A Survey on Open Automotive Forensics

Robert Altschaffel, Kevin Lamshöft, Stefan Kiltz, Jana Dittmann
Advanced Multimedia and Security Lab
Otto-von-Guericke-University
Magdeburg, Germany
email:Robert.Altschaffel|Stefan.Kiltz|Jana.Dittmann@iti.cs.uni-magdeburg; Kevin.Lamshoeft@st.ovgu.de

***Abstract*—Modern cars are very complex systems incorporating an internal network of connecting an array of actuators and sensors to ECUs (Electronic Control Units), which implement basic functions and advanced driver assistance systems. Opening these networks to outside communication channels (like Car-to-X-communication) new possibilities but also new attack vectors arise, as shown by successful access to internal vehicle data from outside the vehicle. Any attack on the security of a vehicle in principle also constitutes an impact on the safety of road traffic, amongst other threats (e.g., privacy concerns). In this paper, we discuss challenges and propose a means to perform a forensic investigation based on an existing process model from desktop IT forensics and using openly available tools in order to reconstruct an attack or an error, leading to an incident. The main contribution is the identification of requirements for tools used within a forensic process in an automotive environment.**

***Keywords-automotive; computer forensics; embedded systems; forensic processes; safety & security*.**

I. INTRODUCTION

Modern cars rely on a broad range of actuators, sensors and ECUs (electronic control units) to perform basic functions, implementing instrumentation and control circuits. Those ECUs form a decentralized network of resource-limited heterogeneous components. The ECUs are also used in driver assistance systems, some of which are directly involved in vital control functions of vehicles, such as steering (e.g., lane assist), braking and accelerating.

These components form a network inside the vehicle, which is more and more connected with interfaces to the outside (e.g., by using mobile communication technology to update traffic status reports for the navigation system). This increasing interconnection makes attacks on automotive IT easier, as was shown by [1].

Any attack on the security of a component within a vehicle carries a potential implication on the safety of road traffic (both intended and just reckless). Error and faults of individual vehicular components can lead to dangerous situations either through direct means (e.g., brake failure), interruption of an assistance function the driver relies on (e.g. ABS) or distraction (e.g. Multimedia).

When there is an attack or an error, it is necessary to reconstruct the event. This might be necessary to fix the problem, prevent further attacks or to prove guilt or innocence of the involved parties. Especially in the latter case, it is necessary that such a reconstruction follows scientific and well-proven principles. These principles are referred to as a forensic process. A forensic process requires traces used for event reconstruction to be gathered and analyzed in an authentic (originating from the subject of the investigation), with integrity (unaltered by external influences or during the course of the investigation) fashion, as well as the whole process being comprehensively documented. Since in the beginning of an investigation it is very often unclear if an incident arises from an error or an attack, an investigation should follow the same principles without regard to the starting hypothesis of the investigator.

The challenge nowadays is that there is a distinct lack of automotive forensic processes that are openly discussed and peer-reviewed within the scientific community. Nowadays typically isolated solutions are applied, often shrouded in secrecy and heavily protected by intellectual property and copyright mechanisms. The work we present aims at establishing a forensic automotive process for an incident investigation within vehicle IT. This is a supplement to the use of Event Data Recorders (EDRs), which are employed in vehicles to record data relevant to traffic accidents. The rest of this work is structured as follows. Section 2 gives an overview on the technical background of automotive Systems and forensics. Section 3 discusses the forensic process in the context of automotive systems. Currently available tools, which might support the forensic process within an automotive system and their suitability, are discussed in section 4. Section 5 discusses the requirements for tools geared towards supporting automotive forensics while section 6 concludes this paper.

II. TECHNICAL BACKGROUND

This section gives a brief overview on the topic of forensic in classical desktop IT and a basic understanding of automotive IT in order to bring these topics together in the following sections. An overview on the topic of EDR will be given in order to better understand the scope of this paper.

