Over-the-Internet: Efficient Remote Content Management for Secure Elements in Mobile Devices

Mohamed Sabt, Mohammed Achemlal, Abdelmadjid Bouabdallah

Over-the-Internet: Efficient Remote Content Management for Secure Elements in Mobile Devices

Mohamed Sabt*† Mohammed Achemlal*‡ and Abdelmadjid Bouabdallah†
*Orange Labs, 42 rue des coutures, 14066 Caen, France
{mohamed.sabt, mohammed.achemlal}@orange.com
†Sorbonne universités, Université de technologie de Compiègne,
Heudiasyc, Centre de recherche Royallieu, 60203 Compiègne, France
{mohamed.sabt, madjid.bouabdallah}@hds.utc.fr
‡Greyc ENSICAEN, 6 Bd Maréchal Juin, 14050 Caen, France

***Abstract*—We propose Over-the-Internet (OTI), a novel system that manages secure element based applications. We demonstrate our solution in the context of NFC ecosystem and show that it can be effectively used for transmitting big applications to the secure element. Our system leverages the GlobalPlatform card specification as well as the GlobalPlatform user-centric ownership model. Our solution integrates the different actors of the NFC ecosystem in its architecture. We propose to leverage the concept of security domain, so that service providers can manage their applications independently from the SE issuer. We implement our solution within available platforms and show that it is secure, fast, reliable and easily deployable.**

I. Introduction

The use of mobile devices such as smartphones has greatly increased in the last years. This rise has the effect of spurring service providers on to deploy their own services for mobile markets. Some of these services, such as mobile payment and e-ticketing, require user authentication and data confidentiality; hence the need to a highly secure environment in which applications should run. The most popular example of such services are NFC applications. The NFC technology is primarily used to make contactless transactions, including those for access, payment and ticketing. The secured area that is used to provide secure execution for applications is commonly referred to as *Secure Element* (SE). The secure element is an environment in which application code and application data can be securely stored. Secure elements come in several flavors. They could be implemented either by an embedded smart card chip, in an SD card that can be inserted in the mobile, or in the SIM/UICC which is used by phone operators to authenticate subscribers on their network.

Regardless of their form-factor, secure elements have known enormous improvement and enhancements in the last two decades. They become programmable and can support multiple applications [1]. Furthermore, applications could be downloaded to secure elements that already have been issued to users. This is a great evolution in the world of secure elements, since applications were traditionally pre-installed on the card and no post-issuance application loading was possible. This inflexible model is no longer appropriate in the open world of smartphones where there exists a huge number of applications proposed by many service providers and users usually prefer to freely choose the applications that they want to use on their mobiles. For applications based on secure elements to succeed, secure elements need to be 'smart', in the sense that users could easily manage their content. For instance, a user might install an NFC payment application that she uninstalls a few days later because she finds another application that proposes better services. The service provider might need to update their application in order to offer more services or to fix some security vulnerabilities.

Post issuance management of the *smart* secure elements can be quite complex. Unfortunately, management systems have not followed the evolution of secure elements. Indeed, current systems are not adapted to download an application of several kilobytes to the secure element. Instead, they were designed to accomplish two main tasks. Firstly, they send a code to activate the pre-installed application after authenticating the user. Secondly, they send some personalization parameters of only a few bytes of size to customize the activated, generic application. Unlike the personalization parameters, the size of secure element applications is measured in kilobytes. Their size is often above 9 kilobytes. Being designed to send a few bytes only, the existing management systems are incapable of reliably transmitting such amount of data for thousands of users.

Nowadays, the technology of secure elements is quite mature. However, the market is still reluctant to adopt it because of the lack of remote content management systems. This lack proves problematic for both users and service providers. Users of smartphones are accustomed to choose between a wide range of applications, and therefore they are naturally uncomfortable with any system that forces them to choose from a limited list of pre-installed applications. As for service providers, they cannot easily deliver their applications to users. Even simple operations, such as updating or deleting applications become hard to accomplish. In summary, Remote content management is essential for applications based on secure elements to achieve their potential. However, the current systems are not reliable enough to perform complex content management operations, including downloading several kilobytes of code, updating code or credentials, and deleting applications. Programmable secure elements offer great opportunities due to their high security, but their future suffers from outdated remote content management systems.

