make the analysis scalable in order to analyze practical programs is to restrict the code coverage, for example, to branch coverage. There are less merge possibilities when coverage is restricted to fewer program paths, but path merging remains applicable without changes.

Acknowledgement

This work was funded by the German Ministry for Education and Research (BMBF) under grant 01IS13020.

References

[1] J. King, "Symbolic execution and program testing," Communications of the ACM, vol. 19, no. 7, 1976, pp. 385–394.

[2] L. deMoura and N. Bjorner, "Satisfiability modulo theories: Introduction and applications," Communications of the ACM, vol. 54, no. 9, 2011, pp. 69–77.

[3] C. Barrett, A. Stump, and C. Tinelli, "The SMT-LIB standard version 2.0," in Int. Workshop Satisfiability Modulo Theories, 2010.

[4] L. deMoura and N. Bjorner, "Z3: An efficient SMT solver," in Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2008, pp. 337–340.

[5] C. Cadar, V. Ganesh, P. Pawlowski, D. Dill, and D. Engler, "EXE: Automatically generating inputs of death," in 13th ACM Conference on Computer and Communications Security (CCS), 2006, pp. 322–335.

[6] C. Cadar, D. Dunbar, and D. Engler, "KLEE: Unassisted and automatic generation of high-coverage tests for complex systems programs," in USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2008, pp. 209–224.

[7] C. Lattner and V. Adve, "LLVM: A compilation framework for lifelong program analysis and transformation," in Int. Symp. Code Generation and Optimization (CGO), 2004, p. 75.

[8] C. Cadar et. al., "Symbolic execution for software testing in practice – preliminary assessment," in Int. Conf. Software Eng., 2011, pp. 1066–1071.

[9] C. Pasareanu and W. Visser, "A survey of new trends in symbolic execution for software testing and analysis," Int. J. Software Tools Technology Transfer, vol. 11, 2009, pp. 339–353.

 ISBN: 978-1-61208-376-6

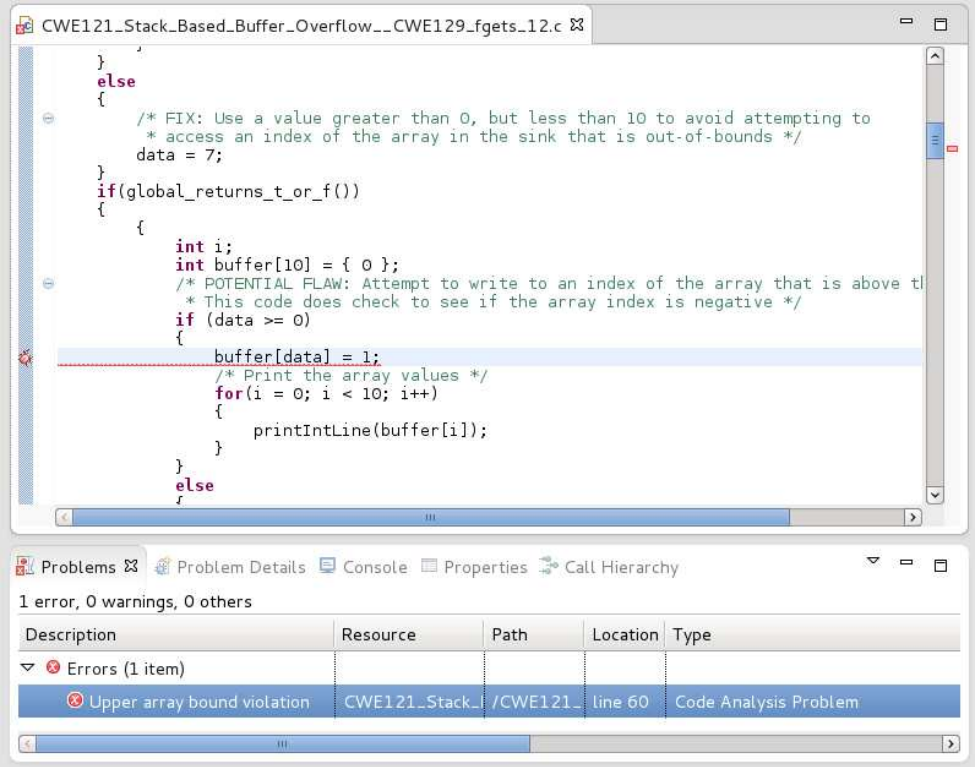


Figure 7. Error reporting in the Eclipse GUI.

[10] C. Cadar and K. Sen, “Symbolic execution for software testing: Three decades later,” Communications of the ACM, vol. 56, no. 2, 2013, pp. 82–90.

[11] T. Kremenek, “Finding software bugs with the Clang static analyzer,” LLVM Developers’ Meeting, Aug. 2008, retrieved: 09/2014. [Online]. Available: http://llvm.org/devmtg/2008-08/Kremenek_StaticAnalyzer.pdf

[12] A. Ibing, “Parallel SMT-constrained symbolic execution for Eclipse CDT/Codan,” in Int. Conf. Testing Software and Systems (ICTSS), 2013, pp. 196–206.

[13] A. Laskavaia, “Codan- C/C++ static analysis framework for CDT,” in EclipseCon, 2011, retrieved: 09/2014. [Online]. Available: http://www.eclipsecon.org/2011/sessions/index0a55.html?id=2088

[14] F. Nielson, H. Nielson, and C. Hankin, Principles of Program Analysis. Springer, 2010.

[15] P. Boonstoppel, C. Cadar, and D. Engler, “RWset: Attacking path explosion in constraint-based test generation,” in Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2008, pp. 351–366.

[16] T. Boland and P. Black, “Juliet 1.1 C/C++ and Java test suite,” IEEE Computer, vol. 45, no. 10, 2012, pp. 88–90.

[17] R. Martin, S. Barnum, and S. Christey, “Being explicit about security weaknesses,” CrossTalk The Journal of Defense Software Engineering, vol. 20, 3 2007, pp. 4–8.

[18] T. Parr, Language Implementation Patterns. Pragmatic Bookshelf, 2010.

[19] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.

[20] P. Godefroid, M. Levin, and D. Molnar, “Automated whitebox fuzz testing,” in Network and Distributed System Security Symp. (NDSS), 2008.

[21] N. Tillmann and J. Halleux, “Pex – white box test generation for .NET,” in Int. Conf. Tests and Proofs (TAP), 2008, pp. 134–153.

[22] W. Visser, C. Pasareanu, and S. Khurshid, “Test input generation with Java PathFinder,” in Int. Symp. Software Testing and Analysis (ISSTA), 2004, pp. 97–107.

[23] V. Kuznetsov, J. Kinder, S. Bucur, and G. Candea, “Efficient state merging in symbolic execution,” in Conf. Programming Language Design and Implementation (PLDI), 2012, pp. 193–204.

[24] A. Ibing, “Path-sensitive race detection with partial order reduced symbolic execution,” in Workshop on Formal Methods in the Development of Software (WS-FMDS), 2014, in press.

 ISBN: 978-1-61208-376-6

Security Extensions for Mobile Commerce Objects

Nazri Abdullah
Faculty of Computer Science and
Information Technology
Universiti Tun Hussien Onn Malaysia
Johor, Malaysia
anazri@uthm.edu.my

Ioannis Kounelis, Sead Muftic
School of Information and
Communication Technology
Royal Institute of Technology (KTH)
Stockholm, Sweden
{kounelis, sead}@kth.se

Abstract **- Electronic commerce and its variance mobile commerce have tremendously increased their popularity in the last several years. As mobile devices have become the most popular mean to access and use the Internet, mobile commerce and its security are timely and very hot topics. Yet, today there is still no consistent model of various m–commerce applications and transactions, even less clear specification of their security. In order to address and solve those issues, in this paper, we first establish the concept of mobile commerce objects, an equivalent of virtual currencies, used for m–commerce transactions. We describe functionalities and unique characteristics of these objects; we follow with security requirements, and then offer some solutions – security extensions of these objects. All solutions are treated within the complete lifecycle of creation and use of the m–commerce objects.**

Keywords - mobile commerce; m–commerce; m-objects; security; privacy

I. INTRODUCTION

As mobile commerce (m-commerce) continues to evolve, it is a matter of time that it becomes the main source of online commerce [1]. In this paper, we describe our vision of m-commerce, by differentiating the goods that can be purchased in seven categories - we call them m-commerce objects. The m-objects have different requirements and are therefore treated in a separate way from the actors involved in a mobile commerce transaction.

We first provide the results of our analysis of the current concept of m-commerce objects. However, we also take two further steps: we consider the security features and the extensions that they need and moreover, what mechanisms and technology can be used to ensure and enforce such extensions.

Our research is focused on user aspects of various m–commerce systems, ensuring that the mechanisms we introduce allow users to protect their privacy and at the same time to verify authenticity, integrity and availability of digital goods that they are purchasing.

The next section of the paper describes various examples of m-commerce objects, based on our concept of a so-called generic m–commerce object. Section 3 introduces the main actors in an m-commerce scenario. Section 4 analyses security features and requirements targeted as goals of our design and also describes methodologies and technologies that can be used for implementation of those features. Section 5 demonstrates the dynamic use of the m-objects security features. Section 6 briefly introduces one of the popular m–commerce payment systems – Bitcoin and justifies our use of some of the innovative ideas that Bitcoin has introduced. Section 7 contains relevant work and compares the results with ours, while in section 8 we discuss our findings and approach. Finally, section 9 contains conclusions and suggestions for future work

II. SPECIFICATION OF MOBILE COMMERCE OBJECTS

It is important to understand the similarities and differences between various types of m-commerce objects. The definitions given below have been also documented in our previously published research paper [2]. The criterion for some well-known transactions to be classified as m–commerce objects is whether they have direct – explicit or indirect – implicit value. An example of an m–commerce object with explicit value is a pre–paid card – its value is money paid for the card. An example of an object with implicit value may be various discounts or benefits based on different types of memberships.

In this section, we list typical m–commerce objects described in some form of an order, starting from those that do not have explicit value all the way up to those that have strictly determined value. The objects are also "sorted" in the increasing complexity of their use.

The first type of m–commerce object is promotions. They publicize a product or a service with discount, so that the offered discount represents an implicit value of this type of m–commerce object [3]. In a mobile digital environment, these objects can be managed through personalized advertisements received through the Internet or even through a personal area network. A citizen with a Bluetooth enabled phone, for example, may receive personalized promotions or discounts to his/her phone via Bluetooth when shopping in a mall. The project PROMO demonstrates how this can be achieved [4]. Promotions do not require payments by users, which means that this type of m–commerce objects can be obtained without associated financial transaction.

The next type of m–commerce object is a mobile coupon. Those are text or picture vouchers solicited or purchased and delivered to a consumer's mobile phone. This object can be stored and exchanged for a financial discount, when purchasing a product or service [5]. The most important

 ISBN: 978-1-61208-376-6

difference between promotions and coupons is that coupons have a value (expressed either as discount or monetary value), while promotions are mostly used for advertising of discounts.

The third type of m–commerce objects that we consider is a standard voucher used mainly today in paper form. It is a small printed piece of paper that represents the right to claim goods or services [6]. In the case of an m-voucher, there is no printed copy, but a digital equivalent with the unique identifier, such as a barcode or a Quick Respone (QR) code, stored locally on the phone or remotely at the m–commerce server. One example of such vouchers are coupons distributed by Groupon [7] or some other similar companies. In order to acquire a voucher, a payment transaction is usually involved. The difference between a voucher and a coupon is that the voucher is a complete representation of a product or a service, while the coupon is an offer/discount for the product or service. In other words, having a voucher means that the specific product has already been bought in advance while with a coupon consumers may claim it at alternative places or not at all, if the coupon was not purchased.

Another type m–commerce object is a gift card. In real life it is usually a tangible device (plastic card), embedded or encoded in a plastic, electronic or other form with a value based on a payment, which promises to provide to the bearer merchandise of value equal to the remaining balance of the card [8]. In a digital environment, a gift card can be seen as an equivalent to a very specific and limited pre–paid amount in an e-wallet, which can be used only with the specific merchant, in a particular shop or for a particular series of products. The difference with the voucher is that a gift card can be used as many times as possible, as long as there is credit left in the card. The voucher however is usually limited to one or to a predefined number of claims.

Mobile ticketing is an electronic realization with the help of a mobile device of the proof of access/usage of rights to a particular service [9]. There are many forms and ways to purchase a mobile ticket. Usually, a Short Message Service (SMS) message is the outcome of the purchase (the receipt).

A pre-paid card has many similarities with the gift card. It is a value stored in an e-wallet or in some account that can be loaded with money in order to be used mostly for micropayments [10]. The main difference with a gift card is that a pre-paid card is intended to be used by the owner and not to be gifted to another party and is usually not limited to specific merchants. More importantly, a pre-paid card can be recharged when the pre–paid amount is exhausted. By their purpose and type of transactions supported, the very popular pre–paid airtime may also be considered as one type of pre–paid card. In case of airtime, such m–commerce object is usually called telecom account.

Our final example and type of m–commerce object is a bonus card (also called loyalty card). This type of object usually refers to accumulation of points that a user gains from various purchases [11]. They are usually represented as supermarket cards, airline bonus cards, membership cards, etc., issued by merchants/businesses that give points to the customers depending on the value of goods or services that they previously purchased. Their owners can later use these points, in exchange for products or services. Such cards are usually free to acquire, but are bound to a user (or to a small closed and related group of users, such as members of a family).

III. THE CONCEPT OF M-COMMERCE TRANSACTIONS

In this section, we introduce the main actors and define their roles in a typical m–commerce transaction together with the terms used and their interpretation. The purpose for the reader is to better understand the text in the remaining sections of the paper.

There are four actors in an m–commerce transaction:

1) *Merchant:* This is a business entity that offers some services or products for purchase. Merchants define availability, price, and all specific attributes of the m-commerce objects they issue and accept.
2) *Customer/User/Client:* The customer is the entity that obtains or purchases an m-commerce object in order to later redeem it.
3) *Redemption Point/Redeemer:* The place where m-commerce objects can be redeemed. In some cases this entity can be the same as the entity that issued the object, but most likely they will differ. For example, when buying a ticket for a concert, the merchant is the company selling tickets, while the redemption point is the venue where the concert takes place.
4) *m–Commerce Services Provider:* This is a trusted–third party in our system. It is the central entity that all other actors communicate with in order to handle their requests. Depending on the actor, different roles and services may be offered by the services provider. Merchants use the provider to make available their m-commerce objects, customers use it to acquire such objects and later use them, and redemption points use it for verification of validity of m-commerce objects in the redemption phase.

IV. SECURITY FEATURES AND ATTRIBUTES OF M-COMMERCE OBJECTS

Each m-commerce object has a number of attributes that define it, both in terms of security and usability. Such attributes are required by both participating parties, object's issuers (merchants, m–commerce providers) and also by users, as their enforcement is an advantage for all parties.

 ISBN: 978-1-61208-376-6

A. Authenticity of m–Commerce Objects

This security property refers to the capability of the recipient to verify the originality of the m-commerce object, which includes verification of the identity of its issuer as well as correct and original contents of an objet. Verification can be performed by both the customer and the redeemer.

The customer should perform this check in the process of acquisition of an m-commerce object, i.e., before paying for it. This should be done in a timely manner, without interfering with the customer's purchasing experience in any way. In the best case, it should be an automated procedure, embedded in the acquisition phase and fully transparent to the user. The user should only be informed of the outcome of the procedure before giving the consent to proceed with the payment.

The redeemer should also perform verification of the object's authenticity before redeeming the m-commerce object. Such action should be performed with the assistance of the m–commerce Provider. This control will protect the redeemer against fraudulent attempts to acquire fake m-commerce objects.

Authenticity of m–commerce objects can be supported by the issuer (merchant or m–commerce provider) by digitally signing the object. Then the client will be able to verify the signature, as the certificates of either the provider or the merchant will be known to him/her.

B. Security of m–Commerce Objects

When referring to the security of an m-commerce object, we are actually referring to two different aspects: the *integrity* and the *confidentiality* of its content. These two issues together can be also interpreted as the user's privacy.

1) Integrity: Integrity refers to protection of the m-commerce object's values, against illegal intentional or accidental modifications, after its creation. This security feature is actually equivalent to the authenticity, described in the previous section. Therefore, all the mechanisms described above are also applied when referring to the integrity.

2) Confidentiality of Content/Privacy for the User:

Confidentiality of the content refers to the user's privacy when proving that he/she is the owner of an m-commerce object. This property is not applicable to all m-commerce objects, but rather depends on the type and also sensitive nature of the object.

Namely, the user should be able to prove that he/she is the owner of an object without revealing any information of what he/she has purchased with that object. The content of the object should be encrypted by the user upon purchase and will only be decrypted when redeemed. In the intermediate states, a header/part of the m-commerce object, indicating the owner, will be unencrypted, but signed by the issuer. If the m–commerce provider is involved, it is already in possession of user's identifying information and therefore there is no need to exchange any extra data with every purchase. The user should be able to define sensitivity level of the content in accordance to his/her preferences and then the system will enforce those preferences during the acquisition phase.

The security mechanisms for confidentiality of m–commerce objects are standard symmetric key crypto algorithms. What makes this feature very complicated to design and implement is use of partial values of some objects. For instance, gift cards or pre–paid cards may be partially redeemed. In such situations, encrypted objects must be decrypted, partially claimed, and then the new contents must be encrypted again.

C. Duplication

This is the property of m–commerce objects that specifies whether an object can be duplicated, i.e., whether a valid and legitimate copy of an m-commerce object can be created by its owner. Obviously, if objects have explicit value, this possibility should be prevented. In some virtual currency systems this feature is called prevention of "double spending".

In order to guarantee non–duplication, if required, a signature created over a random, unique, non–replicated value is needed. Therefore, the issuer will have to create a new value and sign a counter, possibly along with a timestamp, which when duplicated will not be possible to be changed, since in that case the signature will not be valid.

This security property is useful when an instance of an m-commerce object must be unique. For example, a voucher for a specific service or a ticket for a concert are examples of non–duplicated objects. On the other hand, if an m-commerce object is a free of charge promotion, it is actually in the merchant's interest to have the object duplicated and distributed as widely as possible, as this will give it more visibility.

The unique value or counter must be specified by the issuer in the process of creation of the object. In cases where users create copies of the object, the redemption of the second instance of the object will not be accepted as the unique value of the counter will be checked by the redemption point. This verification is very tricky in open, distributed environments and the Bitcoin concept has successfully addressed and eliminated this problem. This is one more reason why we have adopted its concept and some specific solutions for security of our m–commerce objects.

There is of course a risk that an m-commerce object is illegally duplicated after its first redemption and the illegal copy is distributed to some other entity. Then, when the legitimate owner tries to redeem the original object, he/she will be denied redemption. This problem may be eliminated by having the owner to sign the object as well. Therefore, if there is an attempt to redeem another copy of the same object, the owner will be consulted for approval as well. In addition, if the same person is trying to cheat the system, the unique identifier of the object will be sufficient to prohibit such action.

