Innovative Micro Authentication Server (MAS) Providing Trusted Authorization Services To Mobile Users Equipped with TLS Token

Simon Elrharbi
Ethertrust
Paris, France
Simon Elrharbi@EtherTrust.com

Pascal Urien
Telecom ParisTech
Paris, France
Pascal.Urien@Telecom-ParisTech.fr

***Abstract*— This paper introduces an innovative Micro Authentication Server (MAS), providing trusted authorization services to mobile users equipped with Transport Layer Security (TLS) token. This platform was developed in the context of the PODIUM French Fonds Unique Interministeriel (FUI) research project whose goal is to define an Information Infrastructure (IT) referred as mobile cloud platform, deployed in emergency situation.**

Keywords-Security; TLS; Authentication; RACS; Portal

I. INTRODUCTION

In case of crisis, the rapid deployment of homeland security, military and civil defense means, takes place in a context of a multitude of risks and threats. In particular, the IT infrastructure based on physical and mobile networks is a main target for hackers and attackers. This infrastructure, commonly referred as "mobile cloud", provides ad hoc communication, information storage, and computing resources. It delivers services, such as multimedia instant messaging, fast computing of indoors and outdoors 3D mapping. Its security requires strong network access control, and security policy enforcing data confidentiality, resource authentication, electronic signature, and traceability.

Against this background, an ICT research project, PODIUM [1] has been launched to address these needs. It's funded by the French government through the FUI (Fonds Unique Interministeriel). It involves THALES, ETHERTRUST, ANEO, LUCEOR, NEXEDI, IGO, SDIS 13 and the UPMC-LIP6 from university of Paris 6. This consortium intends to develop a platform and a secure mobile cloud, which is also expected to be local and ad hoc, in order to respond to emergency situations.

This paper focuses on security and control access for the mobile network of the tactic, mobile and local cloud. This infrastructure needs to be autonomous, broadband and accessible only to security services, thanks to ergonomic and strong authentication mechanisms.

The users of the tactical and mobile cloud platform may belong to different sectors and trades (geographical or functional), exchanging information through different formats (SMS, chat, GPS, photos, video streams, etc.), and through different means of communication (mobile phones, fixed PCs or laptops, etc.) according to the Bring Your Own Device (BYOD) principle. Hence, according to these constraints, security and trust concepts are critical issues for mobile applications, as well as corporate applications, since they deal with secret cryptographic keys or PKI resources, which both require secure and trusted storage for user's private keys. In addition, privacy and traceability of users must never be exposed to an un-trusted party, and must be independent of any manufacturers.

The underlying security paradigm of the proposed solution is based on that the digital identity of user or machine is fully represented by X509 certificate, associated to asymmetric keys. The certificate does not contain vulnerable data that might be used by an attacker. Only the owner of the private key (that never travels through the network), identified by the X509 Common Name (CN) attribute, can use the certificate. The IP address is linked to its owner by means of authentication procedure. This mutual authentication is performed thanks to the TLS protocol. Trust is enforced by TLS processing in tamper resistant Secure Elements.

Two main features should be considered for the security and access control architecture. First is the connection of users to the mobile communication network, and second is their access to different applications and crisis data, which are used and exchanged.

The wireless mobile network is based on Access Points (APs) providing wireless local area networks (WLAN). APs are connected to an infrastructure made of WiMesh routers, wired routers, switches or hubs, which provides IP services to specific operating area. Additional features include Captive Portal (CP) and Access Control List (ACL) in order to manage users within the Wi-Fi network.

In this paper, we introduce an innovative micro authentication server (MAS), which provides authentication and trusted authorization services to mobile users, equipped with NFC TLS Token (see Figure 1).

 ISBN: 978-1-61208-736-8

We designed a "Captive Portal" in order to manage the authentication for IP addresses. According to Wikipedia, a "Captive Portal is a web page accessed with a web browser that is displayed to newly connected users of a Wi-Fi network before they are granted broader access to network resources".