In this paper, we propose an Internet-based remote content management system for secure elements in connected

devices, such as smartphones. In contrast to previous work, our work does not require changes to existing secure elements or mobile devices and is compatible with the existing content management standards. We leverage the widely accepted card specifications of GlobalPlatform. We make the following contributions:

- We provide a complete and feasible solution for remote management of secure elements in connected devices. We call our solution Over-the-Internet (OTI) as it makes use of TCP/IP connection to transmit code or management commands to secure elements. We leverage the growing speed of the Internet and the proven reliability of the TCP protocol.
- We leverage the recently defined GlobalPlatform consumer-centric model. This model allows end users to control the applications that are installed on their secure elements.
- We demonstrate our solution in the context of NFC applications. There exist various contexts for the secure element applications, and designing a general content management system for a general context is difficult. We have selected the NFC ecosystem because nowadays NFC applications that are based on secure element are increasingly popular. It is worth noting that our solution is naturally extensible to manage any kind of secure element based applications.
- We present full implementation of our solution using commercially available secure elements and devices without any changes. Our prototype shows that OTI is by far faster and more scalable than the existing technologies.

The rest of the paper is structured as follows. Section II defines the problem that we address in more detail. We give a background information on secure elements and OTA architecture in section III. In section IV, the Over-the-Internet architecture is presented, including analysis of its security properties. Section V provides implementation details and performance evaluation. Section VI surveys related work, and we end the paper with a brief summary.

II. Problem Statement

Smart cards have evolved from a single application platform to a feature-rich multi-application platform. However, multi-application smart cards have not successfully evolved into multi-purpose cards. The card issuer, the entity that issued the card, still has full control over the applications installed on its cards. In the traditional smart card industry, cards are issued for one or a specific list of service providers. Cardholders use their smart cards as service enablers, and not as a secure computer platform which can host several applications developed by independent service providers. Traditional multi-application smart cards are not really 'smart'.

The advent of the Near Field Communication (NFC) has influenced the smart card technology. It enables a mobile device to emulate a contactless smart card. The emulated NFC card can be accessed by an external contactless reader, such as contactless-enabled point-of-sale terminals [2]. The NFC technology allows the introduction of smart cards, called secure elements, into mobile devices. Modern mobile devices, namely smartphones, and secure elements have conflicting traditions. Secure elements are issuer-centric where users are not allowed to freely install applications, while smartphones are user-centric where an average user can easily navigate, search, install third party applications. In addition, the application developers do not have to negotiate with the mobile operators in order to be able to propose their applications to users.

Coupled with a smartphone, the secure element cannot be isolated from the so-called "iPhone effect". In 2012, GlobalPlatform proposed a user-centric model for secure elements in [3]. This new model enables secure elements to achieve the full potential of the multi-application smart card technology. Nevertheless, the absence of appropriate application management system has hindered the flourishing of the new user-centric secure elements.

At present, there is no dedicated platform to perform application management. Instead, applications are managed by the OTA platform that is used for UICC remote provisioning. The transport bearer of the current OTA platform is the SMS bearer. The SMS bearer is suitable for small amount of data. More precisely, one SMS can contain only 140 bytes of data. Therefore, it is not suitable for loading heavy applications (e.g., above 9 kilobytes). Some solutions suggest to use the Bearer Independent Protocol (BIP) [4] because it is more reliable and faster than SMS. In spite of being standardized, BIP is not widely implemented in mobile devices [5].

Our goal is to design a content management system for NFC enabled services that use the secure element in the mobile device. In the following, we detail the *requirements* for any efficient and secure solution to this problem.

We first aim to design a solution that could be deployed with ease. A solution that involves many changes to the existing architecture is unlikely to be welcomed in the production environments. *Deployability* is a non-functional requirement, and like most non-functional requirements, it is often overlooked. However, it is a highly regarded requirement for the NFC ecosystem, where standards are predominate and cannot be neglected. Nevertheless, our primary goals are efficiency and security, and they are not traded off against deployability. The deployability requirement implies that our solution must be compliant with the GlobalPlatform Card Specification [6] which is considered as the de facto standards for the management of secure elements.