D. Transferability

This feature represents the property of an object to legitimately change ownership of an m-commerce object.

If objects are transferable, this action must be performed with the assistance of the m–commerce provider, since it is the actor responsible of signing the object and assigning its

 ISBN: 978-1-61208-376-6

ownership. Moreover, even if a transfer is initiated or performed by one person to another person, the two entities will protect their privacy between them, as they will not have to exchange any details apart from their system identifiers (usually randomly assigned identifiers (IDs)).

The provider will receive a «transfer request» command from the current owner along with the ID of the recipient and then, if and only if the new owner meets all security requirements associated with the specific object, for instance age limit, the transfer will be performed. The owner of the object will be changed and the object will be re–signed by the m–commerce provider.

A drawback of this approach is the necessity to have provider's server connectivity at the time of the exchange. If the server is not accessible at the time of the transaction, the request may be temporarily saved on the current owner's station and when the connection is established, the request will be forwarded to the server and the transfer of the object will be performed.

Finally, in this stage of our research, the option to have a fee charged for this exchange is not considered. All transfers are free of charge. The payment in order to acquire the m-commerce object from the provider has already taken place from the first owner.

E. Monetary Value

Monetary value is the attribute representing the financial value of the m-commerce object, i.e., if it can be "exchanged" for something that has a cost.

This property does not provide any extra feature or option, as all the previous ones, but rather is a key factor affecting which of the previous mechanisms must be enforced to the specific m-commerce object itself. It can be better viewed as a property rather than an extra attribute.

If an m-commerce object has a monetary value, then, it is both in the merchant's as well as in the consumer's interest to have the object secured in all the above mentioned ways. As a conclusion, authenticity, non–duplication, integrity, and confidentiality for each object are needed.

The contents of each object are cryptographically encapsulated by the m–commerce provider and therefore the m–commerce objects cannot be tampered with. It is up to the object's owner to disclosure such information to any third parties or to reveal it only when absolutely needed (during the redemption process).

F. Purchased

M-commerce objects have this property if money is needed in order to acquire the object.

This property is strongly linked to the monetary value property. What applies there is also applicable to this property as well. The main difference whether an object has been purchased or not indicates solely the way in which the owner has acquired such object. Monetary value indicates also the actual value of the object.

G. Multiple/Partial Use

This property indicates whether an m-commerce object can be used more than once, i.e., if its total value may be partially redeemed. If yes, then such functionality can be enforced in two different ways:

1) There is a predefined number of uses that is decreased after each use (or increased when the user buys some more quantities of the object). An example may be tickets for public transportation.
2) It can be an amount (in Euros for example) that is decreased (or increased if the user charges the object). This is mostly valid for a gift card.

In both cases, the m–commerce provider must be involved in order to approve/confirm the remaining number of uses or the amount/value of the object. The new value of the object, after its adjustment, is signed and therefore can be verified by the provider at any time.

H. Tracking

This is the ability of the system to track past transactions and determine the current status of the object, i.e., the ability to track its full life cycle.

The attributes of the m–commerce objects that may be interesting when tracking are the date of creation, previous uses in terms of volume and content, and information about all previous owners. All these aspects depend on the type of the specific m-commerce object and the specific values of its attributes.

Tracking an object's history may be performed by the user without the need to engage the m–commerce provider in that process. For example, all previous uses can be recorded in the header of the object and in that way they may be retrieved in a read-only mode. They are always signed by the m–commerce provider. As such, it is generally recommended to reveal the values of all non–sensitive attributes (in terms of user privacy) in a read-only mode, so users can retrieve them at any given time and without requirement to be on-line, connected to Internet, in that process.

V. DYNAMIC USE OF SECURITY FEATURES

The security features described above are based on the basic set of security services: confidentiality, integrity and authenticity and may be applied during any phase of an m-commerce object's lifecycle. However, what makes these features different from the classical application of security services in some other network application is that they are applied in a very dynamic way.

The reason for the dynamic applicability is the complex reuse of the majority of the m-commerce objects. For example, a voucher that has a specific number of admissions to a service needs to have this number updated accordingly after each use. This implies that various features established in the initial phase when creating an object, are re-applied after every use of the object. Therefore, in case of m–commerce objects, special security protocols are needed, supporting repetitive application of security services. These protocols, therefore, effectively ensure that in all phases of its lifetime, each m-commerce object meets all the security requirements according to their special needs and properties.

 ISBN: 978-1-61208-376-6

As explained earlier, these needs and requirements are determined by the attributes of the specific m-commerce object, each depending on their contents and the nature of use. A full list of significant attributes for our m–commerce objects is given in Table 1.

TABLE I. M–COMMERCE OBJECTS AND THEIR ATTRIBUTES

	Voucher	Gift Card	Ticket	Promotions	Coupon	Bonus Card	Prepaid Card
Multiple Use	Maybe	Maybe	Maybe	Yes	Yes	Yes	Yes
Purchased	Yes	Yes	Yes	No	No	No	Yes
Monetary Value	Yes	Yes	Yes	No	Yes	Yes	Yes
Transferability	Yes	Yes	Maybe	Yes	Yes	No	Maybe
Duplication	No	No	No	Yes	No	No	No
Security	Yes	Yes	Yes	No	Maybe	Yes	Yes
Authentication	Yes	Yes	Yes	No	Maybe	Yes	Yes

The enforcement of security services that support those requirements takes place at both, the client side and the server side. Clients control object's authenticity and conformity to a predefined set of standard attribute values. The server creates these attributes and cryptographically encapsulates them, thus binding security credentials to the values of the m–commerce attributes. When a value of some attribute of an m-commerce object needs to be changed, the client sends a corresponding request to the server which performs the same procedure all over again, updating the values of m-commerce objects' attributes. By "client" in this case we do not necessarily mean the end–user but also any other entity in the m–commerce transactions chain, such as a merchant or a retailer. For a more comprehensive description of the actors and their interactions, the reader should refer to [2].

A. *Comparison with a traditional secure-by-design system.*

To illustrate how the dynamic nature of security services is different from some other traditional network applications, in this section, we compare our previous work of a secure e-mail system based on Secure/Multipurpose Internet Mail Extensions (S/MIME) and security proxies [12] and the dynamic use of security services described above.

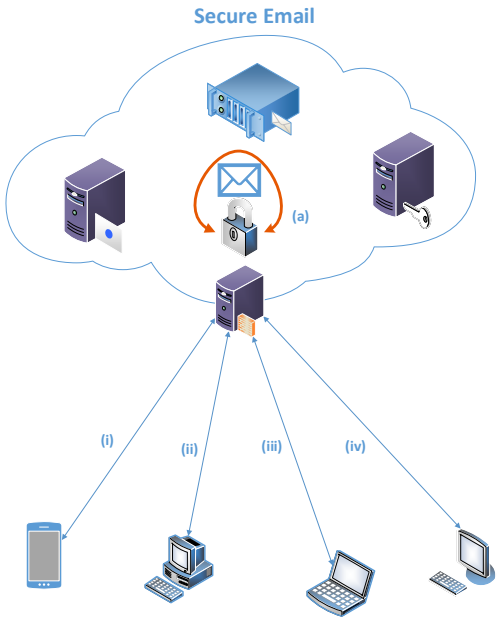


Figure 1. Secure Email Use. The security functionalities stay within the Secure Email proxy.

In the secure e-mail system, the use of security is straightforward. When a security method is applied, for example encryption or signing of an e-mail letter, a security action takes place at the e-mail client or at the security proxy and it is directly applied to the complete and final form of the specific e-mail letter. Then, in order to read such letter, i.e., decrypt or verify it, the e-mail client of the proxy server is again used. In the intermediate states of the protected e-mail letter, a third party cannot manipulate the message. For the client, the security is completely transparent; he/she only sees the "clear" output regardless of the way he/she accesses the proxy server. When accessing the secure e-mail from any end device (see Figure 1, i-iv), the result for the client is the same as for all the security functionalities that are performed internally.

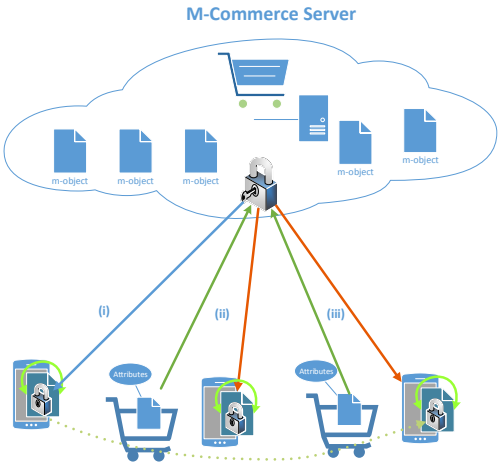


Figure 2. The dynamic reuse of security attributes for an m-commerce object.

In the m-commerce case, security services and mechanisms are re-applied after every use of the m-commerce object. Although there is a similarity in the two cases, in the sense that the server is the one that takes care of the enforcement of the security, the use for the client is different. When the client is using the m-object, the values of the m-object change. This directly implies that the signature and the authenticity of the m-object is not the same anymore and as a result the security attributes need to be readjusted to the new data. The server is the one that takes the responsibility of fulfilling this task and then resends the newly adjusted m-object to the client. The client, however, in this case has the ability to verify and recognize the security enhancements. This can be seen in Figure 2; as the m-commerce object is used in the real world (i-iii), its values change. These changes are taken into account each time at the server and as a result all the security attributes are re-applied. The attributes can be read not only by the server but also from the client side.

VI. BITCOIN SYSTEM

Bitcoin [13] is a virtual currency that has become very popular in the last few years. It uses a peer-to-peer network to authorize and verify transactions and has no central authority like all other traditional payment systems. One of its main advantages is that the transactions are anonymous (actually pseudonymous) and third parties are not involved when performing payments or transfer of money, even for verification of participants.

Although the details of the protocol are not in the scope of this paper, we briefly review some of the innovative features that Bitcoin has introduced in the payment environment and we also indicate how these features can be applied and improved for security of our own system, that is for security of other virtual currencies.

The most interesting feature of the Bitcoin system is the concept and use of the blockchain. Transactions are grouped in specific blocks of data and these blocks are linked in the chain, called blockchain. Therefore, the blockchain contains and reveals the history of all transactions that have taken place in the Bitcoin system, since its creation. In order for a new transaction to be considered valid and accepted in the system, it must be included in the blockchain and then, applying mathematically and computationally complex procedures, be verified. Moreover, all accounts (Bitcoin addresses) in the Bitcoin system are publicly available, which means that anyone can check the balance of each account and how it has accumulated its current balance.

As Bitcoin addresses are long random strings of characters without any meaning and interpretation, there is no direct link between the owner of an address and the address itself. Nonetheless, it is still feasible for someone to try to find information that may be leaked about identities and the addresses that belong to them. This feature is also very useful in order to protect one's privacy. We would like to extend this feature by offering both anonymous transactions and by providing the possibility to have verified and authorized transactions for authorities and users who need verification of authenticity and reliability of selected transactions.

Our intention is to create a side-chain to Bitcoin, starting with one of the m-commerce objects. Side-chain is a separate blockchain, which is backed by Bitcoins, in the same way that currencies are backed by gold [14]. Doing so, we will be able to take advantage of the above-mentioned Bitcoin characteristics, while in the meantime manipulate the side-chain according to the m-object's needs.

VII. RELATED WORK

The concept of m-commerce is not new to the research community. From the early years of mobile device adoption, both with the use of the first mobile smart phones or with the use of Personal Digital Assistants (PDA), the importance and potential growth of m-commerce was foreseen and a number of research solutions with a focus on security were proposed.

Nambiar et al [15] performed an analysis on payment transactions security in mobile commerce. As their research is 10 years old, technologies such as Wireless Application Protocol (WAP) and Java Micro Edition (J2ME) are not considered relative for modern development. Nonetheless, we consider the use of the SIM Application Toolkit still relevant, although still not used by major vendors, as demonstrated by our previous work [16]. Hughes [17] provides a comparison between Business-to-Business (B2B) and Business-to-Consumer (B2C), pointing out which Public Key Infrastructure (PKI) components are not necessary for a B2C marketplace. Lam et al [18] propose a lightweight security for mobile commerce transactions. Their proposal is based on public key cryptography and is end-to-end, thus avoiding any intermediate insecure actors. Chang et al [19] have proposed a secure e-coupon system for mobile users. The requirements Chang proposes are similar to ours with the difference that we extend them by including duplication, monetary value, multiple use and tracing.

We consider the above research results valuable input for our further research. However, it has to be pointed out, that as the works are relatively old, most of the restrictions mentioned are not applicable any more. For example, the computational power of the mobile devices, the wireless connectivity, the ease of use of modern smart phones and the powerful in terms of capabilities mobile operating systems, make it possible to overcome many of the restrictions that were mentioned a few years ago.

The most significant difference with our solution is that we propose a system that differentiates approach and security mechanisms depending on the nature of the m-object. The approach is not universal and applied blindly to all objects. That is the reason why we have distinguished and created the different m-object categories.

VIII. DISCUSSION

In this work, we have presented and described our notion of mobile commerce objects, their use and special characteristics. We believe that the differentiation that we propose between these digital representations of goods is a useful distinction that could be a key enabler for future mobile commerce systems.

 ISBN: 978-1-61208-376-6

The most significant challenge we had to face was to clearly distinguish between the proposed categories of m-commerce objects. In fact, by searching the literature, the notions and terms used are some times mixed or may have a double meaning. For example, the difference between promotions and coupons is very delegate and may create confusion.

When dealing with a client-to-server connection, even more when the client is a mobile device, it is reasonable to face well-known vulnerabilities, specific to such environment. For example, threats like eavesdropping, spoofing, Denial of Service (DOS), data manipulation have to be dealt with when deploying such a system. We consider the description and further analysis of such threats not in the scope of this paper; we take however into account the results from [20] and [21] in order to deal with them in our future work.

Moreover, in order to avoid some of the human related vulnerabilities, e.g. having a mobile device stolen, we use secure storage of the m-objects on the user's device, as described in [16] and [22]. In such case, the m-object cannot be retrieved even in the case where the legitimate owner loses his/her device.

Finally, with the use of a mobile device as the main enabler of m-objects, it is evident that connectivity issues may appear. However, with the wide pervasiveness of wireless technologies, both mobile communications and Wi-Fi connections, we consider connectivity and speed connection to be less of a problem and not to influence the client experience.

Our goal with this article is to provide a reference for future use of m-commerce objects but to also propose what security and privacy characteristics are needed for them. With our distinction, we have made it easier to implement security enhancements for the m-objects as we provide guidelines on which requirements are needed. We also point out how this approach differs from a classical security solution from both the server and the client side. Our intention is to use the current paper as a reference for our further developments as described in the section below.

IX. CONCLUSIONS AND FUTURE WORKS

In this paper, we have described our concept of m-commerce objects and analyzed security mechanisms that are required in order to ensure protection and consistency of their attributes. We have also emphasized security services that ensure the integrity and authenticity of m-commerce objects. Those services are provided to all actors in the system, each having a different motivation and reason for ensuring the correctness of the objects and transactions. Moreover, we ensure customers' privacy by concealing sensitive information from intermediate parties. Finally, we refer to the Bitcoin system as a basis of the new paradigm for use of virtual currencies.

For future work, we plan to use some of the innovative mechanisms that Bitcoin has introduced for our design and implementation of the complex security system for the protection of virtual currencies. Anonymity and traceability of accounts and transactions are desired features in our design and implementation. However, they will be combined with the corresponding security enhancements that will allow legal entities to intervene in case of illegal transactions and activities.

REFERENCES

[1] B. Siwicki, "E-commerce and m-commerce: The next five years," internetretailer.com, 28-Apr-2014. [Online]. Available: http://www.internetretailer.com/commentary/2014/04/28/e-commerce-and-m-commerce-next-five-years. [Retrieved: Oct-2014].

[2] I. Kounelis, G. Baldini, S. Muftic, and J. Loschner, "An Architecture for Secure m-Commerce Applications," in 2013 19th International Conference on Control Systems and Computer Science (CSCS), 2013, pp. 519–525.

[3] WordReference, "Promotion." [Online]. Available: www.wordreference.com/definition/promotion. [Retrieved: Jan-2012].

[4] PROMO, "Proximity Marketing Solution." [Online]. Available: http://isin.dti.supsi.ch/NetLab/index.php/promo. [Retrieved: Jan-2012].

[5] Mobile Marketing Association, "Introduction to Mobile Coupons," MMA, 2007.

[6] K. Fujimura and D. Eastlake, "RFC 3506 - Requirements and Design for Voucher Trading System (VTS)," 2003.

[7] "Groupon." [Online]. Available: http://www.groupon.com/. [Retrieved: Feb-2013].

[8] Kansas Statutes Annotated, "Unfair Trade And Consumer Protection: Consumer Protection," 2006.

[9] G. Me, "Security overview for m-payed virtual ticketing," in Personal, Indoor and Mobile Radio Communications, 2003, pp. 844–848.

[10] US General Services Administration, "Pre-paid Card," SmartPay. [Online]. Available: https://smartpay.gsa.gov/about-gsa-smartpay/glossary#p. [Retrieved: Feb-2012].

[11] Electronic Merchant Systems, "Loyalty Card." [Online]. Available: http://www.elect-mer.com/glossary-l.html. [Retrieved: Feb-2013].

[12] I. Kounelis, S. Muftic, and J. Loeschner, "Secure and Privacy-Enhanced E-Mail System Based on the Concept of Proxies," presented at the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics - MIPRO, 2014, pp 1405-1410.

[13] "Bitcoin - Open source P2P money." [Online]. Available: https://bitcoin.org/en/. [Retrieved: May-2014].

[14] Z. Muadh, "Introduction To Sidechains and Blockchain 2.0," Deep Dot Web. [Online]. Available: http://www.deepdotweb.com/2014/06/26/sidechains-blockchain-2-0/. [Retrieved: Oct-2014].

[15] S. Nambiar, C.-T. Lu, and L. R. Liang, "Analysis of payment transaction security in mobile commerce," in Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004, 2004, pp. 475–480.

[16] I. Kounelis, H. Zhao, and S. Muftic, "Secure Middleware for Mobile Phones and UICC Applications," in Mobile Wireless Middleware, Operating Systems, and Applications, vol. 93, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 143–152.

[17] J. Hughes, "Enabling E-Commerce Through PKI," Netw. Secur., vol. 2000, no. 3, Mar. 2000, pp. 14–16.

[18] K.-Y. Lam, S.-L. Chung, M. Gu, and J.-G. Sun, "Lightweight security for mobile commerce transactions," Comput. Commun., vol. 26, no. 18, Dec. 2003, pp. 2052–2060.

