ISSN: 0976-3031

Available Online at http://www.recentscientific.com

CODEN: IJRSFP (USA)

International Journal of Recent Scientific Research
Vol. 9, Issue, 5(G), pp. 26963-26967, May, 2018

International Journal of Recent Scientific Research

DOI: 10.24327/IJRSR

Research Article

SECURE ELEMENT – AN OVERVIEW

Ruchi Rautela., Devesh Manjrekar and Isha Gondhalekar

Department of MCA, VESIT Mumbai, Maharashtra, India

DOI: http://dx.doi.org/10.24327/ijrsr.2018.905.2159

ARTICLE INFO

Article History:

Received 10th February, 2018
Received in revised form 6th March, 2018
Accepted 24th April, 2018
Published online 28th May, 2018

Key Words:

NFC; SE, UICC, microSD, RFID, APSD.

ABSTRACT

Secure Element (SE) is being used in coordination with Near Field Communication (NFC) for reasons primarily concerned with security. NFC payment services like Apple Pay, Samsung Pay and Android Pay heavily rely on their SE for secure transactions. Since NFC technology transfers sensitive data during payment, the importance of SE is therefore highlighted. SE is a tamper resistant chip that facilitates the secure storage and transaction of payment and other sensitive credentials. This paper focuses on SE with respect to security aspects. SE architecture and forms of SE are also discussed in detail. Paper also explains the working of each form of SE.

INTRODUCTION

Today, wireless communication is the most important and desirable form of communication. These communication technologies include Wi-Fi, Bluetooth, and NFC etc. NFC is a type of technology which is based on the existing standards provided by RFID infrastructure. NFC operates at a radio frequency of 13.56 MHz, within a range of 4 - 10cms and is used widely for data exchange and mobile payments [1]. Security plays a crucial role when exchange of sensitive data is involved. Such an exchange takes place while making NFC payments. The responsibility of securing such valuable data lies with the SE. This sensitive information can be as simple as credit/debit card details. When touched to an NFC reader, the reader reads the data stored in the SE of the smartphone. For this reason, SE has been adopted by large organizations for purposes primarily related to payments.

Paper organization is done as follows: In the following section, implementation of SE is presented. In Section III, SE is reviewed. Section IV Explains the necessary standards associated with SE. Architecture of SE is analyzed in Section V. Possible attacks scenarios are discussed in Section VI. Section VII coves the API's for secure element access. Finally, the paper is concluded in Section VIII.

Background

SE simply is a secure chip residing traditionally in a NFC-enabled device. A typical architecture of a secure chip consists of various components like secure microcontrollers, CPU, operating system, memory, crypto engines, timers, communication ports etc. When a NFC application demands high levels of security such as payment applications, it is stored inside these secure chips called secure element.

SE provides dynamic environment where the application code, its confidential data is stored and the application code is securely executed. For example: for a payment application all the personal data like account number, expiry date, passwords, card numbers are stored in secure element and then the safety of its secret information can be trusted upon. To be more specific, application inside the SE performs several tasks like, handshaking with the POS terminal, responding to queries received from terminal, authenticating the card, filtering data to be shared etc. but it is the SE that provides secure execution environment for applications to perform all their defined tasks [6].

SE

As defined by Global Platform, SEis a tamper-resistant platform, typically a one chip secure microcontroller capable of securely hosting applications and their confidential and cryptographic data (e.g. key management) in accordance with the rules and security requirements set forth by a set of well-identified trusted authorities.

There are three different form factors of SE. Removable form factors are universal integrated circuit card (UICC) SE and

Corresponding author:* **Ruchi Rautela
Department of MCA, VESIT Mumbai, Maharashtra, India

microSD SE. Embedded Secure Elements are chips directly bonded on the motherboard of the device [7].

Secure Element (SE) is a microprocessor chip which can store sensitive data and run secure apps such as payment. It acts as a vault, protecting what's inside the SE, applications and data from malware attacks that are typical in the host which is the device operating system.