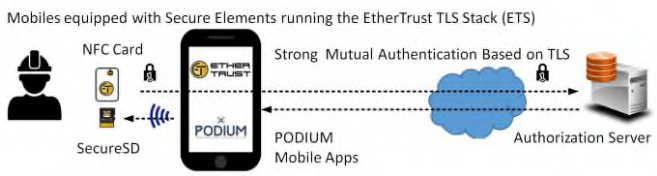

Figure 1. Authentication and Authorization services based on micro authentication server and TLS token.

In our context, the CP IP address is known and shared by mobile cloud users. The users need to sign-up to connect to the mobile network via an Access Point, and are identified by their IP and MAC addresses for getting access.

An important part of the end-user experience is the login process that he must proceed to access to the wireless access point (usually a form requesting login and password). The Access Point enables the scaling of the number of devices supported on the deployed mobile network that may add up quickly. The "login" and "password" requested for access to services will consist of a certificate and a signature, respectively. The approach adopted here is that the allocation of captive portal services must be the result of strong authentication based on the EtherTrust TLS Stack (ETS) technology combining digital certificate and TLS protocol implemented in secure elements [8].

Some mobile cloud services require specific authorizations. Because in a BOYD context the security of smartphone is unknown, the micro authentication server acts as hardware Secure Module (HSM). To reach this goal it runs a RACS server [6] managing a grid of smartcards. According to RACS, user authentication is based on TLS and certificate. A smartcard, typically hosting a private key is remotely used for signing procedures.

The structure of the paper is as follows: first, a description of the different hardware and software components incorporating the captive portal and by extension the micro authentication server using the ETS technology. Then, we focus on the implementation of the trusted authorization services designed to mobile users equipped with TLS Token.

II. HARDWARE ARCHITECTURE

The hardware architecture is described by Figure 2. The micro authentication server (MAS) is built over a Raspberry PI3 board, smartcard readers, PKCS#11 smartcards, Wi-Fi access points, Android smartphone and NFC javacard [9].

The system overall block diagram of the micro authentication server is illustrated by Figure 3. A Raspberry Pi3 board manages an Apache WEB server and a RACS server dealing with PKI smartcards. The captive portal is based on iptables resources. The TLS stack used for mobile authentication purposes is running in a NFC javacard.