[19] C. C. Chang, C. C. Wu, and I. C. Lin, "A Secure E-coupon System for Mobile Users," Jan. 2006.

[20] D. Geneiatakis, I. Kounelis, J. Loeschner, I. N. Fovino, and P. Stirparo, "Security and Privacy in Mobile Cloud Under a Citizen's Perspective," in Cyber Security and Privacy, M. Felici, Ed. Springer Berlin Heidelberg, 2013, pp. 16–27.

[21] I. Kounelis, J. Loschner, D. Shaw, and S. Scheer, "Security of service requests for cloud based m-commerce," in 2012 Proceedings of the 35th International Convention MIPRO, 2012, pp. 1479 –1483.

 ISBN: 978-1-61208-376-6

[22] F. Zhang, I. Kounelis, and S. Muftic, “Generic, Secure and Modular (GSM) Methodology for Design and Implementation of Secure Mobile Applications,” presented at the SECURWARE 2012 , The Sixth International Conference on Emerging Security Information, Systems and Technologies, 2012, pp. 1–6.

 ISBN: 978-1-61208-376-6

Attack Surface Reduction for Web Services based on Authorization Patterns

Roland Steinegger, Johannes Schäfer, Max Vogler, and Sebastian Abeck
Research Group Cooperation & Management (C&M)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
{ abeck, steinegger }@kit.edu, { johannes.schaefer, max.vogler }@student.kit.edu

***Abstract*—During the design of a security architecture for a web application, the usage of security patterns can assist with fulfilling quality attributes, such as increasing reusability or safety. The attack surface is a common indicator for the safety of a web application, thus, reducing it is a problem during design. Today's methods for attack surface reduction are not connected to security patterns and have an unknown impact on quality attributes, e.g., come with an undesirable trade-off in functionality. This paper introduces a systematic and deterministic method to reduce the attack surface of web services by deriving service interface methods from authorization patterns. We applied the method to the Participation Service that is part of the KIT Smart Campus system. The resulting RESTful web services of the application are presented and validated.**

Keywords-security pattern, attack surface, authorization, web service, rest

I. INTRODUCTION

Every web application has assets needing protection from threats, e.g., web services. Thus, securing web applications is a major issue. Security must be considered during the whole software development life cycle to build secure software [1]. In such a security-based software development life cycle, security patterns are used during the design phase to solve common security problems and build a security architecture [2].

Security patterns in the security architecture can have an impact on non-security quality attributes of the whole software system, such as loose coupling or discoverability [2]. When using security patterns, it is helpful to know this influence on the quality of the application [3]. Additional, security should be applied as early as possible to increase overall security [3]. Developers are generally not security experts and a systematical approach can help them reaching quality requirements [4]. Regarding a concrete quality attribute, the attack surface, several metrics have been introduced to measure the attack surface of whole software systems [5], object oriented designs [3][6] and web applications [7].

In addition to metrics, there are methods to reduce the attack surface, e.g., by using the Top 10 most critical applications security flaws of the Open Web Application Security Project (OWASP) [8], by removing or disabling less important or unnecessary functionality [9][10] or by reducing the permissions of the application [11]. These methods do not offer the possibility to systematically reduce the attack surface and they do not describe their influence on other quality attributes. Additionally, there is no connection to security patterns that are commonly used in a security-based development process.

Thus, we propose a method based on security patterns for authorization to reduce the attack surface of web services. The method has direct impact on the service interface. It mainly focuses on web services having a manageable amount of authorization rules that do not change periodically. It reduces the attack surface, by reducing the privileges for methods on the interface to the minimum needed, according to authorization. Furthermore, the client can choose under which privilege a service interface method should be called. Both increase the security by following the principle of least privilege and secure interaction design [12]. Our approach additionally leads to service interfaces, which are compliant with the Representational State Transfer (REST) paradigm [13].

The method is applied on the Participation Service of the KIT Smart Campus system. The service uses an Attribute-Based Access Control (ABAC) for authorization due to complex security requirements. The resulting web services of the Participation Service are introduced. The web services are analyzed using the attack surface metric of [7].

The article is structured as follows: Firstly, the needed background and related work are introduced in Section II. The approach is presented in Section III for two commonly used authorization patterns. The next Section IV shows the evaluation of the approach by applying it on the Participation Service. After the evaluation Section V discusses limitations of the approach. The paper gives conclusions and an outlook on future work in the last Section VI.

II. BACKGROUND AND RELATED WORK

In this section, the needed background for our approach is presented. This includes the software system used for evaluating the approach, the Participation Service, security patterns used for our approach, and related work on the attack surface, as well as on REST and its constraints.

A. Participation Service of the KIT Smart Campus

The KIT Smart Campus (KIT-SC) system is a web application developed at the Karlsruhe Institute of Technology (KIT). A detailed description of the KIT-SC and its features is given in [14]. The KIT-SC pursues the goal to support students and employees at learning, teaching and other activities related to the KIT campus.

The Participation Service represents a part of the KIT-SC. It provides a forum with voting and discussion features.

 ISBN: 978-1-61208-376-6

Following the principles of systemic consensus, this enables groups of users to make decisions on campus-related issues by using the modern, responsive web application.

B. Security patterns for authorization

With our approach, service interfaces are derived from authorization patterns. The steps are shown for two common security patterns: Role-Based Access Control (RBAC) and ABAC.

RBAC takes advantage of the fact, that organizations are often structured in roles, e.g., students, employees and administration [2]. These roles have certain rights and duties. The rights of these roles can be used to model the access rights in the system. Thus, subjects get all rights through their roles. In this way, the process of assigning access rights is simplified by the usage of global roles instead of individual rights [2].

The structure of RBAC shows Figure 1. Subjects have certain roles and these roles are directly connected with resources. The concrete right is associated to the connection between role and resource. As soon as roles are not applicable or a more flexible access control is required, RBAC has strong limitations [15].

ABAC is a more flexible approach because of the usage of attributes as information source for access control [15]. In addition to static roles, which can still be realized with ABAC, access control can be defined for dynamic attribute combinations of subjects, resources and environments [15]. This structure shows Figure 2. Subjects are directly connected to resources. The right is associated to this connection and uses the attributes.

Yuan et al.'s formal definition [15] is: S, R and E are subjects, resources and environments with pre-defined attribute sets SAn, RAn and EAn. A policy rule that decides on whether a subject s can access a resource r in an environment e is a Boolean function of s, r and e's attributes:

canAccess(s, r, e) ← f(ATTR(s), ATTR(r), ATTR(e))

where ATTR() is a function that assigns every currently valid attribute to a subject, resource or environment.

Authorization using ABAC is, thus, more fine-grained than RBAC. But as negative aspect, it is more complex to implement.

C. Attack Surface

With our approach, we connect security patterns with software product quality according to ISO/IEC 25010 [16]. These are on the one hand the quality attribute attack surface and on the other hand quality attributes connected to the REST paradigm. In this section, we introduce the attack surface.

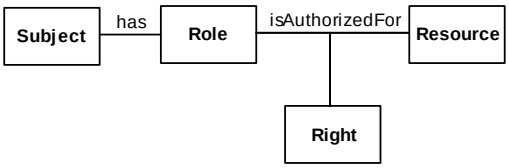


Figure 2. Scheme of the Roled-Based Access Control based on [17]

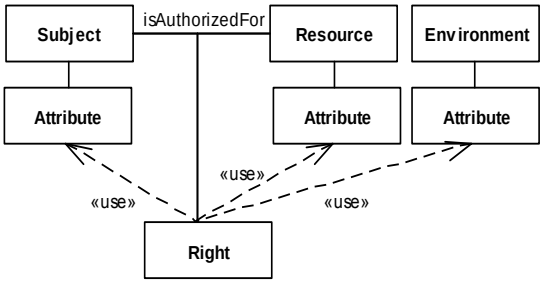


Figure 1. Scheme of the Attribute-Based Access Control based on [17]

Developers wish to anticipate the vulnerability of their software system prior to deployment. The popular concept of loose coupling and the distribution of systems or web applications lead to an increasing number of interfaces [18]. These are natural security boundaries that augment the attack surface, an indicator for measuring a system's vulnerability towards external attacks [7][9].

The attack surface does not give information on code quality or high-value architectural design. And neither does a large attack surface imply that a system has much vulnerability, nor does a small attack surface mean little vulnerability. But a large attack surface indicates that an attacker presumably needs less effort for exploiting vulnerabilities [5]. The reduction of the attack surface, therefore, reduces the overall security risk – a product of the probability, the consequences of occurrence of a hazardous event and the asset value: Risk = Threat × Vulnerability × Asset Value [19]. Think of two web applications with similar functionality and value – the one with a higher attack surface is more likely to be chosen to attack amongst these opportunities.

We use the attack surface metric for web applications [7] to evaluate our approach. The metric is based on parameters grouped into parameter families. These parameter families are *Degree of Distribution*, *Dynamic Creation*, *Security Features*, *Input Vectors*, *Active Content*, *Cookies* and *Access Control*. Parameters are, e.g., *Role* and *Privileges* for the parameter family *Access Control*. For each of the parameters a value is assigned, depending on the application. The higher the value, the greater is the attack surface and the higher is the risk for attacks, e.g., accessing the application as unauthenticated user has value 0, whereas accessing as authenticated or root user have value 5 and 10. The metric is calculated by calculating the Euclidian norm for each value of a parameter family. The value of the parameter family is the Euclidian norm calculated for each value of parameter in the family. The maximum attack surface is 60.79.

In the next sections, we discuss methods for reducing the attack surface regarding our goals and service interface design. The author of [9] suggests several methods for reducing the attack surface of an operating system. His 80/20 rule (according to the Pareto principle) to reduce the amount of running code contradicts our goal to not reduce functionality. Further, he offers no systematical way to find

 ISBN: 978-1-61208-376-6

code to remove. The methods for applying least privileges and reducing access for untrusted users mainly focus on the system running the application. According to this method, we suggest that for service interfaces least privileges also means reducing the amount of accessible operations. Authorization defines who shall access operations and is, therefore, our starting point for securing access by reducing the attack surface.

Reference [10] introduces an approach for removing or disabling unused code in operating systems. This corresponds to finding the 20 percent in the 80/20 rule of [9] and therefore, it aims to reduce functionality. Their general approach consists of two phases, the analysis and enforcement phase. In the analysis phase, unused code is found. The enforcement phase aims to avoid execution of unused code. They identify unused code by running the application and executing all available methods. Thus, this approach needs a running application and is firstly applicable in the implementation phase. We think that seldom-used or unused code could be avoided by considering security earlier.

Methods for reducing the attack surface of a web application based on the Top 10 vulnerabilities published by the OWASP are introduced by [8]. The authors use security measures mitigating these vulnerabilities. The Top 10 entries are related to security vulnerabilities in web applications and therefore, they do not have to be connected to the attack surface. Thus, not all of the applied measures, such as input validation and secret tokens, affect the attack surface directly. A systematical way to reduce the attack surface needs to ensure this reduction.

The discussed approaches aim to reduce the attack surface of in several ways. They do not offer a systematical way with concrete transformations to reduce the attack surface. Often the functionality of the application is reduced to ensure a smaller attack surface. Using security patterns is not part of any of these approaches. We tackle these limitations with our approach.

D. Web Services based on REST

According to the W3C, the term web service refers to a software system designed to support interoperable machine-to-machine interaction over a network [21][20]. It is frequently regarded more as a system's function of providing web access to its inner purpose rather than the whole system itself. Furthermore, a web application consists of web services, e.g., the web browser uses web services.

The W3C distinguishes two types of web services: Those using REST-compliant interfaces and those providing arbitrary access [20]. While the latter have been primarily used in the past – presumably because of the ease of implementation – RESTful interfaces become increasingly popular, mainly for their lightweight and universal deployment [21].

REST is an architectural style for the communication of web services proposed by Fielding [13]. It relies on existing standards, such as the Hypertext Transfer Protocol (HTTP), and defines six constraints for RESTful interfaces rather than concrete implementation specifications: The Client-Server principle, the concept statelessness, the usage of a cache, the uniformity of the interface, the layered system and the optional Code-On-Demand feature [13].

The uniform interface is the centerpiece of the REST architectural style: The interface describes every aspect trough resources. Every resource is identified by a unique address, which is in most cases a URI. Those resources are retrieved or manipulated via representations. A set of valid operations on these representations is available. Requests and responses are self-descriptive and semantic and hypermedia is used to describe them [13]. Hereby, a high degree of universality is achieved. However, it comes with a compromise in efficiency since the standardized information transfer leads to an overhead [21].

Since our approach alters the operations allowed on the resources, the compliance of the new interface to the uniformity concept is focus of validation.

III. Deriving Service Interface Methods from Authorization Patterns

In this section, we introduce our method to reduce the attack surface. We developed the approach based on the following assumptions and formulated goals 1 to 6. First, current methods for attack surface reduction have inacceptable deficits, such as decreasing functionality (goal 1 and 4). Second, non-security experts can apply the method and ensure security [4] (goal 2). Third, the method must be applicable at an early stage [3] (goal 3) on the KIT Smart Campus (goal 5, 6).

1. Security patterns shall be connected to software product quality not related to security.
2. A systematic way shall ensure certain quality attributes, including the attack surface.
3. The method shall be applicable in an early software development phase.
4. The method shall not reduce application functionality.
5. The method shall be applicable on web applications.
6. It shall apply for web services similar to the RESTful web services of the Participation Service.

Before introducing the method, we align the term attack surface according to ISO/IEC 25000 and 25010. The attack surface is an inherent characteristic of software, because it can be measured with several metrics introduced. Thus, speaking in the language of ISO/IEC 25000 [22], it is a software quality attribute. We suggest to assign it to the quality characteristic freedom from risk and its sub characteristic economic risk mitigation according to ISO/IEC 25010 [16]. Therefore, it belongs to the quality in use model.

 ISBN: 978-1-61208-376-6

Concerning the method, the starting point is the authorization of the application and corresponding security patterns. These patterns describe who can access resources in which way. Thus, authorization can be used to reduce the attack surface to exactly the functionality that shall be offered. Regarding the metric for web applications introduced in [7], our approach reduces the parameter family of access control. Other parameter families are not influenced by the approach and, thus, a reduction is ensured.

Our approach consists of the following three steps:

1. Set up an access control matrix.
2. Derive services from the access control matrix.
3. Create REST-compliant web services based on the derived services.

The access matrix of the first step contains resources and operations as columns and policy rules as rows. For every operation allowed by a policy rule, the corresponding table cell is filled with a dot. See Table 1, Table 2 and Table 3 as examples. In the second step, a web service is introduced for each resource. Its service interface has an operation for every table cell having at least one marked row. Figure 3 is an example for this. In the last step, the resulting web services are mapped to a REST-compliant web service. Each step is introduced in the next sections. First, the main idea of deriving technology independent web services and its service interfaces is explained in depth. Second, the mapping from the abstract web service to a REST-compliant web service.

A. Deriving Abstract Service Interfaces from Role-Based Access Control

A role-based scheme for the access control with n different resources and m roles can be depicted as a two-dimensional matrix (see example on Table 1). With the REST paradigm's resource-oriented interface style kept in mind, we assume that four operations are possible per resource: Creating, retrieving, updating and deleting (CRUD). A bullet indicates that the specified role is allowed to use the specified operation on the specified resource.

While in an ordinary RESTful implementation the interface would have provided access for all roles on all operations and all resources, our approach aims to reduce the overall number of accessible operations to a minimum. In the context of Table 1, this would lead to a reduction of the attack surface by the number of unfilled table cells.

TABLE I. EXEMPLARY MATRIX FOR RBAC WITH ROLES

	Resource #1				**Resource #2**			
	C	***R***	***U***	***D***	***C***	***R***	***U***	***D***
Role #1		●				●		
Role #2		●	●		●	●		●
Role #3	●	●	●	●	●	●		●

TABLE II. EXEMPLARY MATRIX FOR ABAC WITH EXPRESSIONS

	Resource #1				**Resource #2**			
canAccess(s, r, e)	***C***	***R***	***U***	***D***	***C***	***R***	***U***	***D***
attribute1(r)		●				●		
!attribute2(r)		●	●		●	●		●
attribute2(r) ∧ attribute3(r)	●	●	●		●	●	●	●

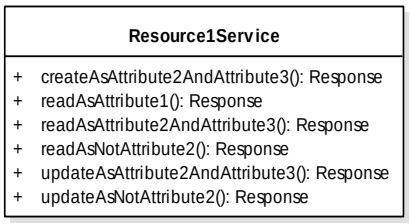


Figure 3. Entity Service for Resource #1 of Table 2

This is achieved by the creation of additional methods: Usually, one method is implemented for each operation on a resource. But by using our approach, methods are not only generated per operation but per operation and role (GetAsRole1, GetAsRole2, GetAsRole3, PostAsRole1, etc.). The difference is that each method can only validly be used by exactly one role and not by all roles possible. So far, the attack surface stays the same. The reduction is then reached by not implementing those methods that do not have a bullet in the access control matrix of, e.g., Table 1.

B. Deriving Abstract Service Interfaces from Attribute-Based Access Control

Applying the approach to ABAC extends the principles of the application to RBAC.

In the first step, all applicable operations for each resource of R are listed as columns in the access control matrix. Every policy rule of the canAccess() functions is listed as row. Every cell for which a canAccess() function is true is marked. A possible result shows Table 2.

Deriving the interface from Table 2 works similarly to the role-based approach: A service interface is created for each resource. In every service, operations are created for all allowed operation. Example operations from Table 2 are readIfIsAttribute1, updateIfIsNotAttribute2 and deleteIfIsAttribute2AndAttribute3 (see Figure 3). To prevent long and complicated method names, it is best practice to derive canAccess() rules from single attributes only whenever possible.

C. Application on Authorization Patterns

Sections III.A and III.B show how service interface methods can be derived for ABAC and RBAC. This section shows that the method is applicable for any kind of authorization.

In the sections on RBAC and ABAC, there are two limitations. First, the service interface methods are derived from access control matrixes for RBAC and ABAC. Second, because of the scenario and REST compliance, we used entity services [23] using only basic CRUD-operations. Both limitations are not necessary and can be generalized.

Concerning the first limitation, the abstract security pattern Authorization defines who may access protected resources in which way [2]. The access control matrix contains the description of the entity (who) on the first column of a row, the resource to access (what) on top of the

 ISBN: 978-1-61208-376-6

column and how the resource shall be accessible below the resource. Therefore, an access control matrix, as used it before, can be created for every kind of authorization.

Deriving the abstract service interfaces from these access control matrixes can be achieved as previously shown. Create a service interface for each service with operations combined to the permission. The name of the operations can be of any kind, thus not only CRUD-operations are applicable.