A. Automotive IT

Modern cars consist of components with fixed logic (or none at all) and components with (re-) programmable logic. The latter often include embedded systems and thus are more important for this paper, although being only useful in

 ISBN: 978-1-61208-582-1

conjunction with electronic devices with fixed or no built-in logic. Of particular relevance for our discussion are:

- **Sensors** measure the conditions of the vehicle's systems and environment (e.g., pressure, speed, light levels, rain intensity etc.) as well as user input.
- **Actuators** are electrically operated and manipulate their environment in non-electric aspects (e.g., mechanics, temperature, pressure, etc.).
- **Electronic Control Units** (ECUs) electronically process input signals acquired via sensors and relay commands to actuators. Some units control critical systems, such as the engine or safety-critical systems like ESC (Electronic Stability Control) or SRS (Supplemental Restraint System), while others control comfort functionality (e.g., door control units). ECUs are custom-tailored compact, embedded systems. Due to high cost constraints in the automotive industry, they operate on a minimum set of resources regarding CPU computing power, mass storage and main memory. Common exceptions are ECUs that handle multimedia functionality. The number of ECUs embedded with a vehicle is still rising - while a luxury car in 1985 contained less than 10 ECUs, the numbers increased to more than 100 in 2010 [2].
- **Direct analogue cable connections** connect sensors and actuators directly to a specific ECU.
- **Shared Digital Bus Systems** are used for communication among ECUs [3]. In modern cars, several different technologies for digital automotive field bus systems are used with different capabilities, requirements and cost factors. The most common automotive field bus system, often forming the core network of vehicle systems communication, is the Controller Area Network (CAN) [4]. This CAN network is often divided into sub-networks such as powertrain/engine, diagnostics, comfort or info-tainment. ECUs are connected to the sub-network and these sub-networks interconnect using a CAN Gateway ECU, which handles the routing of messages to different sub-networks. The CAN message consists of several flags, the CAN ID and the payload. The CAN ID represents the type of a message and implies a certain sender and receiver for the message. It is assumed that a message with the corresponding ID is sent by the ECU normally responsible for this message. In addition, the CAN ID serves as priority.

The above implement essential instrumentation and control circuits for the functionality of today's vehicles.

B. Forensics in Desktop IT

The forensic process aims at finding traces that support the reconstruction of an event. In order to increase the validity of the reconstruction, these traces have to be gathered in a way to preserve authenticity (trace origin) and integrity (trace is unaltered). To ensure this, a range of models for the forensic process exist, both for classical crime scenes [5], as well as for computer forensics in Desktop IT [6]. These models are often practitioner driven and usually break down the forensic process into distinct phases. For this paper, we use the forensic process from [7], as it contains both the practitioner's and the computer scientist's view (see [8]), the latter often being omitted in an attempt to provide guidelines for practitioners only. This model includes investigation steps (practitioner's view), data types (computer scientist's view) and methods for data access (computer scientist's view). Thus, by adhering to this model, both the research aspect as well as the implementation of forensic procedures in practice is supported.

For this first survey on automotive IT forensics we rely on the investigation steps:

- **Strategic preparation** (*SP*) represents measures taken by the operator of an IT-system, prior to an incident, which support a forensic investigation.
- **Operational preparation** (*OP*) represents the preparation for a forensic investigation after a suspected incident.
- **Data gathering** (*DG*) represents measures to acquire and secure digital evidence.
- **Data investigation** (*DI*) represents measures to evaluate and extract data for further investigation.
- **Data analysis** (*DA*) represents the detailed analysis and correlation between digital evidence from various sources.
- **Documentation** (*DO*) represents the detailed documentation of the investigation.

The forensic process is furthermore also divided into live forensic and post-mortem forensics. Live forensics covers the part of the forensic examination performed while the system under investigation is active. Post-mortem forensics covers all the part of the forensic examination while the system under investigation is powered-off. Live forensics offer the possibility to find traces in highly volatile areas such as main memory but often comes with the implication of substantially altering the state of the system under investigation - either by letting it perform its current operations or by querying the system for certain information from the main memory, which actively alters the state of the system. Post-mortem forensics allows access to lesser volatile mass storage and analyze it in ways ensuring integrity of the mass storage device (typically by using read-only adapters) but cannot gain insight into the main memory contents. The consideration when to power off a system under investigation and switch from live forensics to post-mortem is to be decided on a case-by-case basis and represents a crucial decision in every forensic examination.