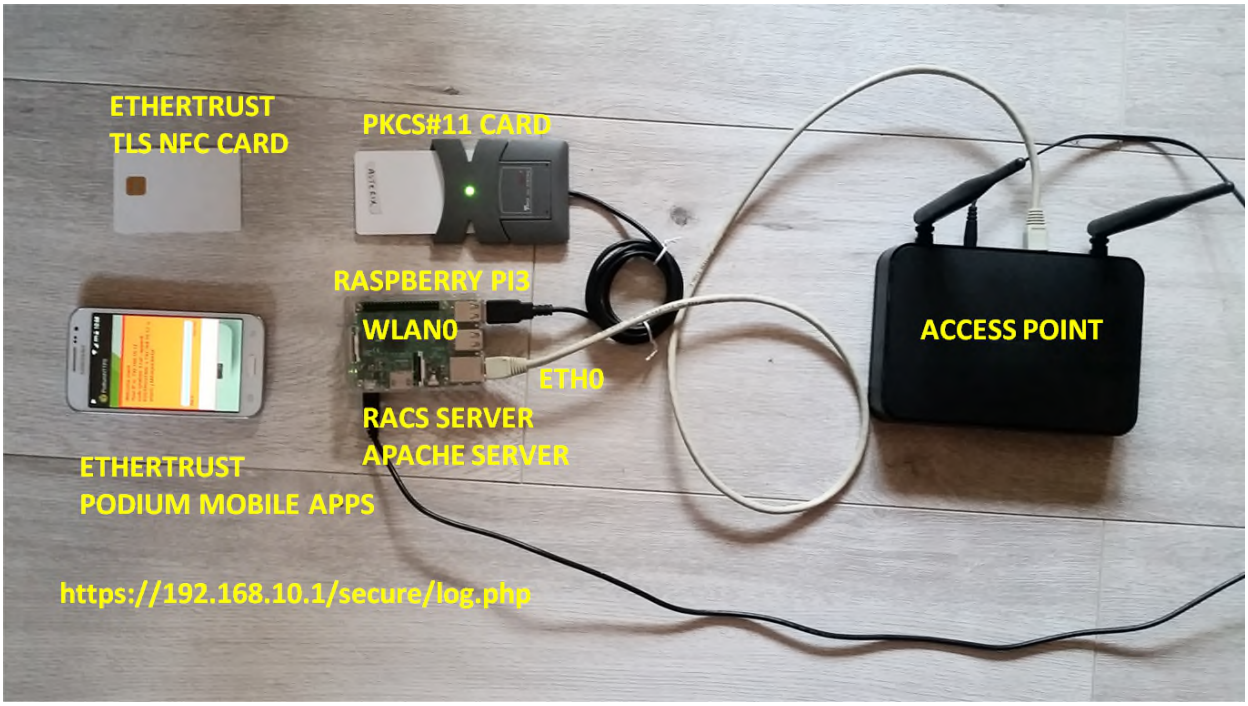

Figure 2. authentication server (MAS) providing trusted authorization services To Mobile Users Equipped with TLS Token through a captive portal

 ISBN: 978-1-61208-736-8

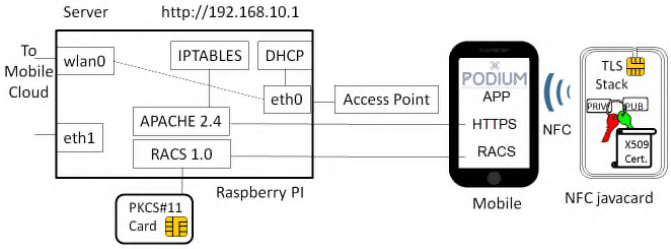

Figure 3. System overall block diagram of the Micro Authentication Server (MAS), providing trusted authorization services to mobile users equipped with TLS token through a captive portal.

A. *Raspberry Pi3*

A low-cost computing environment is supplied by the Raspberry Pi3 (Model B+), which is a multi-purpose advanced reduced-instruction set computing (ARM) processor based credit card sized computer.

The Raspberry Pi 3 Model B+ is a 64-bit quad core processor running at 1.4 GHz. The device connectivity combines dual-band 2.4 GHz and 5 GHz wireless LAN, Bluetooth 4.2/BLE, Ethernet and 4xUSB 2.0 ports. A Secure Digital (SD) memory card slot is also present for loading operating system and data storage.

Raspbian is a provided operating system (OS), but there are various other ARM-Linux OS variants that can run on it. The OS is flashed onto the micro SD. The running device can either be accessed directly using a USB keyboard, mouse and display or via a LAN port by creating a secure shell (SSH) session remotely.

B. *Smartcard Reader*

The smartcard reader is fitted with ISO7816 connectivity requiring the card is manually inserted into the reader by the user. It contains a built-in chip used for electronic processing. The built-in chip acts as a communication controller, passing data to and from the host system and the smartcard and performing functions needed for complete sessions, including: card activation and deactivation, cold/warm reset, ATR response reception, data transfers and configurable timing functions for smartcard activation time, guard time and timeout timers.

Thanks to PCSC-Lite [7] resources, smartcard readers are supported by Raspbian operating system.

C. *PKCS#11 smartcard*

The PKCS#11 standard defines APIs for cryptographic token. The set of objects stored cryptographic smartcards includes certificates and asymmetric keys, such as RSA public private keys.