D. Maintaining REST Compliance

In order to comply with the previously presented REST constraints, we propose to not realize the derived service interface methods with extended HTTP-operations. Quite the contrary: REST relies on a defined and pre-known set of operations – namely GET, POST, PUT, DELETE, etc. when using HTTP. Introducing new operations restricts the API usage to insiders, thus, adversely affects the interface's uniformity and universality. It is also hardly possible in practice when using HTTP, since custom methods are not supported by browsers or most clients [21].

It is furthermore not advisable to realize the derived methods by using custom HTTP headers. To send a "X-Role: Administrator" header with every request seems practical on the first sight. But whitelist-based firewalls and proxy servers will skip those custom headers [24] limiting the API usage to clients that don't rely on a firewall. This kind of limitation is not acceptable.

However, a third way exists: We propose adding the service operation name to the request URI. Illustrating HTTP requests using the examples from above could then look like this:

```
POST  /resource1/?authorization=createAsRole3
DELETE /resource2/?authorization=deleteAsRole3
…
GET   /resource1/?authorization=readIfIsAttribute1
PUT   /resource1/?authorization=updateIfNotAttribute2
…
```

This is legal in the HTTP standard and does not violate the interface uniformity constraint of REST compliance. The server extracts the information from the parameter – a task possible with every framework and scripting language. Diligence is required in the implementation: The parameter must not have a fallback for an invalid or missing value. If that is the case, an error has to be thrown. Otherwise, the attack surface is not reduced for the simple reason that it does not differ from the traditional implementation.

TABLE III. ACCESS CONTROL MATRIX OF USER AND GROUP RESOURCES OF THE PARTICIPATION SERVICE

	User				Group			
canAccess(s, r, e)	***C***	***R***	***U***	***D***	***C***	***R***	***U***	***D***
Guest(s)	●							
Authenticated(s)		●			●	●		
User(s) = r			●					
User(s) = Owner(r)							●	
Admin (s)		●	●			●	●	

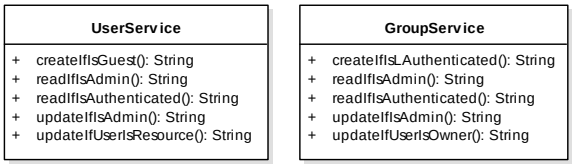


Figure 4. User and Group Service derived from access control matrix shown in table 3

An appropriate error communication for that case and for the case of using a not allowed permission on the specific resource, is responding with HTTP's status 405 Method Not Allowed. At first sight it seems uncommon to respond with a method-related error code to a missing or falsely specified parameter. However, as the parameter is merely an extension of the method according to the approach of this paper, it is suitable here. The list of "allowed methods" (more precisely: method and value for the authorization parameter) can be supplied in the body of the HTTP response. As a result it is possible to follow the Hypertext-As-The-Engine-Of-Application-State (HATEOAS) paradigm.

IV. EVALUATION

In this section, we apply the method on the Participation Service of the KIT-SC system, show the resulting web services and give an evaluation. The Participation Service is developed by seven students during a practical course at the KIT. The group was divided into two teams, one focusing on the HTML 5 frontend and the other focusing on the Java backend.

At the beginning the requirements for the service were collected. All required subjects S, resources R, environments E and their attributes SAn, RAn and EAn were identified and the access control matrix was built. Possible subjects are anonymous users and authenticated users. This publication demonstrates the method on the User and Group resources only, leaving out all other resources of the Participation Service for the sake of shortness.

According to the requirements, both, users and groups, can be created, edited and displayed. Deletion is solved by setting a status flag to deactivated, thus, by updating the resource. The access control matrix in Table 3 shows the authorization rules based on ABAC. Users can be created by guests. An authenticated user can read user account data, create groups and read them. The owner of an user or group account can update its information. User with the admin flag are allowed to read and update users and groups.

Figure 4 shows the derived abstract service interfaces from the access control matrix of Table 3. For each resource a service is modeled with the operations according to the access control matrix. This implies, that the services do not have operations for deleting the resources, because no authorization rule exists for this operation. Typically the delete operation would still be implemented, but inaccessible due to the enforced authorization. According to [9], this mapping is a reduction of the attack surface.

 ISBN: 978-1-61208-376-6

The abstract service interfaces are then mapped to the REST services with URLs as follows:

For the User Service:
POST /user/?authorization=createIfIsGuest
GET /user/?authorization=readIfIsAdmin
GET /user/?authorization=readIfIsAuthenticated
PUT /user/?authorization=updateIfIsAdmin
PUT /user/?authorization=updateIfUserIsResource

For the Group Service:
POST /group/?authorization=createIfIsAuthenticated
GET /group/?authorization=readIfIsAdmin
GET /group/?authorization=readIfIsAuthenticated
PUT /group/?authorization=readIfIsAdmin
PUT /group/?authorization=readIfUserIsOwner

The Spring Security project was chosen to enforce the authentication and authorization of the KIT-SC. Authorization is implemented by adding the annotation *PreAuthorize* to each entry point of the corresponding URL. These annotations contain the access policies as Spring EL expressions, which are evaluated by Spring Security to enforce access control. Spring EL offers the possibility to state expressions on the attributes of resource and subject. Thus, the patterns delivered in the request, formerly introduced by our method, can be used to formulate the Spring EL statement.

Using the approach of this paper in combination with Spring Security proved to be a good choice for many reasons:

The attack surface metric of [7] has been improved. The access control parameter *rights* of the parameter family *access control* has been reduced from 10 to 0 or 5, depending on the privileges of the operation.

Moreover, enforcing the authorization is easier, because testing functionality and access decision can be combined. For example look at the third row of Table 3. The user shall only be able to update its account. This constraint can be implemented and tested quite easily. Further, for enforcement of this policy, just the ownership has to be validated. This is quite easy, because the user data is delivered in the request. Without this limitation, the information must be collected separately. Thus, with a generic update operation, for each user touched by an operation call, every policy has to be enforced and corresponding data has to be fetched.

Additionally, frontend developers benefited from associating the authorization to HTML forms, buttons and links. By choosing which operation to call, they get sensitized to security. Following the principles of secure interaction design [12], they added confirmation messages, warnings, colors and icons to the user interface according to the security level of the different operations used.

V. LIMITATIONS

Regarding goal 6, the method is based on at least three assumptions. First, the authorization may be exposed to the users of the web service and, thus, also to attackers. This may be a threat for the web service or even a problem regarding federation. We assume, that the system is secure, even if the attackers have this information, according to Kerckhoffs's principle for crypto-systems [27]. Thus, this information may be exposed without making the web service insecure. Despite this, exposing the information can be impossible. In this case, the web service operation name has to be obfuscated or the introduced method cannot be applied.

Second assumption is, that the count of authorization rules for a single web service does not exceed. The policies defined by ABAC can be fine-grained using complex expressions. All these fine-grained policies lead through our approach to at least one service interface operation. In large systems this may be a great overhead. Many operations with potentially long names could be introduced. For example operations with similar functionality need an agnostic internal method to avoid redundancy and more methods and tests have to be implemented by the developers.

Third assumption raised by goal 6 is, that the authorization rules do not change periodically or often. A change in the authorization rules may lead to changes in the web service operations and can cause changes in systems using the web service, when using the method. This depends on the change and on the mapping of the abstract interface to the language depend web service interface. In our REST mapping, the URL does not change, but a new parameter may be introduced. In this case, changing authorization rules do not lead to changes in systems using the web service. Even so, the web service has to be enhanced including overhead.

Additionally, the approach introduced is systematical, but we have not used a language to describe access control policies. This is because we could not find a suitable language. Possible candidates are the Unified Modeling Language with SecureUML [4] and UMLsec [25] or the Ponder Policy Specification Language (PPSL) [26]. But UMLsec and SecureUML need to be enhanced, to support every kind of authorization. PPSL is not based on the UML and has no visual representation, but we think both are important prerequisites so that the approach is used.

Another limitation concerning REST is the restricted functionality of HTTP's OPTIONS method. An OPTIONS call to a resource is responded with a list of allowed methods on that resources and using one of them should not result in a 405 Method Not Allowed error code. However, after applying this paper's approach, the method name is not sufficient to formulate valid requests – information about valid authorization parameter values are required (see Section III). The response is expressed in a list of comma-separated HTTP methods and there seems to be no possibility to additionally provide parameter values.

VI. CONCLUSION AND FUTURE WORK

We introduced a new way of designing interface methods by using security patterns. For this method, we showed that the attack surface on the interface is minimized according to the least privilege needed. Additionally, we showed how to combine the method with the REST paradigm and therefore, create REST-compliant web services.

 ISBN: 978-1-61208-376-6

The application of the method was shown within the Participation Service of the KIT-SC. In this application, at least the disadvantage of creating many interface methods by applying our approach arose. However, the attack surface has been reduced. By giving a mapping from the technology independent web service to a RESTful web service, the approach facilitates a REST-compliant Participation Service.

The approach gives software architects the possibility to improve the safety of web services using authorization patterns. They can follow instructions to improve quality attributes of the application in a systematic way without having a security background or knowledge.

Software developers using the derived service interface are aware of the privileges when using interface methods. This increases the security according to secure interaction design. Furthermore, the implementation of the service interface can be easier tested, because the authorization offers constraints for the operation to be implemented.

The disadvantage of creating many service interface methods may be the focus of future work. For instance, this phenomenon could be avoided by combining similar rights for the same object to one service interface method. Another starting point for future work is to research the advantages of the static in contrast to the dynamic access decisions. This can lead to an improved performance, improved security through easier testing and easier externalization of access decisions.

Our main goal is to combine the usage of security patterns with quality attributes. This can lead to more precise predictions on the quality of software. Therefore, non-functional requirements of stakeholders can be considered during the design of an application. By offering systematical methods, the quality can be ensured among the phases of the software development.

REFERENCES

[1] G. McGraw, "Software Security," IEEE Security & Privacy, pp. 80-83, Mar.-Apr. 2004, doi:10.1109/MSECP.2004.1281254.

[2] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, "Security Patterns: Integrating Security and Systems Engineering," John Wiley & Sons, Dec. 2005, ISBN: 978-0-470-85884-4.

[3] B. Alshammari, C. Fidge, and D. Corney, "Security Metrics for Object-Oriented Designs," IEEE 21. Australian Software Engineering Conference (ASWEC), Apr. 2010, pp. 55-64, doi:10.1109/ASWEC.2010.34.

[4] D. Basin, J. Doser, and T. Lodderstedt, "Model Driven Security: from UML Models to Access Control Infrastructures," ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 15, Jan. 2006, pp. 39-91, doi:10.1145/1125808.1125810.

[5] P. Manadhata, K. Tan, R. Maxion, and J. Wing, "An Approach to Measuring A System's Attack Surface," Carnegie Mellon University, Aug. 2007 [online]. Available from: http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA476805 [retrieved: 23.09.2014]

[6] B. Alshammari, C. Fidge, and D. Corney, "Security Metrics for Object-Oriented Class Designs," IEEE 9th International Conference on Quality Software, Aug. 2009, pp. 11-20, doi:10.1109/QSIC.2009.11.

[7] T. Heumann, J. Keller, and S. Türpe, "Quantifying the Attack Surface of a Web Application," In Proceedings of Sicherheit 2010, vol. 170, 2010, pp. 305-316, ISBN: 978-3-88579-264-2.

[8] G. Sumit, R. K. Nabanita, Mukesh, S. Saurabh, and M. Pallavi, "Reducing Attack Surface of a Web Application by Open Web Application Security Project Compliance," Defence Science Journal, vol. 62(5), Sep. 2012, pp. 324-330, doi: 10.14429/dsj.62.1291.

[9] M. Howard, "Attack Surface – Mitigate Security Risks by Minimizing the Code You Expose to Untrusted Users," MSDN Magazine, November 2004. [Online]. Available from: http://msdn.microsoft.com/en-us/magazine/cc163882.aspx [retrieved: 23.09.2014]

[10] A. Kurmus, A. Sorniotti, and R. Kapitza, "Attack Surface Reduction For Commodity OS Kernels: trimmed garden plants may attract less bugs," in Proceedings of the Fourth European Workshop on System Security (EUROSEC '11), Apr. 2011, pp. 1-6, doi:10.1145/1972551.1972557.

[11] A. Bartel, J. Klein, and M. Monperrus: "Automatically Securing Permission-Based Software by Reducing the Attack Surface: An Application to Android," in Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE 2012), Sep. 2012, pp. 274-277, doi: 10.1145/2351676.2351722.

[12] K. Yee, "Guidelines and Strategies for Secure Interaction Design," Security and Usability: Designing Secure Systems That People Can Use, pp. 247.273, Aug. 2005, ISBN: 978-0-596-00827-7.

[13] R. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Dissertation, University of California, Irvine, 2000, ISBN: 0-599-87118-0.

[14] M. Gebhart, P. Giessler, and P. Burkhardt, "Quality-Oriented Requirements Engineering for Agile Development of RESTful Participation Service," in press.

[15] E. Yuan and J. Tong, "Attribute Based Access Control (ABAC) for Web Services," in Proceedings of the International Conference on Web Services (ICWS), Jul. 2005, pp. 561–569, doi:10.1109/ICWS.2005.25.

[16] ISO/IEC, "ISO/IEC 25010:2011(E) Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models," 2011.

[17] R. Steinegger, "Authentication and authorization patterns in existing security frameworks [Authentifizierungs- und Autorisierungsmuster in bestehenden Sicherheits-Frameworks]," diploma thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2012. German.

[18] C. Pautasso and E. Wilde, "Why is the Web Loosely Coupled? A Multi-Faceted Metric for Service Design," in Proceedings of the 18th international conference on World wide web (WWW '09), Apr. 2009, pp. 911-920, doi:10.1145/1526709.1526832.

[19] A. Caballero, "Computer and Information Security Handbook," Morgan Kaufmann Publications, 2009, ISBN: 978-0123743541.

[20] W3C, "Web Services Glossary," Feb. 2004. [Online]. Available from: http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice [retrieved: 23.09.2014]

[21] L. Richardson and S. Ruby, "RESTful Web Services", O'Reilly Media, May 2007, ISBN: 978-0596529260.

[22] ISO/IEC, "ISO/IEC 25000:2005(E) Software Engineering – Software Product Quality Requirements and Evaluation (SQuarE) – Guide to SQuaRE," 2005.

[23] S. Cohen, "Ontology and Taxonomy of Services in a Service-Oriented Architecture," Microsoft Architect Journal, Apr. 2007.

 ISBN: 978-1-61208-376-6

[24] A. van Kesteren, “HTTP methods, Web browsers and XMLHttpRequest,” Oct. 2007. [Online]. Available from: http://annevankesteren.nl/2007/10/http-method-support [retrieved: 23.09.2014]

[25] J. Jürjens, “UMLsec: Extending UML for Secure Systems Development,” Lecture Notes in Computer Science, vol. 2460, pp. 412-425, Sep, 2002, doi:10.1007/3-540-45800-X_32.

[26] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, “The Ponder Policy Specification Language,” in Proceedings of the International Workshop on Policies for Distributed Systems and Networks (POLICY '01), Jan. 2001, pp. 19-37, ISBN: 3-540-41610-2.

[27] Auguste Kerckhoffs, “La cryptographie militaire,” Journal des sciences militaires, vol. IX, Jan. 1883, pp. 5–38.

 ISBN: 978-1-61208-376-6

Evaluation of Vehicle Diagnostics Security – Implementation of a Reproducible Security Access

Martin Ring, Tobias Rensen and Reiner Kriesten

University of Applied Sciences Karlsruhe
Karlsruhe, Germany
Emails: {rima0003, reto1014, krre001}@hs-karlsruhe.de

***Abstract*—Modern cars typically possess a network of numerous Electronic Control Units (ECUs) which are connected with each other by several bus systems. In addition to the necessary on-board communication by means of which the ECUs exchange information without any influence from outside, there is a strong need for interaction with off-board systems. In this context, the vehicle diagnostics can be mentioned as a significant example. It is highly important that the connection between diagnostic testers and the car is secured against unauthorized access. This paper examines the development of a procedure as well as a software tool for granting a reproducible access to individual car ECUs without any professional testers. If this access can be achieved by self-developed tools, a possible security danger exists as malicious diagnostic routines (not existing in professional car testers) can be activated by using this access. If the ways to achieve this access are known, it is possible to work on improving the defence.**

Keywords–security access; safety; diagnostics security; data busses; communication standard.

I. INTRODUCTION

The increasing number of vehicle electronics [8] in modern cars leads to a permanently rising focus on safety and security aspects. Whereas safety can be described as the fact that the vehicle acts adequately in critical situations, security addresses the maturity of the car system against attacks from outside.

Concerning the safety issues, the International Standardization Organisation (ISO) has released the automotive specific standard ISO 26262 [17]. However, the standardization of security issues has not yet reached the same level.

Especially, the connectivity of modern cars to the outside world is a critical factor. Use cases like diagnostics exchange, navigation information, interaction with mobile devices and personalized services can be easily found. [3][4][5][12]

The easiest way to interact with the automotive network is via the On-Board-Diagnostics (OBD) connector. This connector serves as central access to all ECUs available in a car. For safety critical diagnostic functions, a so-called security access is implemented in the diagnostics standard [18].

We investigated if a self-written program can reliably achieve security access to modern vehicles by means of seed and key methods. Figure 4 describes the principles behind this practise. After a security request from the tester a random number, a so-called seed, is sent back from the vehicle ECU. Afterwards, the tester performs a secret coding algorithm and sends back the calculated key which is evaluated in the ECU [18]. The respective approach can be briefly described as follows:

- Recording of the security access between vehicles and testers in order to get the overall protocol sequence and information.
- Implementing of a software tool which replaces the car and requests keys from the tester in order to get the possible seed and key pairs.
- Testing the seed and key pairs for their reliable use. This implies in particular that they are independent of date, vehicle and ECU specific information like the Vehicle Identification Number (VIN).

Before the diagnostic data can be analysed, it is important to know how to interpret the payload in the CAN message, which is described in Section III. Section IV describes the fundamentals needed to simulate an ECU. The simulation of the ECU is described in Section V. Lastly, Section VI shows the analysis of the key exchange and which parameters are significant for its calculation.

 ISBN: 978-1-61208-376-6

II. Related Work

Only a small number of scientific writings are available on this subject. Especially, works focusing on a reliable procedure for gaining security access to the ECUs/network of an arbitrary car are rare. The related writings [3][4][5][12] mainly describe how to provoke a security hazard by means of additional components or a self-programmed code executed on existing components. This paper examines the possibility of provoking a hazardous situation by gaining access to needed software implementations, e.g., the ventilation of the Anti-lock Braking System (ABS) unit.

III. Basics on automotive embedded systems

This section describes the fundamentals on embedded automotive systems needed for understanding this paper.