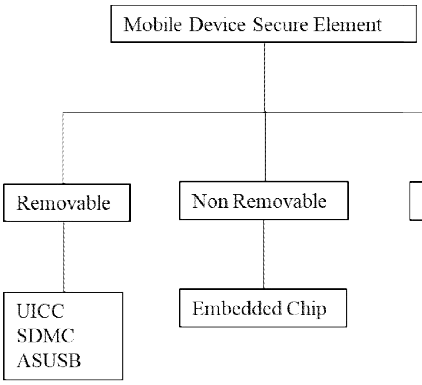

Figure 1 Elements of SE

The communication through SE is shown below.

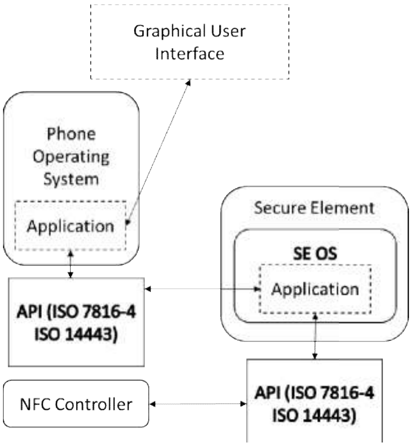

Figure 2 Communication structure of SE

UICC SE

The UICC is smart cards used in cellular telephony, which can be dispatched in different sizes. They are called SIM, which is actually the name of the application presented by the UICC to access GSM networks. Smart SD cards have a similar form as usual SD cards, but internally include a SE, and support an extended set of SD commands to communicate with the SE.

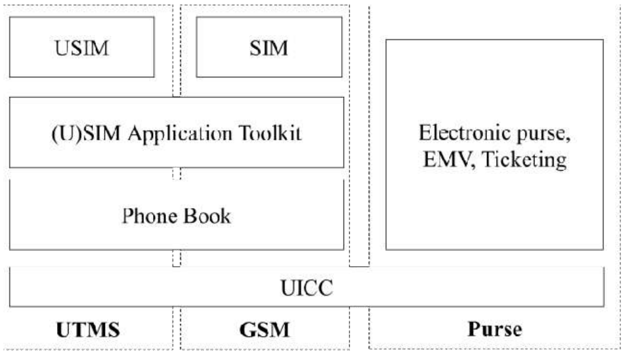

Figure 3 UICC SE Structure

Embedded SE

The secure element can be integrated into the phone itself by the phone manufacturer as an Embedded Secure Element since it is a part of the phone. It is issued and managed either by the Original Equipment Manufacturer (OEM) or Mobile Network Operator (MNO).

MicroSD SE

It transforms smartphones into contactless mobile payment devices with NFC microSD secure elements. The microSD secure element provides issuers and consumers with a variety of highly secure mobile payment devices. Data encryption technologies protect personal information through a personalization workstation with a smart card reader to help ensure the secure element can only be used by the microSD owner.

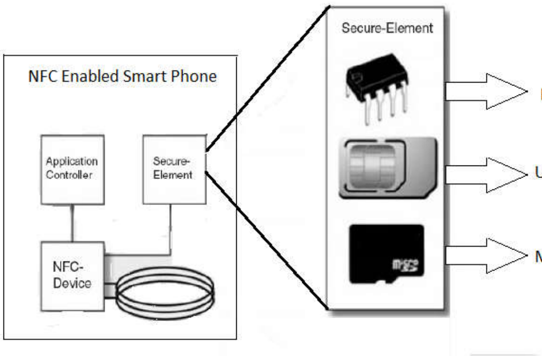

Figure 4 Form Factors of SE

Placements of SE: The Secure elements are placed in the mobile phone and its communication flow architecture is shown in the Figure 5 for SE on an embedded chip, Figure 6 for SE on an external SD memory card and Figure 7 for SE on a UICC.