Smartcard, also known as *Secure Element* (SE), is a confined and secure computing environment in which cryptographic calculations, readings, writes or deletes of sensitive data to be protected in non-volatile memories.

D. *Wi-Fi Access Point*

The Wi-Fi access point is configured as a transparent MAC bridge between IEEE 802.3 and IEEE 802.11 local area networks. It gets an IP address from the micro server (i.e., the Raspberry Pi board), and doesn't provide IP routing services.

E. *Android Smartphone*

An Android Smartphone runs the PODIUM application. It comprises two main elements involved with NFC communications. First, an NFC interface following the ISO 14443 standards. Second, a *Secured Element* (SE) in which host applications (such as PODIUM Applet) and keys are stored. The architecture of mobiles handsets may support several SE of different types: SIM card, embedded SE, or a secure memory card.

F. *NFC smartcard*

Smartcard also known as secure Element (SE) is a tamper resistant microcontroller, equipped with host interfaces such as ISO 7816, ISO 14443, SPI, or USB, whose security is enforced by multiples hardware and logical countermeasures. The security level of both electronics chips and associated operating system are ranked by certifications according to the *Common Criteria* (CC, ISO 15408) standards.

Near Field Communication (NFC) smartcard means that smartcard communicates over a 13.56 MHz air interface, in accordance with ISO/IEC-14443 and NFC Forum standards.

III. SOFTWARE ARCHITECTURE

Software components are divided in two categories: micro authentication server and mobile applications.

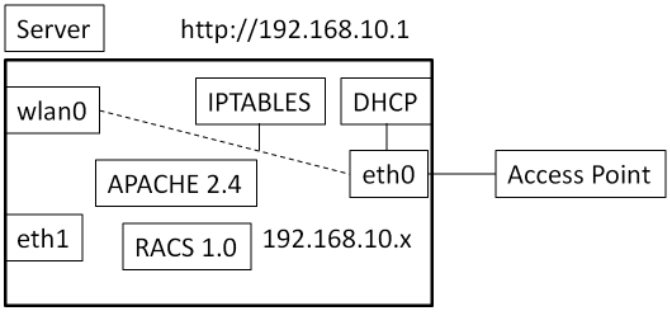

Figure 4. Software architecture of the micro authentication server.

The micro authentication server (see Figure 4) is configured with two network interfaces: Wi-Fi (wlan0) and Ethernet (eth0); an optional additive Ethernet interface (eth1) may be provided by an USB Ethernet token. The Ethernet port delivers DHCP services. The features of iptables [2] are used to enable routing services between network interfaces. An apache server is configured in order to provide user's authentication over TLS, thanks to X509 certificate. Authenticated users can download scripts that control their connectivity to the mobile cloud, i.e., send commands to iptables. They can also access to a RACS server, which requires a certificate in order to use dedicated secure elements.

 ISBN: 978-1-61208-736-8

A. IP tables

The iptables program [2] is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel. It is executed with root privileges.

Three commands (see Figure 5) are needed to manage the server routing context, in order to set a route, delete a route, and dump the route list:

- the append command sets a route for a client (with an IPclient address) between interfaces eth0 and wlan0.
- the delete command deletes a route for a client (with an IPclient address) between interfaces eth0 and wlan0.
- the dump command reads the filter rules

append	sudo iptables -t nat --append POSTROUTING -s IPclient -o wlan0 -j MASQUERADE
delete	sudo iptables -t nat --delete POSTROUTING -s IPclient -o wlan0 -j MASQUERADE
dump	sudo iptables-save

Figure 5. iptables main commands.