C. Event Data Recorders (EDR)

EDRs describe a range of various devices installed within cars to record data in case of an accident. EDRs are in general use since 1990 [9]. The implementations are generally vendor-specific and are often added functionality of the SRS ECU [10]. Data sets recorded by these devices were only recently standardized [11] and include e.g.:

- The forward and lateral crash force.
- The crash event duration.

 ISBN: 978-1-61208-582-1

- Indicated vehicle speed.

The forensic use of this data is well researched (see [12]). While this data gives insight into accidents, it would not be enough to investigate a malicious attack on automotive IT.

III. REVIEW OF THE FORENSIC PROCESS IN AUTOMOTIVE ENVIRONMENTS

Forensic investigations on automotive IT come with a broad range of challenges originating from the nature of automotive IT. These challenges include:

- The low storage capacity in the ECU means that there is little storage available to store fault codes and event logs. Sometimes fault codes are implemented in a ring buffer where older fault codes are frequently overwritten with newer ones. Time stamps and even a system-wide time base for fault codes are very uncommon.
- CAN Bus communication contains neither explicit senders nor receivers offering no form of sender authenticity. Any message on the CAN Bus can originate from any attached device.
- Access to memory and mass storage is managed by the respective MCUs and is typically inaccessible due to intellectual property and copyright protection measures. In Desktop IT, mass storage generally is easily separated from the system under investigation and attached to a workstation. Here, write-blockers are utilized to prevent all write-operations on the mass storage are possible and hence the integrity is guaranteed. In automotive IT, (parts of) mass storage often is part of the MCU silicone itself, rendering the access a very complex issue.
- Components are seldom standardized. This includes ECUs, mass storage, memory, the message transferred via the bus systems, etc.

These challenges have a great impact on the forensic process on automotive IT. However, with the inclusion of a strategic preparation (SP) step, the selected forensic process model allows to mitigate some of these effects at least as the strategic preparation step allows to prepare a system before an incident occurs (forensic readiness).

The starting point of a case-specific forensic investigation is the operational preparation (OP). In this step, an overview on possible traces is developed. A discussion on what traces shall be gathered and in which order is made. A careful weighting process is initiated, in which the potential gain from the traces is weighted against the structural impact (i.e. side effects on the data contained in the system) of their acquisition. This includes the consideration if live forensics shall be performed at all. To allow for a well-considered decision, in the following we present considerations on live forensics and post-mortem forensics.

A. *Live forensics in Automotive IT*

Live forensics is performed when IT systems inside the vehicle in question are still active and not powered off. During this state, the vehicle contains traces in the *communication* between the various ECUs, their *main memory* and their *mass storage*.

Access to *main memory* and *mass storage* in general is only possible by sending requests to the respective ECUs. This can be done during the normal operation of the car or during some specially initiated diagnosis sessions. In each case, this type of data gathering carries the same implications as in Desktop IT - sending these requests alters the state of the system under investigation (structural impact). Hence, it alters the communication on the bus system transferring the requests to (and the answer from) the ECU, the state of the gateway (usually external tools performing diagnostic requests would be directly attached to the gateway which then routes the requests to the specific bus network) and the specific ECU. While these implications seem grave, it might still be worth when the investigators take these implications into account during the discussion of the conclusiveness of the traces. Hence, the investigator should have an idea of what specific data should be requested in order to keep these implications low.

Communication concerns the data transferred on the various communication channels within the vehicle. These channels include the various CAN bus systems, which form the backbone of vehicular communication. Another technology, used for communication between ECUs, is MOST (Media Oriented Systems Transport, see [13]). From the forensic perspective, both of them have a lot in common. Both of them are broadcast, which means that any device attached to any of these networks can receive all communication on this bus. While it would be possible to set some gateway ECUs into a type of monitoring mode, akin to a monitoring port in Desktop IT routers, this would alter the state of the gateway ECU. It is however, possible to include a data tap in the various networks (as a form of SP) in order to capture communication data if necessary.

