UDC 004.9

Mobile driving license system deployment model with security enhancement

V. Blynkov[2,3, a], V. Yaremenko[1,3]

[1] *National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»*

[2] *National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Institute of Physics and Technology*

[3] *Samsung R&D Institute Ukraine (SRK)*

Abstract

The existing potential of mobile devices attract government to use mobile devices at the government level. Therefore, the ISO 18013-5 [1] standard for identity identification was developed. It provides a mechanism for obtaining and trusting driver's license data. This paper describes an approach to deploy mobile driver's license system and an approach to enhance security for data storage based on analysis of existing methods and models.

Keywords: mobile driving license, embedded Secure Element, trusted execution environment, NFC

1. Introduction

Today, virtualization of everything becomes more relevant, more and more ordinary processes are moving to the digital world. Taking into account, that majority of people have smartphones it allows to deploy mobile driver license system on state level, which can be a benefit to both of sides users and government [2]. Therefore, it is suggested to transfer physical documents to the digital world. In this case, it is purposed to use the smartphone as a carrier of sensitive user information (in particular driver's license). On the one hand, it facilitates the life of the user, there is no need to carry documents that can be lost, on the other hand it makes process of identity verification much easier and far reliable, since this system will guarantee the authenticity of the data and will not cause distrust on either side.

Currently exists a technology, but appears a problem of the deployment of such system into the real environment based on the government and device's manufacturers limitations.

2. Background

In this section, presented a brief overview of technologies which will figure in this work.

2.1. Trusted Execution Environment

The TEE concept is based on a virtualization abstraction of device hardware and resources to host and execute security-sensitive applications and store sensitive data on a mobile platform. In its turn Rich Execution Environment (for example Android OS) can not provide a required security level for processing sensitive data and it is vulnerable to a large number of attacks. TEE provides a mobile phone processor (CPU) extension that enables it to operate normally or in secure mode or in high security mode. Typically, it separates to two modes: normal world (NWd) and secure world (SWd). TEE provides only secondary protection compared to embedded Secure Element (eSE) [3]. TEE's goal is to isolate sensitive resources hardware and software (for example CPU, memory, GPU, storage, data) so that only authorized applications, known as trusted applications (TA), are accessed and executed in a highly secure context. The TEE security kernel is implemented by a secure kernel that manages the separation of critical processes, security domains, and secure access to device resources and any associated drivers [3].

2.2. Trusted application

TA is executed in the context of TEE and protected by software and cryptographic isolation. Such an application is usually a small binary code that implements the TEE APIs. They are cryptographically signed, securely loaded and responsible for security-sensitive transactions. TA is also known as the Trustlet. Any TA communicates with other applications such as RA through the TEE Client API, which provides a connector interface between the secure kernel and Rich OS. TA also gained access to secure storage in TEE. TA facilitates the creation of a secure channel with other trusted entities, such as eSE, for the purpose of sharing sensitive security data. It plays a key role in the reliable collection of user credentials and the execution of security sensitive operations [3].

2.3. Normal World

Normal world - runs the standard software stack that the user expects to see: Linux, Android or the like [4].

[a] v.blynkov@gmail.com

The ARM TrustZone concept is based on the division of the runtime environment into secure and unsecured ones. Moreover, Normal World (NWd) does not have access to Secure World (SWd), whereas the latter can be accessed anywhere. This approach affects not only the processor, but also memory, bus transactions, interrupts, peripherals within System-on-a-Chip (SoC) and, including, software.

2.4. Secure World

Secure world - runs a separate and typically unseen operating system that provides security services to applications running in the NWd and to the NWd operating system itself. SWd stack is very compact and contains only minimum code necessary to service the security functions of the NWd software [4].

2.5. Java Card

JavaCard is a type of SmartCard. A smart card is a portable and tamper-resistant computer. Unlike magnetic stripe cards, smart cards carry both processing power and information. Therefore, they do not require access to remote databases at the time of a transaction [5].

Java Card technology offers a way to overcome obstacles hindering smart card acceptance. It allows smart cards and other memory-constrained devices to run applications (called applets) written in the Java programming language. Essentially, Java Card technology defines a secure, portable, and multiapplication smart card platform that incorporates many main advantages of the Java language [5].

2.6. Java Card Applets

Java Card applets should not be confused with Java applets because of identical name - applets. A Java Card applet is a Java program that adheres to a set of conventions that allow it to run within the Java Card runtime environment. A Java Card applet is not intended to run within a browser environment. The reason the name applet was chosen for Java Card applications is that Java Card applets can be loaded into the Java Card runtime environment after the card has been manufactured. That is, unlike applications in many embedded systems, applets do not need to be burned into the ROM during manufacture. Rather, they can be dynamically downloaded onto the card at a later time [5].