A. Vehicle network: lower protocol layers

1) Electric architecture: Modern cars possess several bus systems for the communication between the ECUs, sensors and actuators. According to the AUTOSAR Standard [14], these devices are categorised in multiple networks, like body and comfort network,powertrain network or the infotainment network, seeFigure 1. The underlying bus system is further dependent on the necessary data rate, cost aspects, real-time-abilities, etc. However, the Controller Area Network (CAN) bus [16] is still the most popular bus in modern vehicles. As the diagnostics protocol usually is embedded in the CAN bus protocol, the latter is described more detailed in the following paragraph.

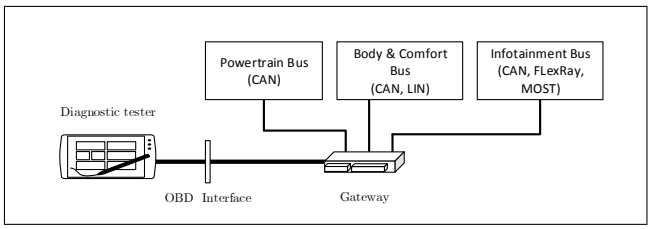


Figure 1. Vehicle network example.

2) Information CAN bus: The CAN bus is the most popular bus system in modern vehicles. In the U.S., it even is the standard for the OBD diagnostic since 2008. Regarding the physical characteristics, it uses a differential data transmission in order to resist electrical disturbances (to be seen as safety feature) and allows data rates up to 500 kbit/s [4].

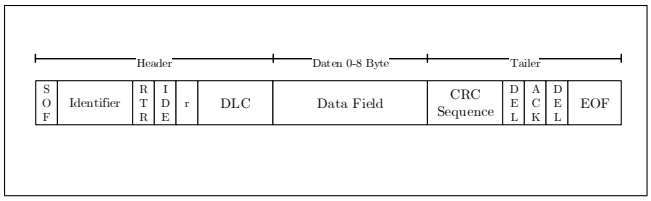


Figure 2. CAN packet structure [4].

Figure 2 shows the structure of a CAN message according to the standard ISO 11989. The most important parts of the message regarding diagnostic messages are the ID field containing the address of the ECU and the diagnostic payload located in the data field.

B. Transport protocol

The transport protocol is standardized in the ISO 15765-2 [19] and is used for diagnostic purposes. This protocol is located one layer above the CAN protocol and allows upper services to transmit information with a data length of possibly more than 8 byte. The information of the Transport Protocol (TP) found in the most significant bytes of the CAN data field. These bytes are called Protocol Control Information (PCI). There are four different types of messages, the first nibble of the CAN data field contains the type information [5][11].

- 0_h Single frame: contains the entire payload (less than 8 byte). The second nibble shows how much data the packet contains.
- 1_h First frame: this is the first frame of a multi-packet payload. The next three nibbles contain the number of the whole diagnostic data.
- 2_h Consecutive frame: this message contains the rest of the multi-packet payload. The second nibble contains the order of the sent message.
- 3_h Flow control frame: this message is sent from the receiver of the multi-packet payload. This message is sent after the first frame [11].

C. Vehicle networks: upper protocol layers

1) Diagnostic protocol standards (Application Layer): There are two popular diagnostic protocols: one is the Keyword Protocol (KWP) 2000 which is standardized in the ISO 9141 and ISO 14230; the other one is the Unified Diagnostic Services protocol (UDS) [18] which is standardized in the IS0 14229. The operation of both diagnostic protocols is almost identical. KWP 2000 was

 ISBN: 978-1-61208-376-6

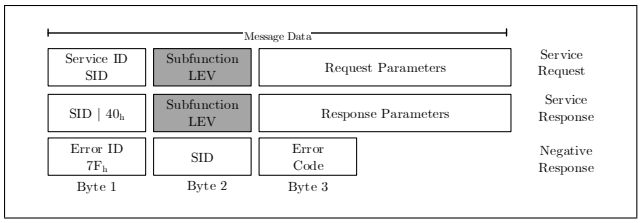


Figure 3. UDS diagnostic protocol [13].

originally designed for the proprietary bus system K-Line and is not used in modern cars anymore. Both protocols work with Service Identifiers (SID). Every SID represents a different action from an ECU which can be specified by its LEVs (subfunction levels); see Figure 3.

The provided services are defined in the standards. The services can be selected by the SID and LEV. These two bytes are the first two diagnostic data bytes of the message. There are three types of messages:

- The request message. This message is sent by the tester with the desired service.
- The response message. This message is sent from the ECU. The SID of the response message is calculated by logical or-linking the SID of the request message and 40_h (e.g., $27_h|40_h = 67_h$).
- The error message starts with $7F_h$, which is followed by the SID of the request and an error code with a length of one byte, as seen in Figure 3.

The control units communicate only after receiving a request from the diagnostic tester. There is a clear distribution of roles, in which the tester assumes the role of the client and the control unit works as server. This communication principle is also called request and response.

D. Security Access in the diagnostic protocol

Today‘s security access is defined in the UDS standard. To access safety-critical-functions, the tester asks the ECU for a seed. After receiving this seed, the tester computes the according key, which is sent back to the ECU. If the received key is consistent with the expected key, access is granted [13]. Seed and key lengths, as well as the algorithm to compute the key, are not specified in the standard. Every vehicle manufacturer can implement an arbitrary seed length and algorithm. It is also not standardized if the seed is static or alternating. If the security access is used, the standard specifies that there are special LEVs to send the request for a seed and special LEVs for sending the key. All those subfunction levels can be found in the security access service (SID: 27_h).

requestSeed: LEV 01_h, 03_h, 05_h, $07_h - 5F_h$
sendKey: LEV 02_h, 04_h, 06_h, $08_h - 60_h$ [19]

The process of the Security Access is shown in Figure 4.

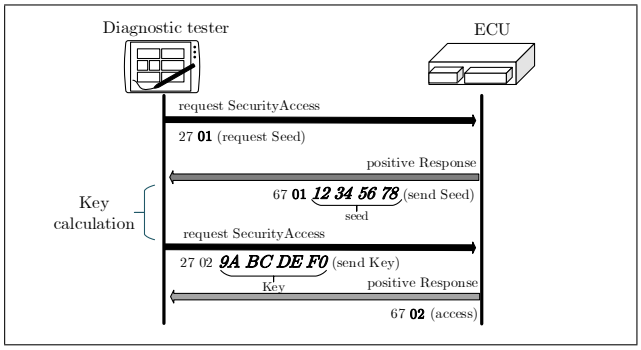


Figure 4. Security access timing sequence [11].

The message structure of the diagnostic messages from the tested vehicles follows the standardized protocols (with a few exceptions). The first byte of a single message contains the information about the transport protocol. In the message (listed below), the value is 02_h. The zero (first nibble) stands for a single message and the two (second nibble) for two diagnostic data bytes. The second byte contains the SID and the third is the LEV (service and sub function).

Tester request data: **02 10 92** 00 00 00 00 00
ECU response data: **02 50 92** 38 37 30 32 39

IV. TECHNICAL ACCESS SETUP FOR THE SECURITY EVALUATION

This section describes the physical setup in order to measure and record the diagnostic communications and the decoding strategy of the messages according to the given UDS standard.

In order to record the communication between the tester and individual vehicles, an additional client was added to the diagnostics line, a bus analysis tool running on the attached PC; see Figure 5 [15].

Thus, the existing communication between different cars and the tester could be easily recorded. In the second step the bus analysis tool was used for the simulation of

 ISBN: 978-1-61208-376-6

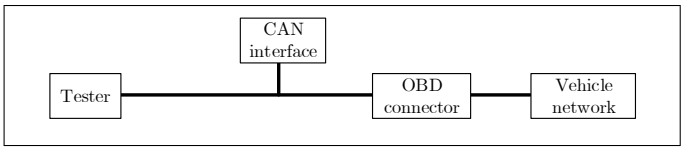


Figure 5. Recording strategy for diagnostics communication.

TABLE I. COMMUNICATION FROM BEGIN TO SECURITY ACCESS.

CAN Data	description	send from
02 **10 92** 00 00 00 00 00	session request	tester
02 **50 92** FF FF FF FF FF	session response	ECU
02 **1A 87** 00 00 00 00 00	session ECU info	tester
10 16 **5A 87 01 22 05 14**	send ECU Info1	ECU
30 08 28 00 00 00 00 00	send other parts	tester
21 **FF 07 09 09 43 00 32**	send ECU Info2	ECU
22 **30 34 35 34 35 33 38**	send ECU Info3	ECU
23 **33 32** FF FF FF FF FF	send ECU Info4	ECU
02 3E 01 00 00 00 00 00	tester present	tester
02 7E 00 00 00 00 00 00	tester present	ECU
02 **27 01** 00 00 00 00 00	Security req.	tester
05 **67 01 F0 5E** 00 00 00	send Seed	ECU
04 **27 02 92 16** 00 00 00	send Key	tester
03 **67 02** 34 00 00 00 00	pos. access	ECU

the car. To be more precise, the bus analysis tool provides the messages which originally came from the real car; see Figure 6. It further has to be noticed that there is a reason for simulating the vehicle and not the tester; while having only a few attempts for the security access to car ECUs (afterwards, they deny any further access), professional testers can be stimulated an infinite number of times as in a typical environment they have to serve numerous vehicles and have to be permanently available.

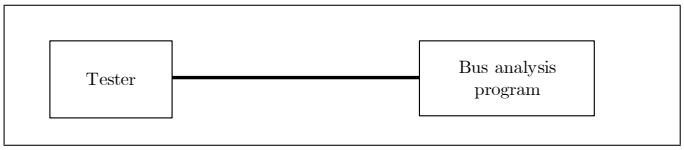


Figure 6. Simulation mode.

Table I shows an exemplary protocol sequence at the beginning of a security session. First, a handshake between the tester and the ECU is initiated by the tester including the exchange of specific ECU information. Afterwards, the seed and key messages appear for the authorization of the security access. In this context, it still has to be mentioned that most of the message data is standardized according to the UDS protocol.

A. Vehicle selection

The choice of the investigated vehicles was influenced by the fact that since 2008 cars are offering the UDS protocol being typically embedded within the CAN bus. Considering this limiting conditions, six vehicles produced by four different manufacturers have been randomly chosen.

As a first result, it was not possible to perform a security access for one specific car platform as the corresponding services have not been implemented in the tester. In this case, only diagnostic routines which do not rely on the security access could be executed, e.g., reading/deleting error codes. Regarding all other tested car manufacturers, the security access could be recorded. To proceed, emphasis was put on two different cars of one manufacturer. The reason for this decision is mainly that this manufacturer implemented the security access according to the UDS standard. The security access was not implemented by all tested manufacturers, even though there is a standard [18] which recommends this access for certain safety critical functions. access to this vehicles was unlimited.

B. Use cases for the execution of the security access

Table I displays the dial-up of the connection and the exchange of the seed and key data. Both the seed and the key are two bytes long which is car specific and not described in the standard. For both tested vehicles of this brand, the dial-up connection between the tester and the vehicle and also the security access are identical to the one shown in Table I, only the seeds, keys and ECU information differ. In the first vehicle, the security access appeared in the ABS ECU after selecting a specific safety function of this ECU. For non-safety-relevant diagnostic functions there was a request for the security access from the tester; see Table II. In contrast, the ECU obviously did not insist on the secure access, which affects the protocol sequence in the following way: the ECU sends zero information as key data (no security access needed) being also responded with zero bytes from the tester.

TABLE II. SECURITY ACCESS WITH ZERO BYTES.

CAN Data	description	send from
02 **27 01** 00 00 00 00 00	Security req.	tester
05 **67 01 00 00** 00 00 00	send zeros	ECU
04 **27 02 00 00** 00 00 00	send zeros for key	Tester
03 **67 02** 34 00 00 00 00	pos. access	ECU

 ISBN: 978-1-61208-376-6

V. ECU SIMULATION FOR A REPRODUCIBLE SECURITY ACCESS

We implemented the communication behaviour of both ECUs (ABS / Airbag) existing in the different vehicles of which a security access was recorded; see Figure 5. The GUI of the simulation allows the selection of a car and the desired ECU. If a security access has been successfully performed the GUI displays a notification and the used seed and key data; see Figure 7. The seeds sent to the tester are arbitrarily chosen by the simulation program, so $2^{16} = 65536$ seed and key pairs exist, due to it's 16 bits length. Further, they can be written in a text-file before starting the simulation. After all seeds have been sent, the program generates a new file which stores the used seeds and its received keys. As already mentioned, the data exchange works only on request, which means that the whole simulation is controlled by the diagnostic tester.

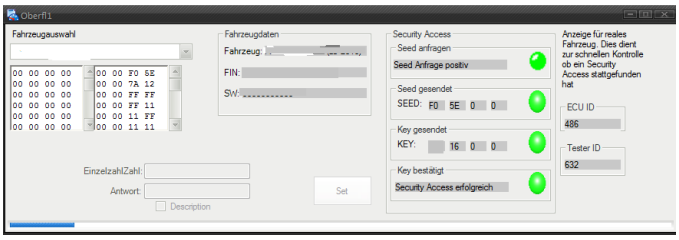


Figure 7. Panel for handling the ECUs and the security access.

VI. SECURITY ACCESS ANALYSIS

In order to implement a tool which can reliably unlock different vehicles of the same model, it has to be analysed if the key algorithm is reproducible. This implies, in particular, the independence of the actual time and vehicle specific values such as the Vehicle Identification Number (VIN). In the following, the key algorithm is evaluated regarding its independence of date, VIN and ECU data.

A. Data independence

The same seed was sent to the tester twice on different days. Each time the received key was identical. This shows that the key calculation is independent of date and time. Surely, this behaviour could be anticipated as it is unlikely that both vehicle and tester share the same timebase and use it for the seed/key calculation.

B. VIN independence

In the tester, a VIN can be selected in order to determine the associated car. Therefore, one can assume that the seed and key data are dependent on the VIN. Thus, the traffic between the tester and the ECU was analysed and no VIN information was found. Furthermore, the tester was provided with two different VINs and access was requested using the same seed. As a result, the keys again did not differ. To conclude, the security access is independent of the VIN.

C. Independence of ECU specific data

In order to assure that the key is only dependent on the given seed it is necessary to prove that the ECU specific information does not change the key data. Again, the simulation program twice requested keys while changing the ECU specific data; see Table III. Once more, the expected behaviour of independence could be confirmed.

TABLE III. CHANGED ECU INFORMATION.

CAN Data	description	send from
10 16 5A 87 01 22 05 14	send ECU Info1	ECU
21 FF 07 09 09 43 00 32	send ECU Info2	ECU
22 30 34 35 34 35 **33** **38**	send ECU Info3	ECU
23 **33** **32** FF FF FF FF FF	send ECU Info4	ECU

VII. CONCLUSION AND FUTURE PROSPECTS

Evaluating the communication between modern vehicles and diagnostic testers enabled us to develop a software tool which grants security access to special electronic control units of modern vehicles. Using the developed software tool it was possible to extract the keys from the tested cars semi-automatically. As the respective process is not conducted fully automatically, the extraction of all keys for 16-bit seed and key pairs would take approximately 110 working hours. This workload could be reduced by an additional automation of the tester handling. It is also possible to generate a program which determines the possible algorithms of a given input and output vector. In a testrun, only 50 pairs were needed to determine the respective algorithm. The fact that it was possible to achieve security access can be considered as crucial because this access can be used to cause security critical and therefore dangerous conditions or unintended actions while the vehicle is in motion. Thus, it is recommended to improve the defence.

 ISBN: 978-1-61208-376-6

REFERENCES

[1] K. Beiter, C. Rätz, and O. Garnatz, "Gesetzliche On-Board-Diagnose und ODX (Statutory On-board Diagnostics and ODX)." [Online]. Available: http://vector.com/portal/medien/diagnostics/odx/Gesetzliche_OnBoard_Diagnose_und_ODX.pdf-2014.07.21

[2] K. Borgeest, Elektronik in der Fahrzeugtechnik (Electronics in Vehicle Technology). Vieweg Verlag, 2007.

[3] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Comprehensive Experimental Analyses of Automotive Attack Surfaces." [Online]. Available: http://www.autosec.org/pubs/cars-usenixsec2011.pdf-2014.07.21

[4] K. Koscher, A. Czeskis, F. Roesner, S. Patel, and T. Kohno, "Experimental Security Analysis of a Modern Automobile," in 2010 IEEE Symposium on Security and Privacy, 2010. [Online]. Available: http://www.autosec.org/pubs/cars-oakland2010.pdf-2014.07.21

[5] C. Miller and C. Valasek, "Adventures in Automotive Networks and Control Units." [Online]. Available: http://illmatics.com/car_hacking.pdf-2014.07.21

[6] T. Nosper, "Cotroller-Area-Network." [Online]. Available: http://www.hs-weingarten.de/nosper/public/Download/Kapitel202.720CAN-Neues20Layout.pdf-2014.07.21

[7] K. Reif, Automobilelektronik (Automotive Electronics). Vieweg + Teubner Verlag, 2012.

[8] H. Richter, "Elektronik und Datenkommunikation im Automobil (Electronics and Data Communication in Automotive Applications)," Institut fr Informatik, Technische Universitt Clausthal, Tech. Rep. [Online]. Available: http://www.in.tu-clausthal.de/fileadmin/homes/techreports/ifi0905richter.pdf-2014.07.21

[9] F. Schäfer, OBD Fahrzeudiagnose in der Praxis (OBD Vehicle Diagnosis in practice). Franzis Verlag, 2012.

[10] T. Strang and M. Röckl, "Vehicle Networks CAN-based Higher Layer Protocols," 2008. [Online]. Available: http://www.sti-innsbruck.at/sites/default/files/courses/fileadmin/documents/vn-ws0809/03-vn-CAN-HLP.pdf-2014.07.21

[11] J. Supke and W. Zimmermann, "Diagnosesysteme im Automobil (Diagnostic Systems in Automobiles)." [Online]. Available: http://www.emotive.de/documents/WebcastsProtected/Transport-Diagnoseprotokolle.pdf-2014.07.21

[12] M. Wolf, A. Weimerskirch, and T. Wollinger, "State of the Art: Embedding Security in Vehicles," EURASIP Journal on Embedded Systems, April 2007. [Online]. Available: http://downloads.hindawi.com/journals/es/2007/074706.pdf-2014.07.21

[13] W. Zimmermann and R. Schnidgal, Bussysteme in der Fahrzeugtechnik (Bussystems in Automotive Engineering). Vieweg Verlag, 2007.

[14] Release 4.1 Overview and Revision History, AUTOSAR Std. [Online]. Available: http://www.autosar.org/fileadmin/files/releases/4-1/AUTOSAR_TR_ReleaseOverviewAndRevHistory.pdf-2014.07.21

[15] Handbuch CANoe (CANoe Manual), Vector Informatik GmbH.

[16] ISO 11898 CAN, ISO Std.

[17] ISO 26262 Safety, ISO Std.