B. *Post-mortem Forensics in Automotive IT*

During post-Mortem forensics mass storage data is the main concern. As pointed out before access to mass storage in automotive ECUs is difficult. Mass storage (at least in part is often realized as (re-) programmable non-volatile memory on the MCU silicone. Access is often only possible using debug mechanisms such as JTAG (Joint Test Action Group, see [14]) or Background Debug Mode (BDM, see [15]) and for intellectual property protection purposes this access is often hindered (e.g., by software fuses or removal of pins on the MCU casing). A further challenge is the interpretation of the resulting data (if the acquisition was successful). Due to space limitations, often compact code with little or no documentation or other means of rendering the data intelligible (e.g., ASCII texts), is used. This severely impacts the usage of two old favorites of IT forensics, i.e. the hexadecimal editor and the string search.

On the border between live Forensics and post-Mortem Forensics stands a hardware-in-the-loop test, where a single component is removed from the automotive system, powered on again and then investigated using diagnosis requests. This often alters the state of the ECU under investigation and the

 ISBN: 978-1-61208-582-1

nature of the hardware-in-the-loop test might also have some influence on gathered traces. With the self-diagnose routines implemented in most of the ECUs, a simulation of all the expected outside behavior from sensors, actuators and busses (e.g., with respect to impedance, capacitance etc.) is paramount to maintain the diagnostic trouble codes and status information (see also [16]) for this approach.

Another data source for forensic investigation are external maintenance logs (see [17]) or vehicle logs.

IV. SURVEY OF EXISTING TOOLS AND THEIR APPLICABILITY TO THE AUTOMOTIVE FORENSIC PROCESS

In this section we want to give an overview on how some currently available open tools, which can support forensic investigations into automotive IT hold up on the requirements of forensic investigations. These shall give some context to the considerations made in Section III on the nature of live and post-Mortem Forensics in automotive IT. Some of the tools presented in this section offer functionality used in different steps of the forensic process (for the selection of the particular forensic process see Section II-B). In these cases, only the functionality relevant to the specific step is discussed in the specific subsections.

A. Strategic Preparation (SP)

There are currently no open source tools, which are designed for the use during Strategic Preparation. However, a range of the tools presented for other steps can be used to gather 'known good' states of the vehicle IT in question. This knowledge can also help during the Operational Preparation. A list of the vehicular ECUs, extracted by the tool *UDSim EC* [18], usually used during Data Analysis, can greatly supplement OP - hence producing such a listing before an incident would be a way of SP. In Section V, we present the design process of a tool specifically for the use during SP.

B. Operational Preparation (OP)

For operational preparation, obtaining any documentation on the electronic and electrical system is paramount. Wiring schemes and electronic parts catalogues, as well as repair manuals are a vital source of information before starting any attempt at data acquisition/gathering. While in previous generations of vehicles failing to prepare properly for the acquisition 'only' resulted in a botched investigation destroying vital data, with the upcoming vehicles operating with hazardous high voltage circuits (e-mobility), the safety of the investigators is on the line.

C. Data Gathering (DG)

As mentioned before the in-vehicle communication offers some traces, which might be of interest for a forensic investigation. There are several cross-platform tools that allow the capturing of data on the CAN BUS. Three of them are now described in detail:

- *SavvyCAN* [19] is a graphical tool for capturing and visualizing CAN frames. It provides modules for logging, sniffing and injecting CAN frames as well as interpretation and dissection of signals.
- *Kayak* [20] uses TCP/IP via *SocketCANd* [21] as an additional abstraction layer, providing simultaneous bus access for several users. It comes with a rich set of possibilities to log, sniff and inject CAN frames as well identifying and interpreting CAN signals. It also comes with several options for visualization (e.g., a simulated cockpit) and replay options.
- *Octane CAN Bus Sniffer*[22] is a project of the George Mason University and provides features for sniffing and injection, cyclic keep-alive transmissions for diagnostic sessions and a transmission interface for fuzzing and flooding.