B. Raspberry Pi Network Configuration

IP Forwarding	in /etc/sysctl.conf, uncomment the line : #net.ipv4.ip_forward = 1
DHCP for interface eth0	in /etc/network/interfaces, replace the line : iface eth0 inet manual, by the lines auto eth0 iface eth0 inet static address 192.168.10.1 netmask 255.255.255.0
DHCP server	sudo apt-get install isc-dhcp-server
DHCP server for interface eth0	in /etc/default/isc-dhcp-server, add the line: INTERFACES="eth0"
DHCP server configuration	in /etc/dhcp/dhcpd.conf set the parameters: subnet, netmask, range, broadcast-address, routers, domain-name domain-name-servers

Figure 6. Setting the micro authentication server network configuration.

The following operations (see Figure 6) are needed in order to configure the MAS network features:

- enable IP forwarding (i.e. packet routing between network interfaces (i.e. eth0 and wlan0) ;
- enable DHCP service on the Ethernet interface (eth0);
- install a DHCP server;
- associate the DHCP server to the Ethernet interface;
- set the DHCP server working parameters.

C. Apache 2.4 server

At the time of writing, the Apache2.4 version was not released for Debian systems. We manually installed it, i.e. we performed source downloading and local compilation. This version is needed in order to disable the TLS ticket option, which is not supported by the current TLS smartcard client.

The WEB server is configured with a certification authority (CA), an X509 certificate and its associated private key. Files located in a protected directory (/secure) can be downloaded only by HTTPS requests, implying client strong authentication, based on its X509 certificate.

The easiest way to design a WEB user interface is to encapsulate iptable commands in bash script (see Figure 9). This approach requests to authorize scripts with root privileges. The user's root privileges are controlled by the etc/sudoers file. The line detailed by Figure 7, enables the root privilege for a bash script (script.sh).

As illustrated by Figure 8, a remote user is authenticated by its certificate, which embeds its identity (the CN attribute).

```
daemon ALL=(ALL) NOPASSWD: /bin/bash
/usr/local/apache2/sbin/script.sh *
```

Figure 7. The sudoers file, needed for script execution with root privilege.

```
<?php
if ($_SERVER['SSL_CLIENT_VERIFY'] != "SUCCESS")
{ Header("HTTP/1.0 401 Unauthorized");  exit; }
echo " Welcome " ;
echo $_SERVER['SSL_CLIENT_S_DN_CN'] ;
echo " IP="; echo $_SERVER['REMOTE_ADDR'] ; echo " ! " ;
$cmd = 'sudo /bin/bash /usr/local/apache2/sbin/script.sh ' .
$_SERVER['REMOTE_ADDR'];
$result = exec($cmd);
 ?>
```

Figure 8. Exemple of php page (on.php) performing user authentication, and establishing a route with iptables.

```
#!/bin/bash
a="sudo iptables -t nat --delete POSTROUTING -s "
b=" -o wlan0 -j MASQUERADE"
c="$a$1$b"
$c
a="sudo iptables -t nat --append POSTROUTING -s "
b=" -o wlan0 -j MASQUERADE"
c="$a$1$b"
echo $c
$c
```

Figure 9. A script that establihes a route for a given IP address betwen eth0 and wlan0

 ISBN: 978-1-61208-736-8

D. RACS 1.0 server for Raspberry Pi

The RACS (Remote APDU Call Secure) protocol is described by an IETF draft [3].

RACS servers manage grid of secure elements (usually smartcards). Remote users are authenticated thanks to the TLS protocol, dealing with strong mutual authentication relying on client and server certificate and associated private keys.

An open implementation is available for Raspberry Pi systems [4].

More details on RACS are available in [6].