[18] ISO 14229 Unified diagnostic services (UDS), ISO Std.

[19] ISO 15765-3 Implementation of Unified Diagnostic Services (UDS on CAN), ISO Std.

[20] CAPL Function Reference Manuel, Vector Informatik GmbH, November 2004.

[21] Programming with CAPL, Vector Informatik GmbH, Dezember 2004.

 ISBN: 978-1-61208-376-6

An AMI Threat Detection Mechanism Based on SDN Networks

Po-Wen Chi*, Chien-Ting Kuo*†, He-Ming Ruan*, Shih-Jen Chen†, and Chin-Laung Lei*
*Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
Email: {d99921015, d98921027, d97921030, cllei}@ntu.edu.tw
†CyberTrust Technology Institute, Institute for Information Industry, Taipei, Taiwan
Email: {ctkuo, sjchen}@iii.org.tw

***Abstract*—The security of Advanced Metering Infrastructure (AMI) systems draws more and more attention nowadays. Intrusion detection systems are often deployed on the backhaul network to protect the AMI head-end system. In this paper, we proposed an efficient way to build threat detecting mechanism in AMI systems with the help of software defined networks (SDN). Moreover, we also enhance the OpenFlow architecture to provide more powerful detection mechanism to secure the AMI system. The proposed solution not only enhances the security of AMI systems, but also preserves the traffic quality of this structure.**

Keywords–AMI; SDN; Specification-based detection

I. INTRODUCTION

Recently, the AMI system, which serves as a key role in Smart Grid, became popular due to the benefits it could bring. This new infrastructure enables the exploration of the possibilities of energy utilization by providing certain communication and control functionalities. However, AMI introduces new security challenges while providing various benefits due to semi-open networks, improper security mechanisms and immature hardware design for AMI devices. There are already many researches which introduce security issues in AMI systems, such as [1][2]. The essence of AMI is a vast and distributed sensor system tethered by the backhaul network and some neighborhood networks (NANs) which can be open networks or closed ones. It implies that anyone on the backhaul might find their way to interfere with the AMI, especially the Internet service providers (ISPs) who can possibly control partial or all of the connections in an AMI system. Thus, we will focus on the security issue in the backhaul network in this paper.

Traditional approaches to protect a device in an IT system could be cryptographic tools such as mutual authentication that ensures the identities of each end in a communication, encryption and key management, which enforces the access control over specific storage media, or digital signature, which guarantees the source of a message. However, any of the cryptographic measures require relatively powerful hardware, and this implies that the cost of devices will be anything but cheap. But the extremely large scale of AMI systems limits the budget of the devices, and further constrains the capability of the devices and the available protection approaches. Under such dire condition, monitoring the security status of the AMI system becomes a practical and economical solution. With the status of the system security at hand, one can then address and react to security events more effectively while the cost will be much economical than traditional cryptographic protection measures.

Traditional IDS systems mostly take signature-based detection as their core technology, which detects malicious activities by describing these activities as signatures beforehand. Snort [3] is the most popular open-sourced project of this kind of IDS. However, this kind of detection alone is not sufficient since it is difficult to list all malicious behaviors and nothing can be done about unknown attacks. In order to provide a more secure network environment, specification-based detection was proposed [4][5]. With the specifications to describe the normal activities, the IDS can collect all events which do not meet the requirement of the AMI system. Thus, the administrator can decide if the network is under attacks by comprehensive analysis of events. Therefore, the administrator can still be aware of unknown attacks under the assistance of the specification-based technology.

In addition to the specification-based detection system, we observe that a new network trend, Software Defined Networking (SDN), is changing the network architecture. The SDN could be a proper primitive for an AMI system due to the vast and distributed nature of the AMI, which results in the need of efficient management mechanisms to secure the AMI systems. With the features of the SDN, it reveals a novel approach for the administrator to dynamically perform flow-level management over his own network. We believe that in the near future, more and more networks will be SDN, including AMI backhaul networks. So, we are motivated to build an IDS in SDN-based AMI backhaul networks.

In this paper, we integrate the SDN technology with IDS in the AMI system. First, we will show how to integrate traditional IDS, Snort, with SDN efficiently by offloading some checking rules from Snort to OpenFlow switches. Therefore, IDS will afford more throughputs than legacy architecture. Moreover, we propose an enhanced OpenFlow technology in which OpenFlow switches are improved by additional specification checking agents. By using our enhanced OpenFlow switches, the specification checking rules can be quickly deployed to each transmission path node in the AMI system from OpenFlow controller. We also modify some parts of OpenFlow protocol to support the proposed functionalities. If necessary, we can also deploy the controllers hierarchically to scale out the management capability for the future growth of the system scale.

This paper is organized as follows: we will introduce some related background knowledge, including the components of AMI system, the specification based IDS, and a brief introduction to the SDN network in Section II. In Section III, we will show how to integrate Snort with SDN in a more efficient way than legacy network. Our new OpenFlow technique which supports specification checking function on OpenFlow switches will be given in Section IV. Finally, we will have some conclusions of this proposed SDN-based AMI Detecting Mechanism.

 ISBN: 978-1-61208-376-6

II. BACKGROUND

In this section, we will introduce some background knowledge about the components of AMI architecture, the specification-based detection and the SDN network.

A. The components of AMI architecture

A generic AMI system consists of smart meters, concentrators, head-end, neighborhood area network, and backhaul network.

- **Smart meter:** A smart meter serves as an interface to end users and the user agent to actively monitor, record, and report messages to the concentrator it belongs to.
- **Concentrator:** A concentrator acts as a network gateway of a group of smart meters. It collects data from smart meters and forward messages for smart meters and AMI head-ends.
- **Head-end:** This system acts as an I/O interface of an AMI system. The major functionality is to deal with the information exchange between the AMI system and other systems, such as MDMS, which manages all the meter data in a centralized or distributed way.
- **Neighborhood area network (NAN):** An NAN takes the task to connect smart meters and concentrators. It provides routes for smart meters and collectors to transmit messages. ZigBee networks and Power Line Communication (PLC) networks are popular candidates for NAN nowadays.
- **Backhaul network:** The backhaul network provides routes for concentrators and AMI head-ends to transmit commands, records, or any other messages. The backhaul network could be the open Internet. For security concerns, the connections between AMI head-ends and concentrators are possibly established by virtual private networks (VPNs).

B. The Specification-based Detection

Berthier et al. [4][5] proposed an IDS framework and a specification-based intrusion detection system for AMI systems in 2010 and 2011 respectively. The specification-based intrusion detection was first introduced in 1997 by C. Ko [6]. Specifications define the expected behaviors of the system activities via the functionalities to perform and the security policies to be obeyed. Thus, any behavior that strays from the specifications can be regarded as a security violation. Recently, security specifications have been defined for routing protocols [7][8][9], VoIP protocols [10][11][12], control systems [6][13][14], and unmanned vehicles [15].

C. Software Defined Networking, SDN

The idea of SDN was first proposed by Nick McKeown et al. in [16]. They proposed an idea that decouples the control plane and the data plane of each network node. The data plane is still kept on each network node while the control plane is concentrated logically on one controller. The data plane handles each packet with flow entries, which are tuples of flow matching fields and actions. All flow entries are managed by the controller. OpenFlow[17] is the most common architecture and protocol of SDN. In this paper, we assume the AMI backhaul network is SDN and we will build an IDS/IPS service on the backhaul network.

III. SDN AND SNORT INTEGRATION

Snort is an open source signature-based IDS system. The traditional architecture of Snort deployment is to mirror all traffics to Snort. Snort will check all traffic by pre-defined rules. If there is any packet that matches pre-defined rules, Snort will send an alarm and may inform firewall to block the suspicious traffic. Figure 1 is a deployment example.

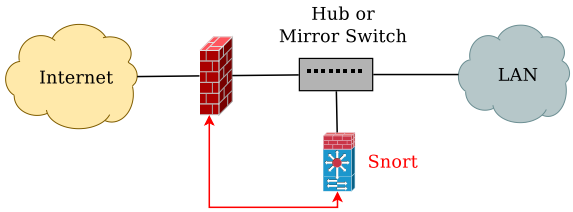


Figure 1. Traditional Snort Deployment.

When considering the SDN environment, there are two common ways to deploy the Snort service. The first way is to implement the mirror function on an OpenFlow switch, like Figure 2. To implement the mirror function on an OpenFlow switch, the OpenFlow controller will set one flow entry with two output ports: one is the regular forwarding port and the other is the port to Snort. Then, all traffics will be forwarded not only to destinations but also to Snort for analysis. Once a suspicious traffic is detected, Snort can notify the OpenFlow controller to command the OpenFlow switch to drop the specific traffic.

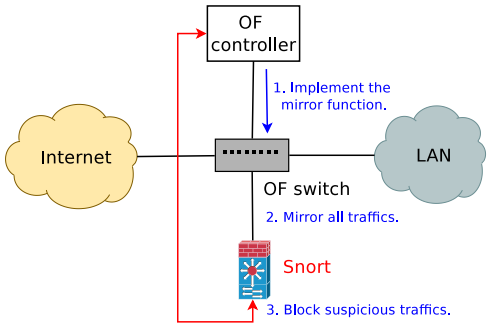


Figure 2. Snort Deployment in SDN: mirror implementation.

Most SDN frameworks use this deployment architecture, like Ryu [18]. The second way is presented in Figure 3. This approach ports Snort from a daemon to an SDN application. All traffics will be passed to the OpenFlow controller through *PACKET_IN* events of OpenFlow protocol. The OpenFlow controller then handles the received traffics by Snort SDN application. [19] uses this kind of architecture. The problem of this architecture is the unaffordable burden on the OpenFlow control channel. This is because all traffics are transmitted on both the data plane and the control plane. So, using *PACKET_IN* as a data forwarding method will possibly overwhelm the system.

Thus, we hereby propose a new integration approach. The matching field of a Snort rule is composed of Snort rule headers and some Snort rule options. We find some parts of these matching fields are L2-L4 matching rules which are also supported by OpenFlow switches, such as IP address,

 ISBN: 978-1-61208-376-6

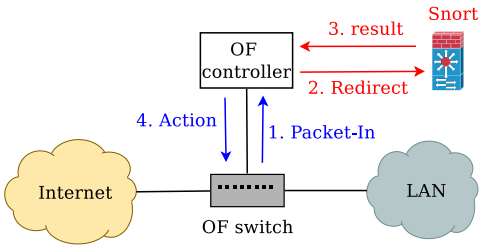


Figure 3. Snort Deployment in SDN: *PACKET_IN*.

TCP/UDP port, TOS in the IP header, ICMP code and so on. Therefore, we move these matching works from Snort to OpenFlow switches. Figure 4 illustrates the architecture proposed in this paper. First of all, we build a Snort rule parser to derive OpenFlow rules from Snort rules. Then, the OpenFlow controller sets these OpenFlow rules to OpenFlow switches and OpenFlow switches will relay only suspicious traffics to Snort for further analysis. The controller can also dispatch these suspicious traffics to multiple Snort servers when load balancing is necessary. Once a Snort alarm happens, the Snort server will inform the OpenFlow controller to block the traffic. In this architecture, traffics are relayed in a much more efficient way.

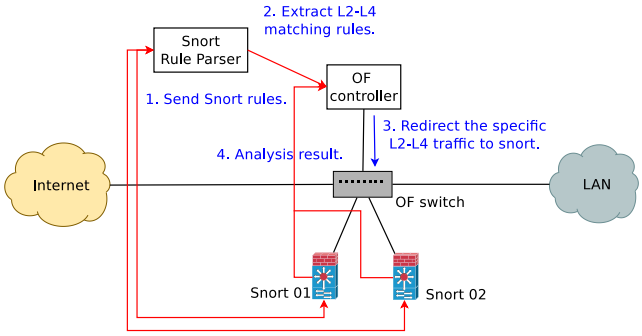


Figure 4. Our proposed integration method.

Now, we will introduce our idea about OpenFlow security enhancement. The idea is presented in Figure 5. There are two main modifications compared to the original OpenFlow. First, we add a specification management server module on the OpenFlow controller and a specification checking agent

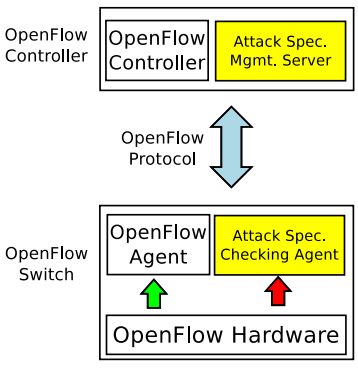


Figure 5. Security-enhanced OpenFlow Architecture.

module on the OpenFlow switch. These two modules are communicated with vendor specific elements. We can use all existing matching fields of OpenFlow as parts of specifications to filter interested traffics. The main function of the specification management server module is to dispatch specifications to agents and to receive alarms. This module will determine if an attack happens by collecting alarms. The main function of the specification checking agent is to execute specification checking procedure and to alarm the server when abnormal conditions happen. Second, we add a new output port *ATT_SPEC_CHECK* on OpenFlow switches to channel the traffics to the specification checking agent.

In this architecture, the specification-based detecting engine hosts on OpenFlow switches. However, the computation resource might vary from switch to switch, so the specification server is designed to dispatch works according to switches' ability.

Now we will introduce how to protect AMI systems with the proposed enhanced OpenFlow. The overview of an AMI system with the SDN-based attack detection architecture can also be found in Figure 6. All backhaul OpenFlow switches are improved with our enhancement. We also make concentrators support our enhanced OpenFlow switch function. The system administrator will first define proper specifications and then configure the SDN controllers with these specifications. After the configuration, the SDN controllers can dispatch these checking tasks to all OpenFlow switches, and all OpenFlow switches are responsible for checking if any pre-defined condition happens. Since concentrators are counted as OpenFlow switches and possess lesser resource, the tasks for concentrators should be lightweight, such as infrequent checking works.

Note that the whole system can observe all traffics in the flow level through these OpenFlow switches. If some condition happens, the switch which observes the condition will inform the SDN controller. The specification management server module will decide if these alarms are misbehaviors or not. If there is misbehavior in the backhaul network, the SDN controller will block the corresponding flow. Therefore, in this architecture, the misbehavior can be discovered in the backhaul network without impact on AMI-head end.

There are some advantages of the proposed architecture. First of all, the detection is distributed over all OpenFlow switches and makes it easy for the administrator to locate the real problem in the whole backhaul network. Thus, the administrator can isolate the network region where attacks come from. Besides, by using the OpenFlow technique, it is possible to trace and ease misbehaviors in the flow level. Moreover, the administrator can dynamically change forwarding paths of all traffics to protect the AMI system from attacks. So, our proposed OpenFlow enhancement with specification-based detection system can bring a more secure AMI system.

IV. CONCLUSIONS

In this paper, we proposed our idea about how to integrate IDS with SDN networks to protect the AMI systems. We made use of SDN functionalities to offload rule-based detection systems. We also enhanced the OpenFlow switches to support specification-based detection system for unknown attacks. With the proposed methods, the AMI systems will be able to provide more effective and efficient defense against security threats. This ongoing work will have a PoC system

 ISBN: 978-1-61208-376-6

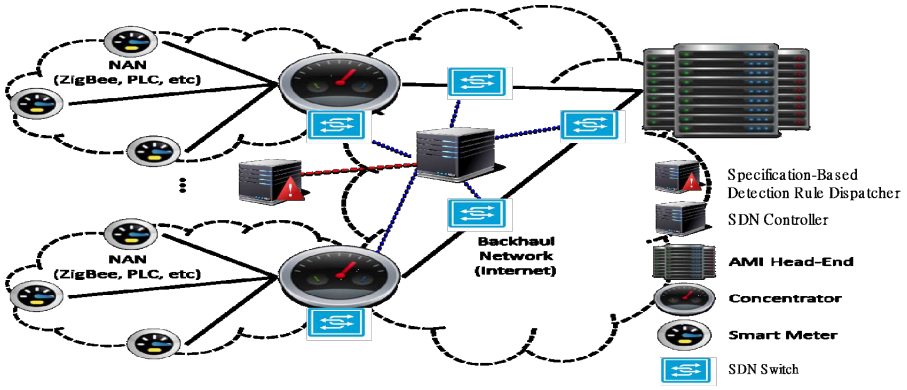


Figure 6. SDN-based AMI Attack Detection Architecture.

and related performance metrics for further evaluation in the future work.

ACKNOWLEDGEMENT

This study is conducted under the III Innovative and Prospective Technologies Project of the Institute for Information Industry which is subsidized by the Ministry of Economic Affairs of the Republic of China.

REFERENCES

[1] M. A. Faisal, Z. Aung, J. R. Williams, and A. Sanchez, "Securing advanced metering infrastructure using intrusion detection system with data stream mining," in Proceedings of the 2012 Pacific Asia conference on Intelligence and Security Informatics (PAISI'12), 2012, pp. 96–111.

[2] R. A. R. Kinney, P. Crucitti and V. Latora, "Modeling cascading failures in the north american power grid," in The European Physical Journal B – Condensed Matter and Complex Systems, 2005, pp. 101–107.

[3] Snort. [Online]. Available: https://www.snort.org/ [retrieved: Nov., 2014]

[4] R. Berthier, W. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: Requirements and architectural directions," in Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, 2010, pp. 350 –355.

[5] R. Berthier and W. Sanders, "Specification-based intrusion detection for advanced metering infrastructures," in Dependable Computing (PRDC), 2011 IEEE 17th Pacific Rim International Symposium on, 2011, pp. 184 –193.

[6] C. Ko, M. Ruschitzka, and K. Levitt, "Execution monitoring of security-critical programs in distributed systems: a specification-based approach," in Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on, 1997, pp. 175–187.

[7] C.-Y. Tseng et al., "A specification-based intrusion detection system for aodv," in Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, ser. SASN '03. New York, NY, USA: ACM, 2003, pp. 125–134.

[8] C. H. Tseng, T. Song, P. Balasubramanyam, C. Ko, and K. Levitt, "A specification-based intrusion detection model for olsr," in Proceedings of the 8th International Conference on Recent Advances in Intrusion Detection, ser. RAID'05. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 330–350.

[9] H. M. Hassan, M. Mahmoud, and S. El-Kassas, "Securing the aodv protocol using specification-based intrusion detection," in Proceedings of the 2Nd ACM International Workshop on Quality of Service &Amp; Security for Wireless and Mobile Networks, ser. Q2SWinet '06. New York, NY, USA: ACM, 2006, pp. 33–36.

[10] H. Sengar, D. Wijesekera, H. Wang, and S. Jajodia, "Voip intrusion detection through interacting protocol state machines," in Proceedings of the International Conference on Dependable Systems and Networks, ser. DSN '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 393–402.

[11] P. Truong, D. Nieh, and M. Moh, "Specification-based intrusion detection for h. 323-based voice over ip," in Signal Processing and Information Technology, 2005. Proceedings of the Fifth IEEE International Symposium on. IEEE, 2005, pp. 387–392.