None of these tools does provide any mechanisms to ensure integrity or authenticity of the gathered data and hence external mechanisms needs to be implemented to ensure authenticity and integrity of the gathered data. However the passive reading access does not come with a structural impact.

Another source for possible traces is the gathering of diagnostic data from ECUs. One possibility to gather this data is to use the OBD2 functionality of modern cars. open-source like *Freediag* [23], *OBD2-Scantool* [24] or *O2OO Data Logger* [25] support a wide range of protocols and primarily work with ELM237 based interfaces. These tools allow querying diagnostic trouble codes and diagnosis of ECUs as specified in OBD. There is a structural impact as these tools do transmit messages while establishing, maintaining and performing diagnostic sessions. In addition there are no mechanisms to ensure integrity or authenticity of the gathered data.

D. Data Investigation (DI)

Some of the tools used during the Data Gathering can also help during the Data Investigation by handling prior captured data. This includes, for example:

- *SavvyCAN* can visualize CAN frames. It provides modules for the interpretation and dissection of signals. It supports several formats of CAN signal databases.
- *Kayak* can be used to identify and interpret CAN signals. It also comes with several options for visualization (e.g., a simulated cockpit) and replay options.
- *Octane CAN Bus Sniffer* also offers multiple filtering options and XML signal definitions.

While these tools offer no functionality to ensure integrity and authenticity of the investigation results the integrity of the data under investigation can be ensured by using copies of the original data.

E. Data Analysis (DA)

A number of different tools can be used during the DA. While all these tools can be connected directly to the CAN of

 ISBN: 978-1-61208-582-1

an active automotive this is not advisable from a forensic point of view. Connecting these tools to a virtual CAN device, which replays a trace of CAN communication captured during DG preserves integrity of the trace under investigation. These tools include:

- *CANToolz* also referred to as *YACHT* (*Yet Another Car Hacking Tool*, see [26]) is a framework providing several modules for performing black box analyses of CAN. It can work with multiple interfaces at the same time allowing testing of gateway and firewall functionality. The suite supports UDS and ISO-TP detection and interpretation. Its modular structure allows easy implementation of customizations and extensions. In the current state, it supports integration of different I/O functionalities, such as multiple CAN hardware SocketCAN, TCP tunneling, discovery of ECUs and related services, capture and replay of frames, fuzzing, filtering, sorting, blocking of specific IDs and statistical analysis and interpretation of occurring frames, e.g., for detecting ISO-TP and UDS messages.
- *UDSim ECU Simulator* is a graphical tool for identifying ECUs connected to a bus. It offers three modes: learning, simulation and attack. In learning mode, it identifies ECUs by monitoring their responses to UDS diagnostic queries. Hence it can create a list of available ECUs
- *c0f* (*CAN of Fingers*, see [27]) is a tool for generating fingerprints of CAN busses based on statistical measurements. If fingerprints indicating a healthy system state are known prior to an incident, a following fingerprint might provide an indication of an incident modifying the system state.

As with the tools used during DI these tools offer no functionality to ensure integrity and authenticity of the investigation results. However, the integrity of the data under investigation can be ensured by using copies of the original data.

F. Documentation (DO)

The documentation (according to [7]) can be split into two sections. First, there is the process of accompanying documentation, which maintains an account for all the actions taken by the examiners. This process should ideally be highly assisted by software, recording all parameters and menu selections (see e.g., the script command [28] or the automated documentation in dedicated desktop IT forensic suites such as X-Ways forensics [29]). Within the application context of this article, a mostly manual process involving screenshots, digital photographs, etc. is very likely to be used due to the lack of dedicated forensic software packages as of today.

Using the results from the process accompanying documentation, the final examination report is compiled, which describes the examination process and the results as well the most likely chain of events according to the reconstruction from traces. No dedicated tool support apart from a word processor is typically involved.

V. DESIGN RECOMMENDATIONS FOR FUTURE AUTOMOTIVE FORENSIC TOOLS

As depicted in the prior section, there is a lack of tools geared towards the use in forensic investigation into car IT.