E. Ethertrust TLS stack (ETS Applet)

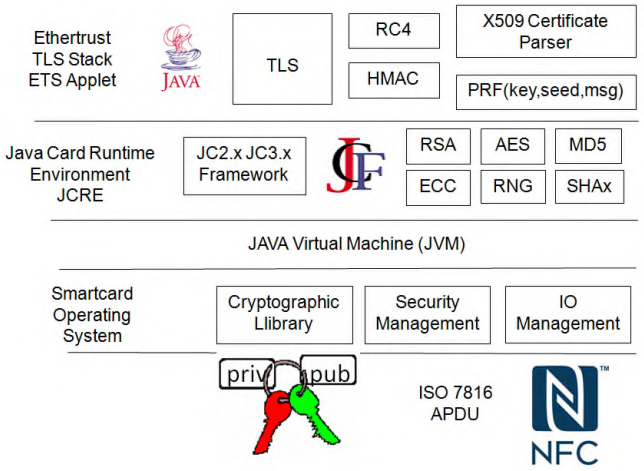

Figure 10. The ETS Applet, in a javacard software environment.

The ETS TLS (see Figure 10) stack is an applet written for javacard [9] that fully processes TLS sessions [8]. The application size is about 20KB for the client only applet. The time required to open a TLS session is about 10 to 2 seconds, depending on the device manufacturer. The four TLS flights of a full session are fully processed by the smartcard. The ETS applet always uses an X509 certificate for client authentication. Upon success the ephemeral session keys, used for record messages encryption and integrity checking, are transferred to the mobile. So, each time the NFC card is tapped against the mobile, a new TLS session is started and afterwards transferred to the mobile.

F. PKCS#11 Java Card Applet

The MUSCLE (Movement for the Use of Smart Card in a Linux environment) project developed an open javacard application (the MUSCLE Applet [5]), which provides PKCS#11 services.

This application generates asymmetric key pairs. It is protected by a PIN code, and computes PKCS#1 signature.

G. Mobile Podium Application.

The mobile application comprises a HTTPS and a RACS client. The TLS layer is provided by the TLS smartcard. The application is started by tapping the NFC card against the mobile.

As depicted by Figure 11 the server address, the RACS TCP port, and the script (on.php) needed for logging purposes are configured by the user.

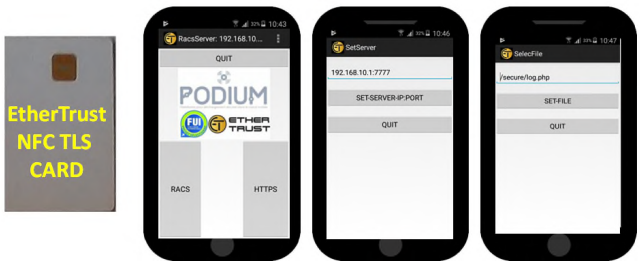

Figure 11. PODIUM Mobile Application.

IV. AUTHORIZATION SERVICES

In this section we detail the portal WEB interface, the authentication procedure, and the signature service.

A. WEB interface

As illustrated by Figure 12 the portal is controlled by three main scripts, on.php for connection to the mobile cloud services, off.php for disconnection, and list.php that collects iptables information.

Scripts located in the /secure directory require a valid certificate for the TLS client, whose access rights are verified

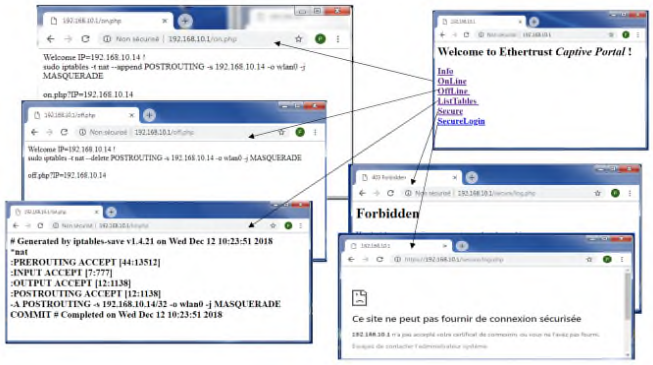

Figure 12. Illustration of the WEB interface with the portal, which comprises hree main scrips on.php, off.php, and list.phpPortal Service

B. Portal Service

The user taps his TLS NFC card against the mobile, and selects the HTTPS service, performing the https://server.com/on.php URL A TLS session is started between the smartcard and the server. The chip is identified by the CN attribute stored in the X509 certificate (client in Figure 13). The script on.php enables the access to the mobile cloud, thanks to iptables resources. An HTTP response is returned to the mobile, whose content is displayed by the mobile (see Figure 13).