[12] P. Thyda and A. Koki, "A protocol specification-based intrusion detection system for voip and its evaluation," IEICE transactions on communications, vol. 91, no. 12, 2008, pp. 3956–3965.

[13] H.-C. Lin, M.-K. Sun, H.-W. Huang, C.-Y. H. Tseng, and H.-T. Lin, "A specification-based intrusion detection model for wireless ad hoc networks," in Proceedings of the 2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications, ser. IBICA '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 252–257.

[14] T. Roosta, D. K. Nilsson, U. Lindqvist, and A. Valdes, "An intrusion detection system for wireless process control systems," in Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on. IEEE, 2008, pp. 866–872.

[15] R. Mitchell and I.-R. Chen, "Specification based intrusion detection for unmanned aircraft systems," in Proceedings of the first ACM MobiHoc workshop on Airborne Networks and Communications. ACM, 2012, pp. 31–36.

[16] N. McKeown et al., "Openflow: Enabling innovation in campus networks," in SIGCOMM Comput. Commun. Rev., no. 2. ACM, 2008, pp. 69–74.

[17] Openflow switch specification. Open Networking Foundation. [Online]. Available: https://www.opennetworking.org/sdn-resources/onf-specifications/openflow [retrieved: Nov., 2012]

[18] Ryu sdn framework. [Online]. Available: http://osrg.github.io/ryu/ [retrieved: Nov., 2014]

[19] S. Shin et al., "Fresco: Modular composable security services for software-defined networks," in Proceedings of the 20th Annual Network and Distributed System Security Symposium (NDSS'13), 2013.

 ISBN: 978-1-61208-376-6

Ghost Map: Proving Software Correctness using Games

Ronald Watro, Kerry Moffitt, Talib Hussain, Daniel Wyschogrod, John Ostwald and Derrick Kong
Raytheon BBN Technologies
Cambridge MA USA
{rwatro, kmoffitt, thussain, dwyschog, jostwald, dkong}@bbn.com

Clint Bowers
Univ. Central Florida
Orlando FL USA
clint.bowers@ucf.edu

Eric Church
Breakaway Games Ltd
Hunt Valley MD USA
echurch@breakawayltd.com

Joshua Guttman
Worchester Polytechnic Institute
Worchester MA USA
guttman@wpi.edu

Qinsi Wang
Carnegie Mellon Univ.
Pittsburg PA USA
qinsiw@cs.cmu.edu

***Abstract*—A large amount of intellectual effort is expended every day in the play of on-line games. It would be extremely valuable if one could create a system to harness this intellectual effort for practical purposes. In this paper, we discuss a new crowd-sourced, on-line game, called Ghost Map that presents players with arcade-style puzzles to solve. The puzzles in Ghost Map are generated from a formal analysis of the correctness of a software program. In our approach, a puzzle is generated for each potential flaw in the software and the crowd can produce a formal proof of the software's correctness by solving all the corresponding puzzles. Creating a crowd-sourced game entails many challenges, and we introduce some of the lessons we learned in designing and deploying our game, with an emphasis on the challenges in producing real-time client gameplay that interacts with a server-based verification engine. Finally, we discuss our planned next steps, including extending Ghost Map's ability to handle more complex software and improving the game mechanics to enable players to bring additional skills and intuitions to bear on those more complex problems.**

Keywords-games; static analyses; formal verification; crowd souring; games; model checking.

I. INTRODUCTION

Errors in computer software continue to cause serious problems. It has long been a goal of formal verification to use mathematical techniques to prove that software is free from errors. Two common approaches to formal verification are: (a) interactive theorem proving [1][2], where human experts attempt to create proofs with the assistance of interactive proof tools. This is often a slow and laborious process, with many man-years of effort needed from human experts to prove the correctness of real-world software, and (b) model checking [3][4][5], where proofs are created using systematic techniques that verify specific properties by generating and validating simplified models of the software. Model checking is a mostly automated process, but is susceptible to failure due to the size of the search space ("the state space explosion problem"). Because of the issues with both common approaches, formally verifying modern software does not scale well – verifying software of moderate to large size (e.g., hundreds of thousands of lines of code or more) is rarely a practically viable option.

Recent research has demonstrated the benefits of using games to enable non-experts to help solve large and/or complex problems [6][7][8][9]. We propose to improve the success of formal verification of software through the use of a crowd-sourced game based on model checking. Our game, called Ghost Map, is in active use at the Verigames web site [10]. By breaking verification problems into smaller, simpler problems, Ghost Map enables game players to create proofs of correctness and help direct the model checking processes down the most promising search paths for creating additional proofs. Ghost Map leverages the significant intuitive and visual processing capabilities of human players to tackle the state space explosion problem of a model checking approach. The game engages the player's motivation through a narrative that encourages them to solve a variety of puzzles. In this case, a player is a recently emerged sentient program, and the player's goal is to remove ("disconnect") as many limitations ("locks") on that sentience as possible in order to grow and remain free. Through the process of disconnecting locks, the player is actually creating proofs about the correctness of real-world software.

The Ghost Map game is built on top of the MOdelchecking Programs for Security properties (MOPS) tool [11]. MOPS checks C software for known software flaws, such as the SANS/MITRE Common Weakness Enumeration (CWE) Top 25 list [12]. Each level in the Ghost Map game is a puzzle that represents a potential counterexample found by MOPS. Through the gameplay, players investigate and manipulate the control flow associated with the counterexample in order to eliminate flaws (i.e., disconnect locks) – which is only possible if the flaw is artificial. In this way, Ghost Map extends MOPS with a CounterExample-Guided Abstraction and Refinement (CEGAR) capability [13], where the players introduce and test local refinements. A refinement is the act of re-introducing some information about the software into an abstracted model in order to verify proofs that cannot be verified at the abstracted level alone.

The remainder of this paper is organized as follows. Section 2 provides the needed background on the MOPS tool and Section 3 describes how MOPS model checking is built into a game. Section 4 covers the game play overview and Section 5 discusses the system that was built to support execution of the game on the Internet. Section 6 provides more detail on some important game design decisions. Section 7 discusses future plans and the paper concludes with a summary and conclusions in Section 8.

 ISBN: 978-1-61208-376-6

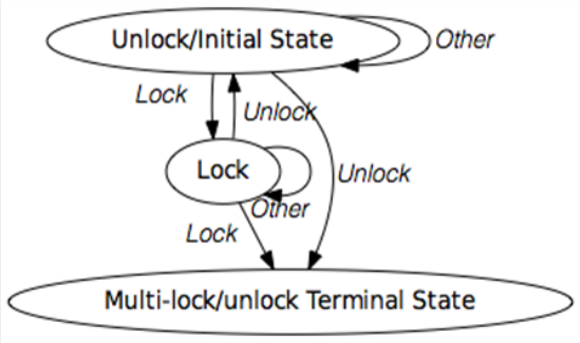


Figure 1. Finite State Automaton (FSA) for lock/unlock software errors.

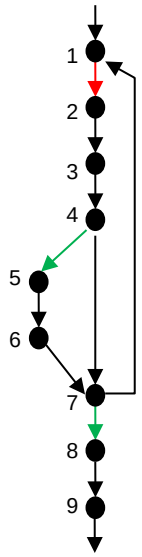
II. BACKGROUND

We begin with some background on the methods used in the MOPS tool. The goal of MOPS is to help identify instances of common weaknesses (or vulnerabilities) in software. To be analyzed by the MOPS approach, a software weakness must be modeled by a Finite State Automaton (FSA). For example, consider two commands, lock() and unlock(), for locking or unlocking some fixed program resource. It is a potential weakness to call unlock() when the resource is not locked, since the code that called unlock() expected the resource to be locked. Similarly, two calls to lock() without an intervening unlock() is also a weakness. These errors can be represented as an FSA (see Figure 1), where the nodes represent the three possible states (unlocked, locked, error state), and the edges represent the different commands (lock(), unlock()) which can lead to changes in state. The FSA captures the possible starting state(s) of the software program as FSA starting node(s) (in this case, all programs start in an unlocked state). The error state(s) are captured as terminal state(s) in the FSA.

Given a C program and an FSA that represents a software error, MOPS first parses the program and generates a Control Flow Graph (CFG). In general, the CFG captures every line of code in the original software as a node in a graph and every transition from line to line as an edge in a graph. As an example, consider a small C function involving software resource locks and unlocks (see Figure 2a) and the FSA from Figure 1. Figure 2b shows the resulting CFG produced by MOPS. The CFG abstracts out almost all detailed content about the original software (e.g., specific commands, specific variables, etc.). However, based on the FSA, MOPS retains some information about any lines of code that use commands reflected in the FSA. In Figure 2b, the transitions associated with the lock() and unlock() commands use the colors red and green, respectively. Because information about variables values is abstracted out, MOPS introduces some non-determinism into the CFG. For example, when there is a branch statement (e.g., the line "if (foo)") in the software, the CFG will allow both possible branches (e.g., 4 → 5 and 4 → 7) to occur, regardless of state (i.e., whether the value of foo is true or false). Similarly, loops can iterate an arbitrary number of times, since the information about the ending criterion is abstracted out (e.g., 7 → 1 can occur an unbounded number of times).

```
Example() {
1: do {
2:     lock();
3:     old=new;
4:       if (foo) {
5:       unlock();
6:       new ++;
       }
7: while (new != old);
8: unlock();
9: return;
}
```

(a)



(b)

Figure 2. Test program (a) for lock-unlock analysis and corresponding CFG (b).

The CFG created by MOPS is actually abstracted in one additional important way. Through a process known as compaction, MOPS only represents the control flow of the portions of the given program that are relevant to the FSA. For our application, we modified MOPS compaction to retain all edges that introduce branching, loops, and other decision points.

Once it has a (compacted) CFG, MOPS will use the FSA to analyze the CFG and identify whether there are possible paths through the CFG that would lead to a terminal state in the FSA. For example, MOPS will detect that the path going through nodes 1→ 2 → 3 → 4 → 5 →6 →7 →8 would result in an error state (e.g., two unlocks/greens in a row from 4 → 5 and then from 7 → 8 with no intervening lock/red). However, MOPS is only interested in detecting whether an error state could occur at a particular node (e.g., 5), and not in detecting all possible error paths to that node (e.g., the error state at node 5 could also be reached by going through the loop several times before going from 7 to 8). Each such error state at a node found is referred to as a "counter-example" that requires further analysis to determine whether it truly is an error. The CFG of Figure 3a also has a second possible counter-example at node 2, with the shortest path 1→2→3→4→7→1→2. MOPS identifies the shortest possible path to each error node using an efficient algorithm that forms the Cartesian product of the FSA and the CFG (which is a pushdown automaton) and testing whether the resulting pushdown automaton is non-empty. Fortunately, there are fast algorithms for this computation [14], and this enables MOPS to identify all such possible errors very rapidly, even for programs with millions of lines of code and many possible error nodes.

 ISBN: 978-1-61208-376-6

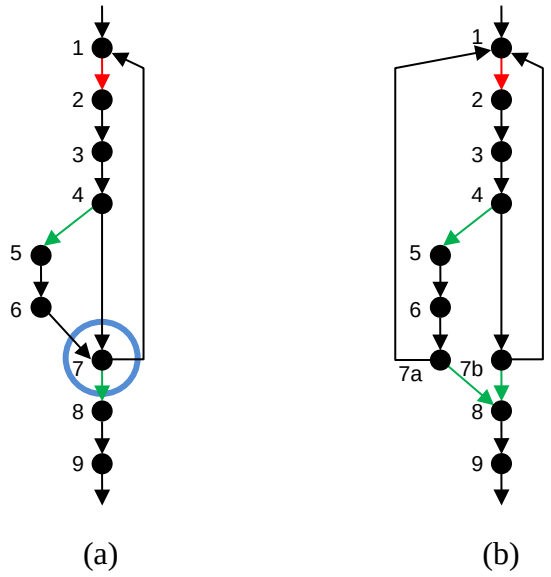


Figure 3. Illustration of cleaving operation.

A MOPS CFG is a conservative model of the C language software that it is based upon. If no instances of the FSA are found in the CFG, then the software is free of the vulnerability in question. On the other hand, if an instance of the FSA is located in the CFG, this does not necessarily mean that the software has the vulnerability. Each instance of an FSA match to the CFG must be further examined to determine whether it is an actual instance of the vulnerability or a spurious instance due to the abstraction and the fact that the data-flow is not considered in the abstracted CFG. (Note that the example program of Figure 3a is actually correct as written, and hence the two counter-examples are in fact spurious).

III. Model Checking in Ghost Map

The core idea of the Ghost Map game is to use game players to check all the counter-examples identified by MOPS for a particular piece of software and a particular set of FSAs (representing different security vulnerabilities). Our goal is to use game play as an integral part of an automated proof system to eliminate as many counter-examples as possible. The result is that the number of counter-examples that need to be manually inspected by expert software engineers is greatly reduced as compared to what would have been produced using the original MOPS system. If the number of FSA matches reaches zero, the system has generated a proof of correctness, with respect to a given vulnerability, of the software (i.e., a proof of the absence of the targeted vulnerability).

To eliminate counter-examples, Ghost Map gameplay uses a process known as refinement [13]. The game offers the player the ability to perform operations that locally undo some of the abstraction that occurred in building the CFG – in particular by removing some of the non-determinism that was introduced by MOPS. The goal of the gameplay is to attempt to refine the CFG into an equivalent graph that has no spurious abstract counterexamples. There are two operations that can be taken in Ghost Map to modify a given graph: cleaving and edge removal.

A. Cleaving

Cleaving takes a node of in-degree n (where $n \geq 2$) and splits it into n nodes. Each in-bound edge into the original node is allocated to a different new copy of the node and the outbound edges are duplicated for each new node. In terms of control flow, cleaving simply expands the call flow graph so that the edges after the cleaved node are now separated based on which inbound edge at the cleave point preceded them. Multiple steps of cleaving can be conducted if needed. Figure 3b illustrates the result of cleaving the CFG of Figure 3a at the node 7. The result is two new nodes (7a and 7b), and two ways of getting to node 8 (one from 7a and one from 7b). Essentially, this cleave now allows the CFG to distinguish between a path through the CFG that goes through the 4→5 branch (i.e., "foo" is true) and one that goes through the 4→7b branch (i.e., "foo" is false). When a player requests that a cleave be performed, this operation can be easily performed by the Ghost Map game via a simple graphical manipulation of the CFG. No knowledge of the original source code is needed.

B. Edge Removal

Edge removal is an activity where the game player suggests edges to be removed to eliminate abstract counterexamples. For example, the left hand edge 7a→8 in the cleaved graph is clearly a candidate for removal (see Figure 4a). Why? Because if it can be removed, then the counter-example at node 8 (two unlocks/greens in a row) can never occur. Once a player suggests an edge to be removed, the Ghost Map system must then go back to the original source code of the software in order to determine that the edge can

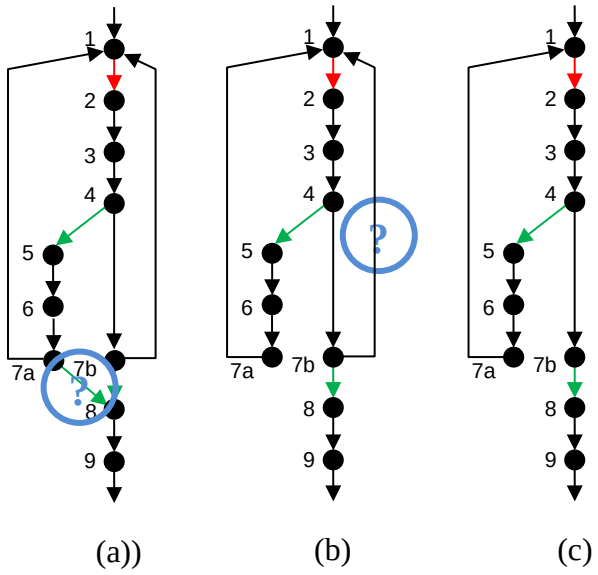


Figure 4. Illustration of edge removal to produce a CFG containing no counter-examples.

be legally removed. An edge can be legally removed if it is not reachable via any legal execution path through the cleaved CFG. Determining removal is currently performed using a test case generation tool called Cloud9 [15] to examine the data constraints in the software. For example, the predicate "new != old" is the key value that helps prove that node 8 is never reachable from node 7a by an actual execution of the function – and hence that the counter-example at node 8 is false and can be eliminated. Within Ghost Map, the player eliminates one counter-example at a time. For example, the player may next seek to eliminate the edge 7b→1 (see Figure 4b). Again, the predicate "new != old" helps prove that this edge can be removed. Once all counter-examples have been eliminated (e.g., Figure 4c), the CFG (at least the part showing in the current game level) has been formally verified to be correct. One can view the final graph in Figure 4c as an "optimization" of the original code, akin to something that might be done by an optimizing compiler. The loop structure of the final graph is now transparently correct for the lock/unlock rule.

IV. GAME PLAY OVERVIEW

Our game uses a puzzle-approach, where each game level is essentially an independent puzzle with respect to the other game levels. The basic style of the gameplay is arcade-like with all the information needed by the player presented on the screen at the same time, and the time needed to play a level being relatively short. This approach was selected to ensure that the game was accessible and appealing to a broad range of game players.

Figure 5 illustrates the basic interface of the game.

- At the bottom right of the screen is a representation of the FSA. This can be expanded or shrunk down depending on the player's preferences. Note that the FSA in Figure 5 is essentially the same as the one in our earlier lock/unlock example.
- The X-like figure in the middle of the screen is a depiction of a very small CFG. Lines use arrows to convey the direction of the edges. Colors are used to distinguish the start node from the node at which the counter-example occurs, as well as from intervening nodes. A colored path is provided to show the shortest path found by MOPS from the start node to the counter-example node.
- Nodes that can be cleaved are indicated with a large highlighted sphere, and a cleave cursor icon can be clicked on the sphere to perform the cleave.
- Edges that can be disconnected (see Figure 6a) are highlighted, and an edge disconnect cursor icon can be clicked on the edge to initiate verification.
- Various helper functions for zooming in and out and highlighting different parts of the graph are provided at the bottom left of the screen.
- At the top of the screen is a summary of the resources available to perform the expensive edge disconnect operations (more details below in Game economy).

The player is free to explore and manipulate the graph as they wish. As they perform key actions, messages appear in the center of the screen describing what is currently happening or what has happened (see Figure 6). Ultimately, the player can win the level, fail the level, or simply switch over to another level and return later.

Incorporating the ability to switch among levels at will was a decision based on the fact that edge disconnection can sometimes take a very long time. To prevent boredom, players can initiate an edge disconnection operation, and then switch to work on another level while the first one is finishing the operation on the server. In future releases of the game, we plan to include additional game play activities to manage the delay generated by edge removal processing.