2.7. Secure Element

A Secure Element is a tamper-resistant platform (typically a one chip secure microcontroller) capable of securely hosting applications and their confidential and cryptographic data (for example cryptographic keys) in accordance with the rules and security requirements set by well-identified trusted authorities. There are different form factors of SE: embedded and integrated SEs, SIM/UICC, smart microSD as well as smart cards. SEs exist in different form factors to address the requirements of different business implementations and market needs [6].

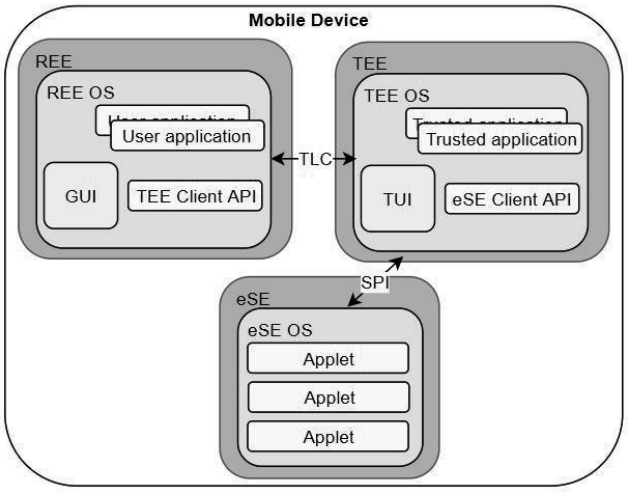

Fig. 1. Mobile device architecture

3. Related Work

There are several possible options for implementing mobile driver's license system (mDL), all of which have common features since they are based on ISO 18013-5 [1] specification, the main differences in these implementations is how to store sensitive data on a user's device and how to authenticate this data [7]. Precisely on the analysis of variants of sensitive data storage will be purposed a new security enhanced approach to store sensitive data on user device and secure representation on reader device as well. Such variants of data storage and processing:

1) Rich Execution Environment
2) Rich Execution Environment + embedded Secure Element
3) Rich Execution Environment + Trusted Execution Environment
4) Rich Execution Environment + Trusted Execution Environment + embedded Secure Element

All the variants of the proposed system are intended to be used on the Android platform devices as it is the most common operating system for mobile devices.The mobile driver's license system must include the reader side - the application for reading and verifying data; the holder side - the application containing sensitive user data (driver's license information) [7].

3.1. Rich Execution Environment

User credentials are stored directly in Rich Execution Environment (REE), that means within the operating system of the mobile device itself. For example, in the package of the application itself, in another application, or as a file in the device storage. In such case, all

cryptographic operations are performed in the REE, which is not completely secure. Compromising device OS can lead to leakage of the user sensitive data or could be vulnerable to replication attacks if device has root access. On the other hand, verification on the reader is also executing in the REE which can also lead to data leakage. The advantages of this approach are that the system can be implemented on any mobile device with the Android operating system [8].

3.2. Rich Execution Environment + embedded Secure Element

User credentials are stored in the embedded Secure Element (eSE) and all cryptographic calculations are also performed in the eSE. The eSE is a separate processor with its own OS which is isolated from the REE. The eSE that installed in mobile devices are EAL4+ [9] or higher certified [10], which permits to gain maximum assurance from security engineering. This allows to resist most attacks, including physical attacks. However, verification of user data is executed in REE, which can cause confidential data leak [8].

This approach is more secure but imposes restrictions on the deployment. Therefore, such a system can only be deployed on devices that have an eSE. Also there is an issue with provisioning of user's data, because access to the eSE only possible with special keys from chip vendor.

3.3. Rich Execution Environment + Trusted Execution Environment

User credentials are stored in Trusted application in TEE and all cryptographic calculations are also performed in TEE. TEE is a special area of the main processor that allows to isolate command execution from REE, it also guarantees confidentiality and integrity. TEE offers higher data security than REE and more functionality than eSE. Thus, the use of TEE is a fairly safe use but does not protect against physical attacks [8].

This approach also imposes its limitations, because such a system can only be deployed on devices that support TEE, but at the moment, almost all modern processors support TEE. The same issue are exist for using TEE as for using eSE, trusted application must be signed by vendor.

3.4. Rich Execution Environment + Trusted Execution Environment + embedded Secure Element

User credentials are stored in the eSE, but all cryptographic calculations are performed in the TEE. This approach the most secure of all represented [8].