 ISBN: 978-1-61208-736-8

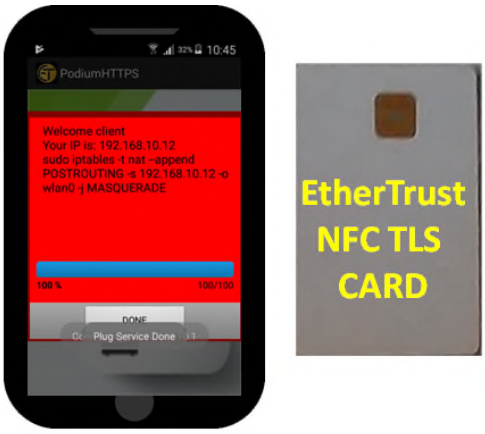

Figure 13. The user screen after a sucessfull authentication with the portal

The server returns the CN attribute found in the X509 certificate ("Client"), and the IP address. The iptables command, started from then php script, is also echoed for debugging purpose.

C. *Signature Service*

Figure 14. The Electronic Signature Service

The user taps his TLS NFC card against the mobile, and selects the RACS service (see Figure 11). A TLS session is opened between the smartcard and the RACS server. The chip is identified by the CN attribute stored in the X509 certificate. Upon success the user selects a smartcard, belonging to the grid hosted by the RACS server, to which it is authorized to access according to its rights. Thereafter, it sends a value to sign to a PKCS#11 chip. The returned signature is displayed by the mobile application (see Figure 14).

V. CONCLUSION

In this paper, we introduce an innovative micro authentication server, dealing with TLS NFC token. We believe that this technology could be applied in IT platforms in which interactions are performed from mobiles, in a BOYD context, whose security level is unknown.

The main advantages of our approach are the following:

- Mutual strong authentication based on standardized technology, i.e. TLS.
- On the client side all operations are computed in a tamper resistant environment.
- No credentials or authentication procedures are handled by the smartphone.

We are currently working on a new generation of TLS token, compatible with TLS 1.3.

REFERENCES

[1] The PODIUM FUI20 project, http://m.competitivite.gouv.fr/toutes-les-actualites/actualite-23/les-resultats-du-20eme-appel-a-projets-du-fui-regions-58-nouveaux-projets-873.html, july 2015, [retrieved: May, 2019]

[2] IPTABLES Manual, http://ipset.netfilter.org/iptables.man.html, , [retrieved: May, 2019]

[3] IETF DRAFT, Remote APDU Call Secure (RACS), https://datatracker.ietf.org/doc/draft-urien-core-racs

[4] RACS_0_1, https://github.com/purien/racs_0_1/tree/master/raspberrypi3_bin/racs_0_1, [retrieved: May, 2019]

[5] Muscle Applet, https://github.com/OpenSC/OpenSC/wiki/Muscle-applet, [retrieved: May, 2019]

[6] P. Urien, RACS: Remote APDU call secure creating trust for the internet", 2015 International Conference on Collaboration Technologies and Systems, CTS 2015, Atlanta, GA, USA, June 2015

[7] PCSC Lite, https://github.com/LudovicRousseau/PCSC, [retrieved: May, 2019]

[8] P. Urien and M. Betirac, A Triple Interfaces Secure Token -TIST- for Identity and Access Control in the Internet Of Things, the Second International Conference on Smart Systems, Devices and Technologies, SMART 2013, 2013 - Roma, Italy

[9] Z. Chen, Java Card™ Technology for Smart Cards, O'Reilly, 2000

 ISBN: 978-1-61208-736-8