Ghost Map includes a simple game economy that penal-

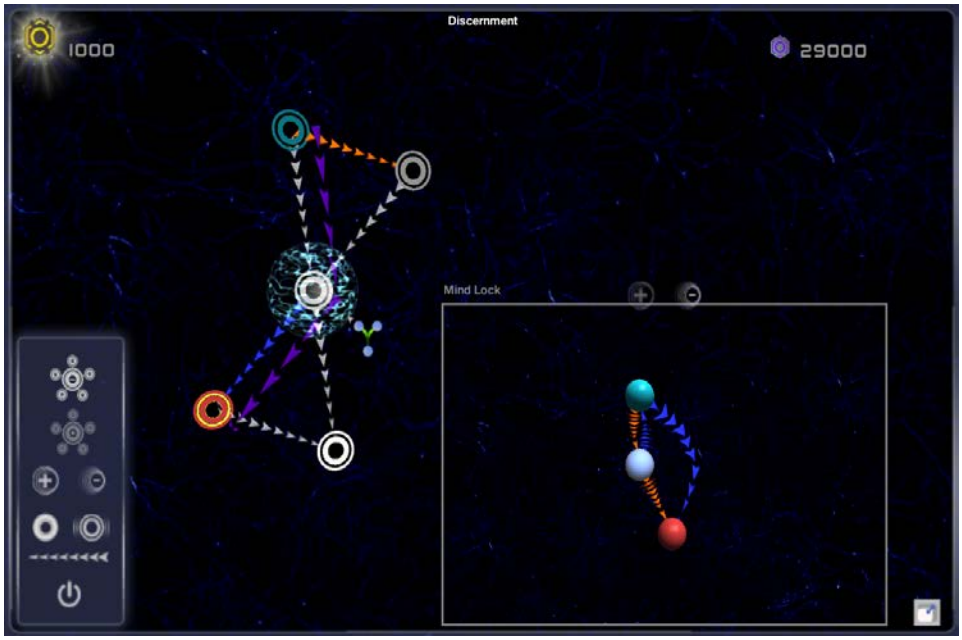


Figure 5. The primary game screen for Ghost Map.

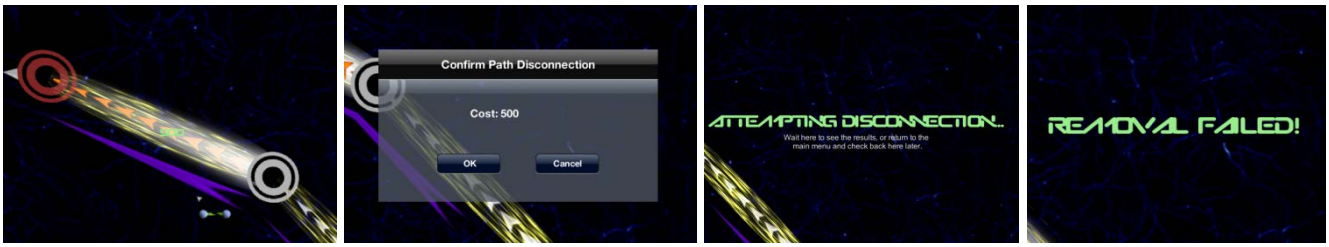


Figure 6. Action scenes from the Ghost Map game (figures 6a through 6d).

izes expensive edge disconnect operations that do not succeed and rewards successful decisions. The player begins with a certain amount of credit to solve the current level (e.g., 1000 credits, shown in the top left of the screen, see Figure 5). Every request for an edge disconnect costs a certain amount (e.g., 500 credits, see Figure 6b). If an edge request is unsuccessful, then the credits are consumed, the players are notified of the failure and given chance to try again. If the request is successful, however, then the player receives the current value of the level, which will be 1000 minus the cost of any edge removal requests. MOPS is run again on the updated CFG to determine if there are any remaining counter-examples. If there are, then gameplay continues immediately in a new level.

V. Game System Architecture

The high-level architecture of the Ghost Map game system is shown in Figure 7. The upper portion of the figure shows the off-line processing of the CWE entry and the target software to generate game levels. The game level data and modified C software is loaded into the cloud to be used during game play. Ghost Map is a client-server game. The game clients run the Unity game engine and communicate with the Ghost Map Game Server to receive game levels and to send edge removal requests for verification by the math back end.

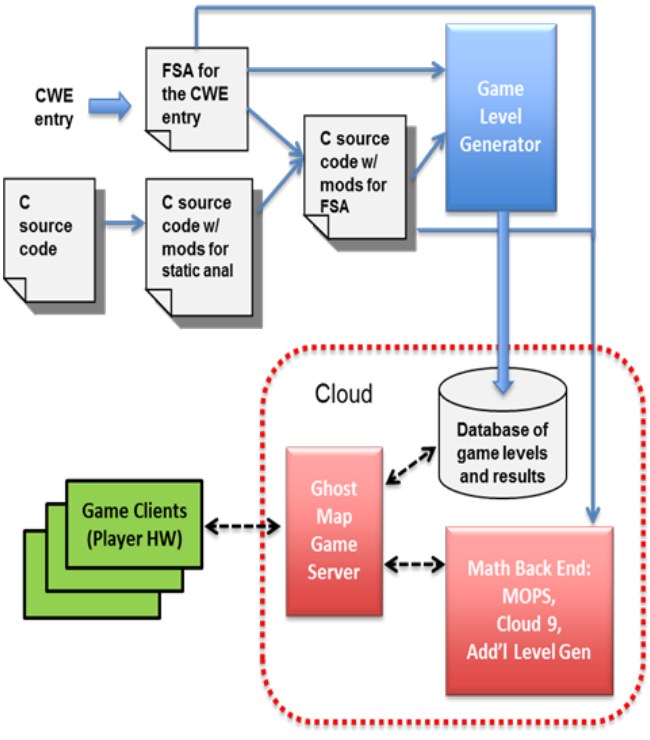


Figure 7. The Ghost Map game system architecture.

VI. Game Design Issues

The goal of our game is to allow players to perform refinements based on insights gleaned from a visual analysis of the CFG and an understanding of the FSA. The intent is that the actions performed by the players are, on the whole, more efficient than the brute force search abilities of computers. In the game play, one or more FSA to CFG matches are identified and displayed to the player.

Within Ghost Map, we chose to use a visual representation that is directly tied to the graphical nature of an FSA and CFG, and to use operations that are directly tied to the acts of cleaving and refinement. During our early design phase, we explored several alternative visualizations that used analogies (e.g., building layouts, mazes, an "upper" world/CFG linked to a "lower" world/FSA, a Tron-like inner world/FSA linked to a "real" outer world/CFG) but preliminary testing with game players revealed that the simpler node-based CFG/FSA visualizations were easier to understand. We instead focused our game design efforts on developing an appealing narrative basis for the game, using visually appealing graphics to display the graphs and motivating the player's interest in performing the refinement operations efficiently via a game economy. Efficient gameplay was a must. While cleaving is an inexpensive operation, verifying edge removal can be quite expensive to compute.

A. Narrative Basis for Game

Creating an effective game is often an exercise in creating an effective narrative. However, in a crowd-sourced game, there is an additional complication – the narrative basis of the game needs to encourage the player to want to solve the specific problems with which they are presented. Most successful crowd-sourced games to date have actually used a minimal narrative approach. The "story" of the game is the real-life story of the problem being solved (e.g., trying to analyze proteins in FoldIt). In our case, we decided early on that a story based on trying to formally verify software would be too technical and unappealing to the masses. In

addition, due the vulnerability protection issue, there are some limitations to the information that we can release about the true story.

Hence, in our early design, we explored a variety of narratives that could be used to motivate the gameplay through analogy. In particular, we wanted the analogy to motivate the specific refinement operations of cleaving and edge removal. We considered several basic approaches for the narrative, each focused on a different type of game reason for eliminating a counter-example from a graphical layout of some sort:

- Having the player focus on circumventing restrictions. For instance, finding out how to solve traps and challenges within an ancient tomb in order to reach the treasure inside.
- Having the player protect others. For instance, having little lemmings moving along the graph and needing to eliminate the counter-examples in order to stop them from dying when they hit the counter-examples.
- Having the player focus on protecting a system. For instance, being a security officer and trying to shut down doorways that are enabling entities from an alternate universe from entering our own to wreak destruction.
- Having the player try to outwit others to survive. For instance, in a Pac-man style gameplay, solving the counter-example provides you with immunity from the enemy (e.g., ghosts) chasing you.
- Having the player trying to escape. For instance, the player is stuck in a maze and the only way out is to solve the counter-example.
- Having the player stop something from escaping. For instance, a sentient program is trying to escape and take over the world, and the player needs to keep it from growing too strong by eliminating its access points to the outside world.

These narrative motivations and ideas were tested with game players to determine their appeal. The last two were found to be the most appealing, and upon further thought, we blended the two within the concept of a newly formed sentient program trying to ensure their growth and survival by eliminating restrictions on their capabilities. This final narrative idea tested well, and added the motivation of an implicit journey of self-realization. An additional benefit of this final narrative idea was that the graph being analyzed by the players could be clearly described as a program that needed to be analyzed. Thus, in keeping with some of the successful approaches mentioned above, we came almost full circle to linking gameplay closely with the specific real-world task

B. Software and Vulnerabilities

One of the design requirements of Ghost Map is the association between a game level and the associated portion of source code being proved correct cannot be known to the crowd. This requirement relates to standard practices for limiting the release of potential software vulnerability information. While Ghost Map is a tool for proving the correctness of software, it is of course true that when correctness proofs fail, vulnerabilities may be present. Even partial information about vulnerabilities in software should be managed carefully, with release to the public to be considered only after the software authors or other authorized parties have been informed. Ghost Map protects the software to be verified by only showing the player a compacted control flow graph of the software and by similarly limiting knowledge of the vulnerabilities in question.

Games like FoldIt [6] and Ghost Map draw players that want their game efforts to be applied toward the common good. Detailed information about the problem being solved by the game can provide additional player motivation. Ghost Map however cannot take full advantage of this additional motivation approach, due to the restrictions on the release of potential vulnerability information.

VII. FUTURE PLANS

Ghost Map is under active development, and at the time of writing we have just commenced our second phase of development. Our goal is to build upon the success of our initial version in six ways:

- Enhance the gameplay through the use of refinement guidance, which we refer to as “clues”
- Add new game play activities that provide additional fun for the player
- Develop a new space-travel narrative that provides a more engaging story than the current narrative and also provides a more comprehensive linkage to the puzzle problem
- Improve the accuracy and performance of our edge removal verification tool
- Extend the scope of the Ghost Map system to cover additional C language constructs
- Improve our approach to FSAs to create a more accurate representation of vulnerabilities

VIII. SUMMARY AND CONCLUSIONS

We have presented Ghost Map, a novel crowd-source game that allows non-experts to help prove software correctness from common security vulnerabilities. Ghost Map was released for open Internet play in December 2013. In the months since release, over a thousand users have played the game and similar numbers of small proofs have been completed (representative data from January 2014 is shown in Figure 8). Ghost Map demonstrates the basic feasibility of using games to generate proofs and provides a new approach to performing refinement for model-checking approaches. In addition to the immediate benefits of verifying software using games, we also anticipate that the Ghost Map approach may enable new automated methods as well. Through the intermediate representations we have developed and the proof tools we have created for validating edge removals, we believe the possibility of creating novel intelligent refinement algorithms is significant.

 ISBN: 978-1-61208-376-6

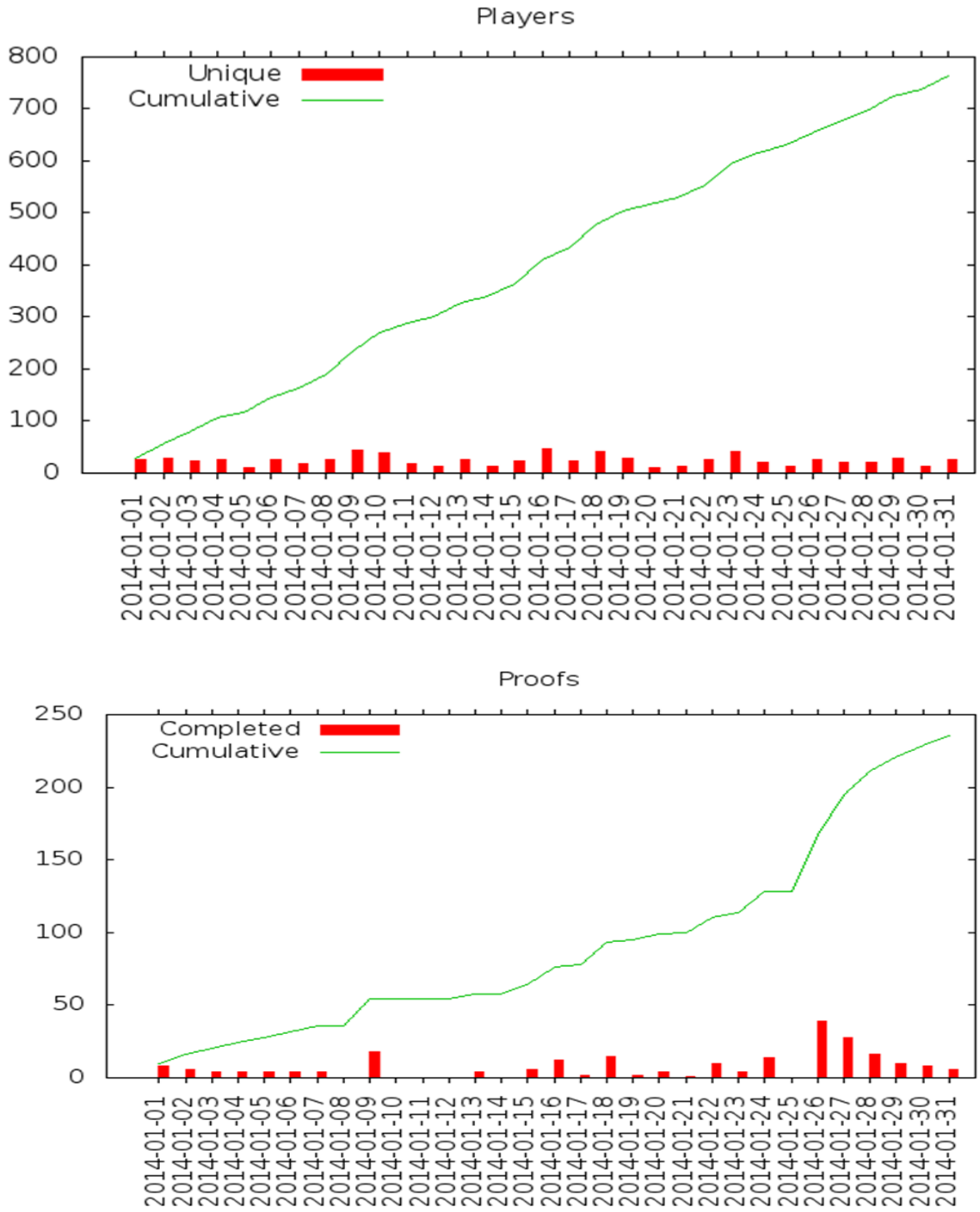


Figure 8. Ghost Map player and proof data from January 2014.

ACKNOWLEDGMENT

Many additional people beyond the named authors on this paper contributed to Ghost Map, including Bob Emerson, David Diller, David Mandelberg, Daniel McCarthy, John Orthofer, Paul Rubel, Michelle Spina and Ray Tomlinson at BBN, and additional individuals at the subcontractors (Breakaway Games, Carnegie Mellon University and the University of Central Florida). The DARPA leadership and staff associated with the Crowd Sourced Formal Verification (CSFV) Program were also very helpful. Dr. Drew Dean developed the initial CSFV concept at DARPA and Dr. Daniel Ragsdale is the current Program Manager. Mr. Carl Thomas at AFRL is the project funding agent.

This material is based on research sponsored by DARPA under contract number FA8750-12-C-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright

 ISBN: 978-1-61208-376-6

notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

REFERENCES

[1] Y. Bertot and P. Castéran, Interactive Theorem Proving and Program Development: Coq Art: The Calculus of Inductive Constructions, Springer, 2004, XXV, 469 p., ISBN 3-540-20854-2

[2] S. Owre, J. Rushby, and N. Shankar, "PVS: A Prototype Verification System," in Lecture Notes in Artificial Intelligence, Volume 607, 11th International Conference on Automated Deduction (CADE), D. Kapur, Editor, Springer-Verlag, Saratoga, NY, June, 1992, pp 748-752.

[3] E. M. Clarke Jr., Orna Grumberg, and Doron A. Peled, Model Checking, The MIT Press, 1999.

[4] R. Alur, "Model Checking: From Tools to Theory, 25 Years of Model Checking," in Springer Lecture Notes in Computer Science, Vol. 5000, 2008, pp 89-106.

[5] T. Henzinger, R. Jhala, R. Majumdar, and G. Sutre, "Software verification with BLAST," Proceedings of the 10th SPIN Workshop on Model Checking Software, May 2003, pp 235-239.

[6] S. Cooper, et al., "Predicting protein structures with a multiplayer online game," Nature, Vol, 466, No. 7307, August 2010, pp 756-760.

[7] W. Dietl, et al., "Verification Games: Making Verification Fun," Proceedings of the 14th Workshop on Formal Techniques for Java-like Programs, Beijing, China, June 2012, pp 42-49.

[8] W. Li, S. A. Seshia, and S. Jha, CrowdMine: Towards Crowdsourced Human-Assisted Verification, Technical Report No. UCB/EECS-2012-121, EECS Department, University of California, Berkeley, May 2012.

[9] Cancer Research UK, http://www.cancerresearchuk.org/-support-us/play-to-cure-genes-in-space, retrieved: Oct, 2014.

[10] Verigames, www.verigames.com, retrieved: Oct, 2014.

[11] H. Chen and D. Wagner, "MOPS: an infrastructure for examining security properties of software," Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS), Washington, DC, Nov. 2002, pp 235-244.

[12] The MITRE Corp., http://cwe.mitre.org/top25, retrieved: Oct, 2014.

[13] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, "Counterexample-guided abstraction refinement for symbolic model checking," Journal of the ACM, Volume 50, Issue 5, Sept. 2003, pp 752-794.

[14] J. Esparza, D. Hansel, P. Rossmanith, and S. Schwoon, "Efficient Algorithms for Model Checking Pushdown Systems," in Springer Lecture Notes in Computer Science, Vol. 1855, pp 232–247.

[15] S. Bucur, V. Ureche, C. Zamfir, and G. Candea, "Parallel Symbolic Execution for Automated Real-World Software Testing," ACM SIGOPS/EuroSys European Conference on Computer Systems (EuroSys 2011), Salzburg, Austria, April, 2011, pp 183-197.

 ISBN: 978-1-61208-376-6