4. Statement of the problem

This paper discusses the advantages and disadvantages of using different approaches to store sensitive user data on mobile devices, verifying it and deployment mobile driver's license system as a whole. Based on analysis of this approach a new security enhanced system will be purposed. As it is about system that intended to use confidential data on state level, the security of the system is more important than simplicity of it realization [7].

The object of study is the personal identification via mobile devices.

The subject of the study is the approaches for secure store and verify sensitive data on mobile devices.

5. Design

The main objective is to design a secure system than can be certified and used on a state level [2], in order to be trusted for both of sides: user side and government side. To provide the highest level of security approach that combining all security measures must be used with purposed improvements for security enhancement. Such system could be complicated in realization but deployment and provisioning for the end-user shall be simple and straightforward.

5.1. Approach details

Based on all the advantages and disadvantages of different approaches, approach that uses Rich Execution Environment, Trusted Execution Environment and embedded Secure Element was chosen this approach best meets the security conditions. Purposed improvements to this approach:

1) REE will only be used to send commands to the eSE and TEE and for transmitting encrypted data, no direct access to sensitive user data would be provided for REE.

2) eSE will be used to store user credentials and for all cryptographic calculations, such as encryption,decryption, verifying user data.

3) TEE will be used to display sensitive user data using Trusted User Interface (TUI) [11]. TUI allows a Trusted Application to interact directly with the user via a common display and touch screen, completely isolated from the REE [11], and no way to copy, or still data from it.

5.2. mDL lifecycle

Lifecycle of the proposed approach on mobile device:

1) Execution of program starts with android application in Rich Execution Environment, it is connected with Trusted Execution Environment via TEE Communicator or TA Connector (TLC). TLC facilitates communication between the application in Rich OS (Android), and the corresponding Trusted Application residing within TEE [3].
Since application in the REE can't get access to sensitive data stored in the eSE it shall send request to TEE corresponding Trusted Application in order

to get ability to retrieve necessary data from the eSE.

2) Corresponding Trusted Application in TEE receives request from Rich OS application, provides additional security measures, forms request to eSE and sends it via Serial Peripheral Interface (SPI). SPI is a synchronous serial communication interface in full duplex mode, designed to provide simple and inexpensive high-speed pairing of micro-controllers and peripherals primarily in embedded systems [12].

3) In eSE special JavaCard application called Applet, retrieves request from TEE and processes it. In case if data shall be represented to the user interface, sensitive data passed to the TEE, where it is represented to screen via Trusted User Interface. In case if sensitive data should be transferred, Applet performs cryptographic calculations, forms special secure structure ready for transferring and pass it back to the TEE.

4) In case of data transfer TEE acts as intermediary between eSE and REE which provides high level of security and guarantees that sensitive data stored in the eSE will remain untouched.

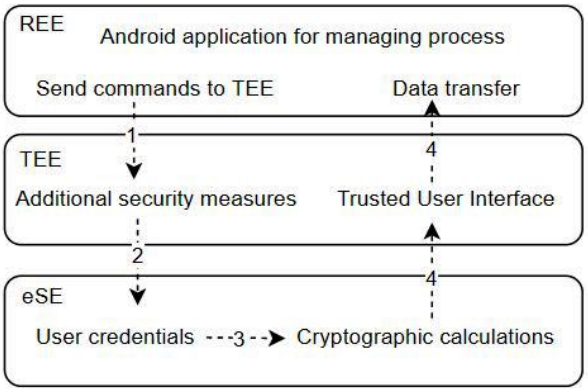

Fig. 2. mDL lifecycle on mobile device

5.3. Approach weaknesses

The proposed approach has the following weaknesses:

1) As it is proposed to use TEE, it will be necessary to obtain the consent of the vendor, the keys for the signing Trusted Application [8] to have ability install it to the TEE.

2) The same issue with obtaining the consent from the eSE vendor to be able to install the software to the eSE.
However applets in eSE written on Java, this is not quite ordinary Java, it is a special stripped-down Java developed for JavaCard which has its own API and a lot of things are cut from it, which will also lead to an increase of development time and cost of developing the necessary software for the system.

3) Increases the complexity of development, respectively, the cost and duration, in order to take into account all the security measures in designing such a system.
But from the other hand all the weaknesses of the system and possible high cost of development are leveled by the level of its security which can provided by realization of all the above mentioned technologies.

6. Deployment model

First of all, when it comes to a system that will function at the state level [2], the relevant laws governing electronic credentials and identity identification should be adopted.

The next step is to reach an agreement between government and the device vendors in order to get keys to have possibility installing the necessary software to the eSE and TEE.