To support the forensic process a tool should:

- the collected/processed data should be useful for the forensic process
- ensure the integrity and authenticity of the collected/processed data
- have a minimized and well-known structural impact
- document the actions performed

An exemplary tool and its design process is described here:

The exemplary tool should be able to support DG by capturing bus traffic. This data is useful for the forensic process as it covers the communication between the various ECUs.

We developed a prototype using open source hardware and software. A Raspberry Pi 3 [30] running Raspbian [31] and PiCAN2 [32] as well as CANtact [33] boards were used to connect to the CAN bus. The Raspberry Pi is controlled via SSH and runs a WiFi Access Point, allowing easy access. We developed a CLI tool, which adapts the concepts of the Linux Forensic Transparent Bridge [34] to automotive CAN networks. In order to create a session for examination, the user has to set name and password which are later used for generating HMACs (Keyed-Hash Message Authentication Code, see [35]). SocketCAN [36] is used for both Cantact and PiCAN boards, allowing passive capturing of network traffic by candump from can-utils, as well as Wireshark/tshark, and neglecting any structural impact by only performing passive read functions. Our tool comes with an automated setup for SocketCAN and allows to set filters for specific IDs. If data is recorded by candump, it can be played back to any other CAN interface (e.g., to a virtual CAN), which then can be monitored by Wireshark as well. This can be useful for further analysis of the network data. We use the default implementation of Python 3 for the HMAC with SHA-512. The concatenation of examiner's name and password is used as key for the HMAC, ensuring integrity and authenticity for the capture.

This setup could also be used as part of Strategic Preparation, as it can be directly installed in to the car (e.g., using a smaller Raspberry Pi Zero) and capturing network traffic for a given period. These captures can be extracted after an incident occurred, providing integer and authentic data.

VI. CONCLUSION

This paper presents the challenges of forensic investigation into potential security incidents in automotive IT. It shows the current state of automotive forensic security and puts the existing isolated solutions into a bigger picture.

 ISBN: 978-1-61208-582-1

A survey on current tools usable for forensic investigations into automotive IT shows the need for dedicated tools geared towards forensics - or at least for the inclusion of means to ensure safety and integrity. As main contribution requirements for such tools are enumerated and the design process of such a tool is presented with the hope to spark the inclusion of forensic functionality in other tools.

ACKNOWLEDGMENT

We like to thank our students working on automotive forensic topics.

REFERENCES

[1] C. Miller and C. Valesek, "Remote Exploitation of an Unaltered Passenger Vehicle," Black Hat USA, 2015.

[2] T. Sugimura, "Junction Blocks Simplify and Decrease Networks When Matched to ECU and Wire Harness". Encyclopedia of Automotive Engineering. 1–7.

[3] A. Hillier, " Hillier's Fundamentals of Automotive Electronics Book 2 Sixth Edition," Oxford University Press, 2014.

[4] Robert Bosch GmbH, "CAN Specification 2.0, 1991" http://www.bosch-semiconductors.de/media/ubk_semiconductors/pdf_1/canliteratur/can2spec.pdf (18/10/2016).

[5] K. Inman and N. Rudin, "Principles and Practises of Criminalistics: The Profession of Forensic Science," CRC Press LLC Boca Raton Florida, USA, ISBN 0-8493-8127-4, 2001.

[6] M. Pollit, "Applying Traditional Forensic Taxonomy to Digial Forensics," IFIP International Federation for Information Processing, Volume 258; Advances in Digital Forensics IV, pp. 17-26, DOI: 10.1007/978-0-387-84927-0_2, 2008.

[7] S. Kiltz, J. Dittmann, and C. Vielhauer, "Supporting Forensic Design - a Course Profile to Teach Forensics," IMF 2015.

[8] S. Peisert, M. Bishop and K. Marzullo, "Computer forensics in forensis", In SIGOPS Operating Systems Review, Volume 42, Issue 3, pp 112-122, ACM, DOI=10.1145/1368506.1368521, 2008.