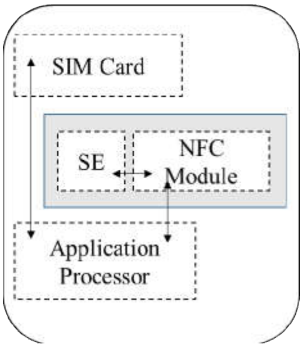

Figure 5 SE on an embedded chip

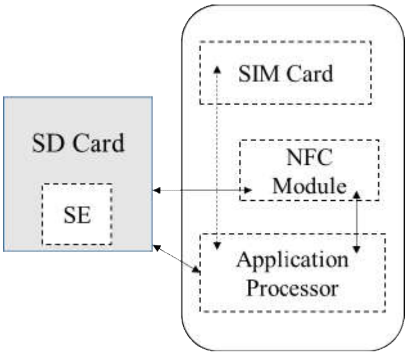

Figure 6 SE on an external SD memory card

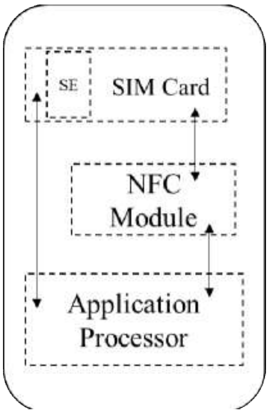

Figure 7 SE on a UICC

Secure Elements handle all sorts of applications that are vital to our modern digital lives:

Authentication

Instead of user name and password, access to an online service may be protected by a strong authentication mechanism, based on credentials stored and processed in the secure element. So, to log into a VPN or your email, a Secure Element could be involved in the background to ensure you are who you say you are.

Digital Signature

Applications may use the SE to digitally sign a document or any data with a key stored in this secure element. This key helps the secure element unlock encrypted data so it can be read. Again, this is used to prove you are you. So, your email program could use connect to the Secure Element to digitally "sign" emails you send, or a government web application could access it when you are using their digital services.

Mobile Payments

Here, the Secure Element securely stores card/cardholder data and manages the reading of encrypted data. During a payment transaction, it acts like a contactless payment card using industry standard technology to help authorize a transaction. The Secure Element could either be embedded in the phone or embedded in your SIM card.

Lifecycle Management

It's crucial that SE-embedded devices are secure throughout their lifecycle. That's why Secure Elements need to have an end-to-end security strategy. It's no use developing a robust security solution for a device which becomes obsolete after a period of use. This is why Secured Elements can be updated continuously to counter new threats.

Standards

ISO/IEC 7816

ISO/IEC 7816 is a multi-part international standard broken into fourteen parts. ISO/IEC 7816 Parts 1, 2 and 3 deals only with contact smart cards and define the various aspects of the card and its interfaces, including the card's physical dimensions, the electrical interface and the communications protocols [9].

Single Wire Protocol (SWP)

SWP is an interface between Contactless frontend (CLF) and UICC (SIM card chip). It is a contact-based protocol which is used for contactless communication. C6 pin of UICC is connected to CLF for SWP support. It is a bit oriented full duplex protocol i.e. at the same time transmission as well as reception is possible. CLF acts as a master and UICC as a slave. CLF provides the UICC with energy, a transmission clock, data and signal for bus management. The data to be transmitted are represented by the binary states of voltage and current on the single wire [8].

Architecture

Secure Element follows the widely accepted HTTP Client – Server communication model. Due to this fact its deployment becomes compatible with existing infrastructure and also inherits the advantages of HTTP communication like scalability and high availability [6].

To initiate interaction, a communication session is established between client and server. All further communication taken place within that session. Often the client is responsible for initiating the communication session.

The main components involved in SE architecture are:

1. Service provider
2. Admin Server
3. Mobile phone
4. Secure Element
5. Application

In this arrangement, the admin server acts as HTTP server and admin agent act as HTTP client. The communication session is termed as administrative session.