Following is to deploy the system itself. The main element of this system is an Issuing authority. Its primary task is to generate and sign credentials of the user, secondary is to store all generated certificates and act as server for data verification if needed.

Next, Credential Management System requests specified signed credentials from the Issuing authority of a certain user and passes it to the Trusted Service Manager.

Trusted Service Manager is a key element of the system. Its stores all the necessary keys from eSE and TEE to be able to provision the necessary data and install required software.

The final element of this system is a mobile device of end-user. The necessary software will be installed on the device and signed credentials will be provisioned.

The proposed model is shown in Figure 3.

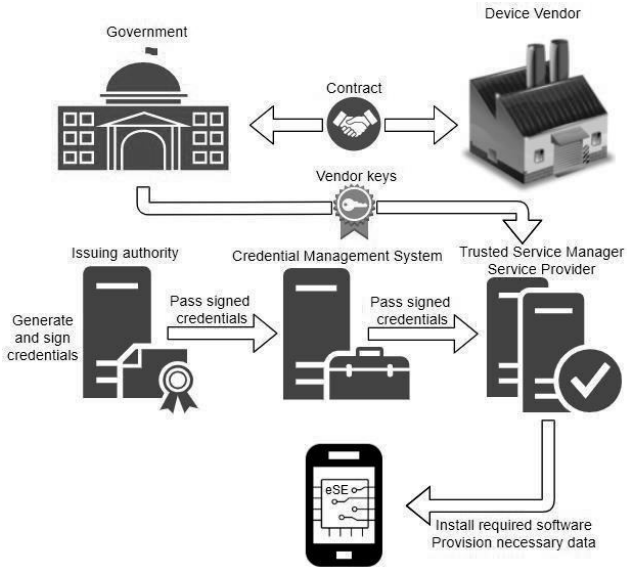

Fig. 3. Deployment model of mDL system

7. Conclusion

In this paper various approaches with different combination of technologies which provides security to storing sensitive user data on a mobile device were analyzed and identified advantages and disadvantages of these approaches.

After analysis of different approaches based on provided level of security approach was selected for further investigation on how to apply measures for security enhancement. Security enhancement measures were purposed on the basis of this approach to increase provided security level for the system.

Based on investigation of how to install all necessary software to mobile device, the deployment model of electronic driver's license system is presented.

In the future, it is planned to develop a more efficient deployment model of system that could reduce the cost of resources for implementation

References

[1] "ISO/IEC CD 18013-5 information technology — personal identification — iso-compliant driving licence." https://www.iso.org/standard/69084.html. Accessed: 2019-09-17.

[2] "Digital driver's license." https://www.gemalto.com/govt/traffic/digital-driver-license. Accessed: 2019-09-17.

[3] Z. Ahmad, L. Francis, T. Ahmed, C. Lobodzinski, D. Audsin, and P. Jiang, "Enhancing the security of mobile applications by using TEE and (U)SIM." http://www.isg.rhul.ac.uk/~lishoy/ATC2013LishoyFrancis.pdf. Accessed: 2019-09-17.

[4] B. Candaele, D. Soudris, and I. Anagnostopoulos, "Arm trustzone," in *Trusted Computing for Embedded Systems*, pp. 35–45, 2015.

[5] Z. Chen, L. Frindly, T. Lindholm, K. Arnold, and J. Inscore, "Architecture and programmer's guide," in *JavaCard Technology for SmartCards*, pp. 3–10, 2000.

[6] "Introduction to secure elements." https://globalplatform.org/wp-content/uploads/2018/05/Introduction-to-Secure-Element-15May2018.pdf. Accessed: 2019-09-17.

[7] D. Kelts, "The new mobile driver licenses." https://medium.com/@dkelts.id/mobile_driver_licenses_mdl_how_to_use_iso_18013_5_5a1bbc1a37a3. Accessed: 2019-09-17.

[8] "Mobile ID: Realization of mobile identity solutions." http://www.securitydocumentworld.com/creo_files/upload/article-files/GlobalPlatform_White_Paper_MobileID.pdf. Accessed: 2019-09-17.

[9] N. Mead, "The common criteria." https://www.us-cert.gov/bsi/articles/best-practices/requirements-engineering/the-common-criteria. Accessed: 2019-09-17.

[10] "Secure element certification process." https://globalplatform.org/wp-content/uploads/2019/03/GP_SE-certification-process-v1.0.pdf. Accessed: 2019-09-17.

[11] R. Hayton, "The benefits of trusted user interface." https://www.trustonic.com/news/blog/benefits-trusted-user-interface/. Accessed: 2019-09-17.

[12] "What is serial synchronous interface." https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019MgLSAU. Accessed: 2019-09-17.