[9] NHTSA EDR Working Group, "Event Data Recorders", https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/nhtsa_edrtruckbusfinal.pdf, 2002 (retrieved: 9, 2017).

[10] W. Bortles, W. Biever, N. Carter, and C. Smith, "A Compendium of Passenger Vehicle Event Data Recorder Literature and Analysis of Validation Studies," SAE Technical Paper 2016-01-1497, 2016, doi:10.4271/2016-01-1497.

[11] NHTSA, "Federal Motor Vehicle Safety Standards; Event Data Recorders", https://one.nhtsa.gov/staticfiles/rulemaking/pdf/EDR_NPRM_2012-12-07.pdf, 2014 (retrieved: 9, 2017).

[12] N. Singleton, J. Daily, G. Manes, "Automobile Event Data Recorder Forensics," 2008.

[13] MOST Cooperation, "MOST Specification Rev 3.0 E2 07/2010," 2010.

[14] R. Johnson and S. Christie, "JTAG 101 IEEE 1149.x and Software Debug, " https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/jtag-101-ieee-1149x-paper.pdf, 2009, (retrieved: 9, 2017).

[15] Freescale Semiconductor Inc., "CPU 12 Reference Manual," http://www.nxp.com/docs/en/reference-manual/CPU12RM.pdf, 2006, (retrieved: 9, 2017).

[16] W. Rosenbluth and H. A. Adams, "Retrieval and Interpretation of Crash-Related Data from Nonresponsive Electronic Control Units in Land Vehicle Systems," In Journal of Testing and Evaluation, Volume 30, Issue 4, pp. 350-361, ASTM International, ISSN 0090-3973, 2002.

[17] H. Mansor, K. Markantonakis, R. N. Akram, K. Mayes, and I. Gurulian:, "Log your car: The non-invasive vehicle forensics", 2016.

[18] https://github.com/zombieCraig/UDSim (retrieved: 9, 2017).

[19] http://www.savvycan.com/ (retrieved: 9, 2017).

[20] http://kayak.2codeornot2code.org/ (retrieved: 9, 2017).

[21] https://github.com/dschanoeh/socketcand (retrieved: 9, 2017).

[22] http://octane.gmu.edu/ (retrieved: 9, 2017).

[23] http://freediag.sourceforge.net/ (retrieved: 9, 2017).

[24] https://www.scantool.net/ (retrieved: 9, 2017).

[25] https://www.vanheusden.com/O2OO/ (retrieved: 9, 2017).

[26] https://github.com/eik00d/CANToolz (retrieved: 9, 2017).

[27] https://github.com/zombieCraig/c0f (retrieved: 9, 2017).

[28] M. Kerrisk, "script(1) - Linux manual page" [Online] http://man7.org/linux/man-pages/man1/script.1.html (24/05/2017).

[29] X-Ways Software Technology AG, "X-Ways Forensics: Integrated Computer Forensics Software" [Online] http://www.x-ways.net/forensics/(24/05/2017).

[30] https://www.raspberrypi.org/ (retrieved: 9, 2017).

[31] https://raspbian.org/ (retrieved: 9, 2017).

[32] http://skpang.co.uk/catalog/pican2-canbus-board-for-raspberry-pi-2-p-1475.html (retrieved: 9, 2017)

[33] https://linklayer.github.io/cantact/ (retrieved: 9, 2017).

[34] S. Kiltz, M. Hildebrandt, and J. Dittmann, "A transparent bridge for forensic sound network traffic data acquisition", In Sicherheit 2010 - Sicherheit, Schutz und Zuverlässigkeit, 5. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI) Berlin, 5-7 Oktober 2010. S. 93-104. 2010.

[35] H. Krawczyk, M. Bellare and R. Canetti, "RFC 2104, HMAC: Keyed-Hashing for Message Authentication," 1997 .

[36] https://github.com/linux-can/can-utils (retrieved: 9, 2017).

[37] https://docs.python.org/3.5/library/hashlib.html?highlight=sha256 (retrieved: 9, 2017).

 ISBN: 978-1-61208-582-1