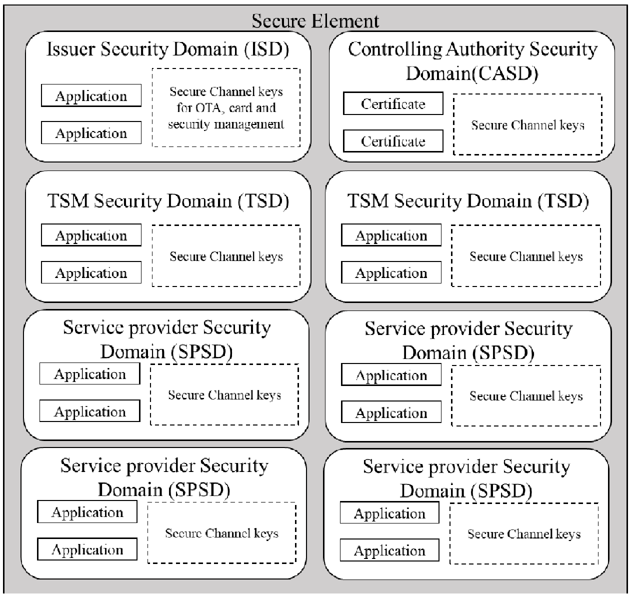
Secure Element
Issuer Security Domain (ISD)
Application
Application
Secure Channel keys for OTA, card and security management
Controlling Authority Security Domain(CASD)
Certificate
Certificate
Secure Channel keys
TSM Security Domain (TSD)
Application
Application
Secure Channel keys
TSM Security Domain (TSD)
Application
Application
Secure Channel keys
Service provider Security Domain (SPSD)
Application
Application
Secure Channel keys
Service provider Security Domain (SPSD)
Application
Application
Secure Channel keys
Service provider Security Domain (SPSD)
Application
Application
Secure Channel keys
Service provider Security Domain (SPSD)
Application
Application
Secure Channel keys

Figure 8 Architecture of SE

The initiation of this admin session can be triggered by:

- A device external event, for example a Push message sent by an Admin Server. Here the request is initiated by admin server
- An Admin Agent internal event, for example a timer. Here the request is initiated by admin agent (client)

- An administered SE application request. Here the request is initiated by application residing in SE.
- A device application request. Here the request is initiated by a mobile application.

The communication between the admin server and admin agent takes place by using HTTP protocol. The admin agent is responsible for interpreting these HTTP responses from server and further converting them into APDUs understandable by application residing in SE. This structure of HTTP request/response and APDUs exchanged between admin agent and application is defined by Global platform specification [6,7].

Depending on the type of secure element the admin agent can be implemented in following ways:

1. ***For UICC SE:*** Admin agent is implemented in the application security domain i.e. APSD. The application resides in the secure element which in turn resides in the UICC card.
2. ***For Embedded SE:*** Admin agent is implemented in the trusted execution environment in mobile phone
3. ***For a micro SD SE:*** Admin commands will be forwarded to SE through mobile application implemented using JSR 177 for java ME.

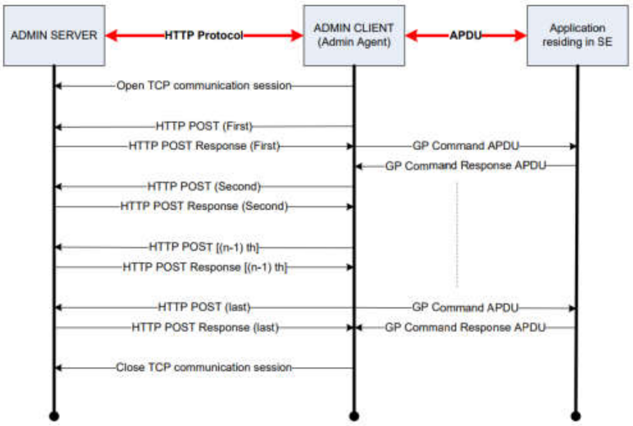
ADMIN SERVER
HTTP Protocol
ADMIN CLIENT (Admin Agent)
APDU
Application residing in SE
Open TCP communication session
HTTP POST (First)
HTTP POST Response (First)
GP Command APDU
GP Command Response APDU
HTTP POST (Second)
HTTP POST Response (Second)
HTTP POST [(n-1) th]
HTTP POST Response [(n-1) th]
HTTP POST (last)
GP Command APDU
HTTP POST Response (last)
GP Command Response APDU
Close TCP communication session

Figure 9 Trigger event

Attacks Scenarios

The application programming interface and the resources of mobile devices are usually restricted by access control policies. Access to critical parts of the system is typically shielded from untrusted applications. However, these are usually software-based restrictions enforced by the operating system. Thus, if an untrusted application can manipulate the behavior of the operating system or if it can elevate its privileges to the level of a trusted application, it can easily circumvent any access restrictions [3,5].