Second, the content management must be performed via a wireless technology. This is beneficial for both users and service providers. Wireless technology allows NFC applications to become a few clicks away from users' mobile devices. The users would enjoy the experience, and the service providers would attract more clients with less cost. The wireless technology must be fast enough to allow new applications to be installed in a reasonable time, and reliable so that no transmission errors affect the correct delivery of data.

Third, security is of paramount importance. Only authenticated *applets* from trusted service providers could be installed into the secure element. Otherwise, an attacker might install malicious code to exploit security flaws in the secure element software. The proposed solution must ensure a secure end-to-end connection between the secure element and the service

provider to ensure a secure management for the content of the secure element. Therefore, some secret keys must be shared between the secure element and the service provider.

Fourth, an NFC application running in the secure element is often accompanied by a graphical user interface running in the mobile device. This user interface is the feature that gives NFC applications advantage over contactless smart cards because it improves users interaction with the application. Applets and graphical interface are interdependent. Indeed, any update to the secure element applet might entail updating the user interface, otherwise the NFC application may not function properly. The problem is that if the user interface is a native mobile app, its life-cycle is managed independently from that of the secure element. A content management system for secure elements must include mechanisms to update the user interface when necessary.

III. Background

A. Secure Element

A secure element (SE) is a tamper-resistant smart card chip capable of running applications (called applets or cardlets) with a high level of security. A smart card is essentially a minimal computing environment with a CPU, ROM, EEPROM, RAM, and I/O port. Recent cards also include cryptographic co-processors that implement common algorithms such as AES and RSA [7]. Modern smart cards can host multiple applications, but only one application can be executed at a time.

The two main de facto standards of multi-application cards are Java Card [8] and GlobalPlatform card specification [6].

1) Java Card: Java Card is a Java-based smart card. It is designed by adding a lightweight Java bytecode interpreter to a smart card's OS and downloading Java class files, which were converted to a specific format beforehand. The system architecture of the Java Card is mainly composed of three layers: the Java Card Virtual Machine (JCVM), the Java Card Runtime Environment (JCRE) which provides a well-structured framework to access the system-level services, and finally the Java Card firewall which isolates the different applets present on the card from each other. The Java Card specification does not specify the mechanism of installing, updating, or deleting applications. The application management is defined by GlobalPlatform card specification.

2) GlobalPlatform: The *Card Manager* is the entity that controls the smart card environment. It contains the OPEN framework that controls the downloading and installing of applications. In addition, it contains the issuer security domain. The security domain is a generic term used to divide the card into independent regions. Each region is associated to separate cryptographic keys that allow the security domain owner to manage its applications without the card issuer's involvement. Each application is associated to a security domain. The security domain created by the card issuer, commonly called the issuer security domain, is the component that controls the security domains. However, it has no access to their content or to their cryptographic keys.

B. Over-the-Air (OTA)

Over-the-Air (OTA) platform was originally designed by mobile operators to provide remote management of their (U)SIM cards. Mobile operators use their OTA platforms for activation and provisioning parameters. The communication between the OTA platform and the (U)SIM is established using APDU command/response encapsulated into SMS messages. The structure of the APDU command is defined in [9]. ETSI has introduced the TS 102 226 [10] standard in order to support security services for OTA communication.

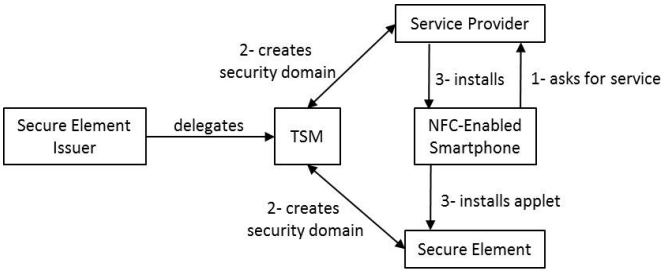

Fig. 1. An Overview of the OTI Architecture

IV. Over-the-Internet Design

A. Overview of the Architecture

The Over-the-Internet framework (OTI) applies the user-centric model to the NFC ecosystem. It allows users to install secure element based NFC applications as effortlessly as they install any other smartphone app. OTI transmits applications to the SE via Internet (e.g., wifi or 3/4G connection). OTI leverages the concept of security domain to enable secure management of applications developed by different service providers. In addition, it leverages the secure channel protocol (SCP) to provide end-to-end secure connection between the service provider and the secure element.