Popular techniques used on many smart phones are "jail breaking" and "rooting". Jail breaking refers to escaping the restrictions imposed by the operating system, so that an application can access resources it usually could not access. Rooting refers to an even sever scenario where the user or an application gains full access to the whole system. Both methods are often used intentionally by device owners/legitimate users to circumvent digital rights management or to gain improved control over their device [3].

A relay attack scenario allows an attacker to remotely use a victim's secure element over a network connection. For instance, at point of sale or an access control gate, nobody would suspect that the communication is actually relayed to a remote device [3,10].

The relay software is installed on the victim's mobile phone. This application is assumed to have the privileges necessary for access to the secure element and for communicating over a network. These privileges can be either explicitly granted to the application or acquired by means of a privilege escalation attack [10].

API'S *For Secure Element Access*

The secure element in an NFC mobile phone can be accessed from two sides. In external mode, the secure element emulates a contactless smartcard to external RFID/NFC hardware. In internal mode, the secure element is accessible from mobile phone applications. The various mobile phone platforms define different APIs and access restrictions for the internal mode.

JSR 177: The Security and Trust Services API (SATSA, JSR 177 [6]) specifies a number of Java programming interfaces for integrating secure elements into Java applications. Specifically the sub-package SATSA-APDU is designed for APDU-based communication with secure elements. JSR 177 is defined for Java ME which is a Java platform specifically designed for devices with limited processing and storage capabilities – like mobile phones. Access to the SATSA-APDU API is protected by Java ME permissions. The permissions for smartcard access are only granted to signed applications. Applications in the manufacturer domain and the operator domain are automatically granted the permission while applications in the trusted third party domain may require additional user interaction in order for the permissions to be granted. As an addition to this basic access control scheme, the SATSA specification recommends a more sophisticated access control model in order to protect the secure element from malicious mobile phone applications. The The SATSA specification makes some important assumptions for the access control model to be secure: Mobile phone applications are bound by all secure element access restrictions and both the mobile phone application and the applet trust the mobile device platform [6].

Nokia's Extensions to JSR 257: The Contactless Communication API (JSR 257 [7]) specifies a Java programming interface for access to contactless targets (RFID/NFC tags and visual tags). Consequently, this API provides access to NFC's reader/writer mode. For their first NFC phones (specifically Nokia 6131 and Nokia 6212), Nokia developed some extensions to the Contactless Communication API in order to support more features of NFC. Besides support for further RFID tag types and for some limited peer-to-peer functionality, Nokia's extensions to JSR 257 also provide access to the embedded secure element of their mobile phones. Both JSR 177 and JSR 257 provide access to smartcards. While JSR 177 is intended for access of specific applets on secure elements connected to or integrated into a mobile device, JSR 257 is intended for access to any contactless smartcards that are accessed through a device's RFID/NFC reader. JSR 257 provides an interface ISO14443 Connection for creating connections to contactless smartcards. This compensates for the missing support of access to the embedded secure element through JSR 177 on their devices. Opening an ISO14443 Connection is subject to protection by Java ME permissions. This scheme requires that an application is in the manufacturer, the operator or the trusted third party security domain. Therefore, only applications that are signed with trusted certificates are granted access to the secure element.

BlackBerry 7: BlackBerry uses an interface similar to SATSA-APDU for secure element communication. Additional helper classes provide information on the available secure elements and allow for easy instantiation of APDU Connection objects. Access to the secure element API is restricted to applications that are signed with BlackBerry Java code signing keys. Code signing keys are provided to developers free of charge but registration with Research In Motion (RIM) is required.

Android: While Android-based NFC-enabled mobile phones, like the Nexus S, have an embedded secure element and also support a UICC-based secure element, Android still does not have a public API for secure element access. Since Android 2.3.4 access to the embedded secure element is possible through an API called com.android.nfc_extras, but this interface is not included in the public software development kit (SDK). This API contains two classes: NfcAdapter Extras and Nfc Execution Environment. NfcAdapte rExtras is used to enable and disable external card emulation and to retrieve an instance of the secure element's NfcExecution Environment class. Nfc Execution Environment is used to exchange APDUs with the embedded secure element. In Android 2.3.4 the NFC-Extras API could be accessed by any application that held the permission to use NFC. In later versions this has been changed to a special permission named com.android.nfc.permission.NFCEE_ADMIN. This special permission is only granted to applications which are signed with the same certificate as the NFC system service. Consequently, access to the secure element is restricted to applications trusted by the manufacturer/provider of the NFC system service.

CONCLUSION

Being a wireless technology NFC is based on RFID standards and operates in three different modes for communication. RFID and Magnetic Induction have key roles in the implementation of NFC. Similarly, SE has the role of security in one of the application of NFC which is NFC payments. SE is responsible for securing data during transactions done using NFC. For this SE comes in three form factors namely UICC SE, Micro - SD SE and embedded SE. All three of these form factors are responsible for security during payment via NFC. SE architecture describes its implementation in a mobile phone. SE has its respective standards which are ISO/IEC-7816 and SWP. Each of these standard deals and defines with communication between different elements of NFC architecture and SE. Although SE is responsible for providing high level security during NFC payments it has a shortcoming which is relay attack. Despite this shortcoming, Android Pay, Apple Pay and Samsung Pay still heavily rely on their respective version of SE for security.

References

1. S. Nahar, S. Kajarekar, D. Manjrekar, and S. Kotian, "NFC and NFC payments: A review," *2016 Int. Conf. ICT Bus. Ind. Gov.*, pp. 1–7, 2016.
2. M. Reveilhac and M. Pasquet, "Promising secure element alternatives for NFC technology," *Proc. - 2009 1st Int. Work. Near F. Commun. NFC 2009*, pp. 75–80, 2009.
3. M. Roland, J. Langer, and J. Scharinger, "Practical attack scenarios on secure element-enabled mobile devices," in *Proceedings - 4th International Workshop on Near Field Communication, NFC 2012*, 2012, pp. 19–24.
4. G. Madlmayr, J. Langer, C. Kantner, and J. Scharinger, "NFC devices: Security and privacy," *ARES 2008 - 3rd Int. Conf. Availability, Secur. Reliab. Proc.*, pp. 642–647, 2008.
5. N. A. Chattha, "NFC - Vulnerabilities and defense," *Conf. Proc. - 2014 Conf. Inf. Assur. Cyber Secur. CIACS 2014*, no. 1, pp. 35–38, 2014.
6. S. Rohilla, "Secure Element An evolution to existing secure technology", *International Journal of Scientific and Research Publications*, vol. 5, no. 7, pp. 1-5, 2015.
7. "GlobalPlatform made simple guide: Secure Element", *GlobalPlatform*. [Online]. Available: https://www.globalplatform.org/mediaguideSE.asp.
8. "Single Wire Protocol", *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Single_Wire_Protocol.
9. "ISO/IEC 7816", *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/ISO/IEC_7816.
10. M. Roland, J. Langer, and J. Scharinger, "Relay attacks on secure element-enabled mobile devices: Virtual pickpocketing revisited," in *IFIP Advances in Information and Communication Technology*, 2012, vol. 376 AICT, pp. 1–12.
11. https://www.justaskgemalto.com/en/what-is-a-secure-element/

How to cite this article:

Ruchi Rautela *et al.*2018, Secure Element – An Overview. *Int J Recent Sci Res.* 9(5), pp. 26963-26967.
DOI: http://dx.doi.org/10.24327/ijrsr.2018.0905.2159

****