Figure 1 shows an overview of the OTI platform. The architecture consists of the following components:

1) Secure Element Issuer: it is the owner of the SE and the responsible of its long-term content management. All secure elements have a mandatory representative of the issuer, known as issuer security domain (ISD). In most cases, issuers delegate the life cycle management of SE to a trusted service manager.
2) Trusted Service Manager (TSM): the TSM acts as the connection point between service providers and the SE issuer. It creates security domains on the SE in order to provide secured blocks of space for service providers.
3) Service Provider (SP): in the user-centric model, it is the service provider that manages the life cycle of applications in their security domain created by the TSM. The TSM has no access to the content of an SP security domain. However, the TSM still could delete the SP security domain and all its contents in case of disagreement.
4) NFC-Enabled Smartphone: most NFC applications are not composed only of an applet on the SE, but they also include a graphical interface. The graphical interface displays some messages to the user.
5) Secure Element (SE): it hosts the sensitive part of the NFC application.

The OTI framework works as follows: the user looks for an applications on her favorite smartphone app store. She selects an application to install on her mobile device. The install process is started, and the corresponding service provider accomplishes three tasks. First, it installs the graphical interface on the mobile device. Second, it asks the TSM to create a security domain in the corresponding SE. The TSM sends the credentials of the recently created security domain to the SP. Third, the SP installs the applet on the SE. The process of installing is described in the following subsection.

B. Application Management

The graphical interface includes a module whereby the SP can exchange command/response messages with the SE. We call this module the "SE Manager". The management of SE applets includes installing, updating and deleting the applet. Each operation is performed by a specific set of commands. We leverage the GlobalPlatform card specification to build these commands. Sending the management commands to the SE is achieved by three steps.

First, the SP sends a push notification to the SE manager. Once notified, The SE manager opens a TCP connection with the SP server. As well, it opens a communication session with the SE. Second, the SP establishes a secure connection with the SE using the credentials of its security domain. The secure connection protects the confidentiality, integrity and freshness of the management commands. Such protection is necessary, since the commands go through untrusted parties, including the SE manager. Third, the management commands are encapsulated in the appropriate format before being transmitted to the SE. Once received, the SE performs some verification operations before executing the commands. The SP is notified of the execution status by a cryptographically protected response from the SE.

Figure 2 shows the layer model of the OTI application management.

C. Secure Channel Protocol (SCP)

Secure communication between the service provider and the secure element is of paramount importance. The integrity and the confidentiality of the management commands should be protected. The SE manager shall see nothing, but encrypted messages received from the SP and to be forwarded to the SE. We leverage the secure channel protocol 03 (SCP03) [11] to establish a secure communication channel between the the SP and the SE. SCP03 is an improved version of the SCP02. It is a symmetric key protocol that is divided into three phases:

Secure channel initiation: this includes mutual authentication between the SP and the related security domain on the SE. It also includes the generation of session keys.

Secure channel operation: the SP and the SE exchange messages within the cryptographic protection of the secure channel session. The channel protects the integrity and the confidentiality of the exchanged messages. The cryptographic operations are done by the AES encryption algorithm.

Secure channel termination: termination occurs when either of the communicating entities determines that no further communication is required or allowed via the secure channel session.

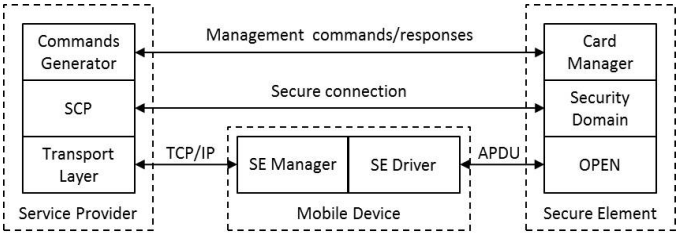

Fig. 2. OTI Layer Model

D. Design Analysis

The OTI framework overcomes the limitations of today's commonly used technology for remote applications (e.g., SMS based OTA). Its objectives are twofold. First, it enforces the "freedom of choice" of users. This is done by creating a space in the SE for each service provider. The SP manages its applications without the SE issuer's involvement. The OTI framework supposes that the infrastructure of the service provider is secure enough to protect the credentials of the created security domains. Second, and most importantly, the OTI framework leverages various technologies to provide faster and more reliable management for SE applications. Applications are transmitted from the SP server to the SE over TCP/IP connection. Since there is no direct connection between the SP and the SE, the communication must go through a bridge that receives the TCP/IP packets from the SP and forwards them to the SE. This bridge is a module installed on the mobile device. For more usability, this module is integrated in the graphical interface of the NFC application. The security of the transmitted application is guaranteed by the well-known and proven SCP03 protocol.

We use web technology for the graphical interface. All current smartphone systems allow native apps to display web pages and execute JavaScript code. Web technology allows for seamless updates of the graphical interface and the SE manager. This is important in order to avoid execution failure due to incompatible applications.

V. Implementation and Evaluation

We implemented a proof-of-concept of the OTI platform. Our prototype is divided into five parts:

1) Secure element: we used Oberthur UICC JavaCard 2.2.2 as the secure element. The used UICC is compliant with the GlobalPlatform card specification. The secure element is the only component in the OTI platform that does not require any modification or additional module.
2) NFC-enabled smartphone: we used Samsung Galaxy SII with Android 4.0.3 (ICS). SmartCard API [12] was installed to send APDU to the UICC.
3) User interface: the user interface was implemented as webview in the Android system in order to avoid the hassle of using the browser. The graphical interface is encoded in HTML5 and CCS3. The SE module was implemented as a JavaScript library. We called this library Ajase (Asynchronous JavaScript and Secure Element). Ajase is the implementation of the SE API recently defined in unofficial draft issued

TABLE I. COMPARISON OF DOWNLOAD TIME OF 9-KILOBYTE-JAVACARD APPLET

Content Management Platform	Average of Download Time
OTW (Over-the-Wire)	18.2 seconds
OTI (Over-the-Internet)	25.7 seconds
OTA (Over-the-Air)	5.42 minutes

by Gemalto and Deutsche Telekom [13]. The user interface is registered as Google Cloud Messaging (GCM) services to receive notification messages from the service provider.

4) Service provider: we integrated a full implementation of the GlobalPlatform card specification 2.2.1 written in Java 7 to the service provider. This library is used to generate the management commands and establish the secure channel.
5) TSM: we used OTA platform to create security domains for service providers.

Performance Evaluation

We conducted a comparison study between OTI (Over-the-Internet) and the SMS-based OTA (Over-the-Air). The study includes OTW (Over-the-Wire) because OTW is considered as the lower-bound of content management systems. OTW is the fastest and the most reliable content management system, since the secure element is directly connected to the TSM via a wire. However, in real context, OTW is not convenient because users prefer receiving updates via a wireless medium.

Table I shows the download time of JavaCard applet of size 9 kilobytes for different content management platforms. We notice that OTI largely outperforms OTA and comes very close to OTW. OTI is not only faster than SMS-based OTA, but it is also more reliable. More comprehensive evaluation study is kept for future work.

VI. RELATED WORK

The NFC ecosystem management is still an overlooked problem. Most work suggest to use SMS-based OTA platform to manage NFC applications, or more generally SE-based applications [14]. An industrial proposal is to replace SMS-based OTA by LTE-based OTA [15]. LTE is the fourth generation of mobile networks, commonly called 4G. LTE is a good substitute for SMS. However, unlike OTI, LTE-based OTA does not apply the user-centric model. The SE issuer has full control over the applications installed on the SE.

GlobalPlatform proposes in [16] to send management commands to the SE using HTTPS connection. This supposes that SE is able to receive and understand HTTP messages. This is true for JavaCard 3.0. However, most of the deployed secure elements are JavaCard 2.x. In [17], Badra et al. propose to combine SSL with BIP in OTA platform. As explained earlier, OTI does not rely on BIP because it is not widely supported by mobile devices. Compared to other work, OTI platform has the particularity to combine both effectiveness and deployability.

VII. CONCLUSION

We introduced OTI, an efficient management system for secure element based NFC applications in mobile devices. OTI transmits management commands over a TCP connection. Hence, it is more reliable than SMS-based OTA. In addition, it is more secure, since OTA platforms are vulnerable to well-known threats [18]. OTI solves the dilemma that previous approaches face because it does not have to trade off deployability and effectiveness.

This paper also introduced the current OTI implementation and experimental evaluation. Results indicate that OTI is an effective and practical management system for real world NFC ecosystem. Our future work will focus on providing a comprehensive framework for comparative evaluation of different application management systems.

REFERENCES

[1] C. Markantonakis, "The Case for a Secure Multi-Application Smart Card Operating System," in *Proceedings of the First International Workshop on Information Security*, ser. ISW '97. London, UK, UK: Springer-Verlag, 1998, pp. 188–197.

[2] "Information technology – Telecommunication and information exchange between systems – Near Field Communication – Interface and Protocol (nfcip-1)," NFC Std, July 2013.

[3] GlobalPlatform, "A New Model: The Consumer-Centric Model and How It Applies to the Mobile Ecosystem," White paper, March 2012.

[4] *Smart Cards; Card Application Toolkit (CAT), (Release 8)*, ETSI TS, January 2009, Technical Specification.

[5] Giesecke&Devrient, "Bearer Independent Protocol (BIP)," White paper, 2010.

[6] GlobalPlatform, "GlobalPlatform Card Specification, Version 2.2.1," http://www.globalplatform.org/specificationscard.asp, January 2011, last access: November 24 2014.

[7] W. Rankl and W. Effing, *Smart Card Handbook*, 4th ed. New Jersey, NJ, USA: Wiley, 2010.

[8] S. Microsystems, "Java Card(tm) Specification 2.2.2 Final Release," http://download.oracle.com/otndocs/jcp/java_card_kit-2.2.2-fr-oth-JSpec/, March 2006, last access: November 28 2014.

[9] *Smart Cards; Secured packet structure for UICC based applications (Release 9)*, ETSI Std. ETSI TS 102 225, April 2010, Technical Specification.

[10] *Smart Cards; Remote APDU structure for UICC based applications (Release 9)*, ETSI TS, April 2010, Technical Specification.

[11] GlobalPlatform Card Technology, "Secure Channel Protocol 03, Card Specification v 2.2 - Amendment D, Version 1.0," http://www.globalplatform.org/specificationscard.asp, April 2009, last access: November 30 2014.

[12] "Secure Element Evaluation Kit for the Android platform," https://code.google.com/p/seek-for-android/, last access: November 30 2014.

[13] Gemalto and Deutsche Telekom, "Secure Element API, Unofficial Draft," http://opoto.github.io/secure-element/, October 2014, last access: November 30 2014.

[14] G. Madlmayr, J. Langer, and J. Scharinger, "Managing an NFC Ecosystem," in *Proceedings of the 2008 7th International Conference on Mobile Business*, ser. ICMB '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 95–101.

[15] Giesecke&Devrient, "The OTA Platform in the World of LTE," White paper, January 2011.

[16] GlobalPlatform, "GlobalPlatform Card, Remote Application Management over HTTP, Card Specification v2.2 - Amendment B, Version 1.1.1," http://www.globalplatform.org/specificationscard.asp, March 2012, last access: November 30 2014.

[17] M. Badra and P. Urien, "Toward SSL Integration in SIM SmartCards," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE Atlanta*, vol. 2, March 2004, pp. 889–893.

[18] G. Madlmayr, J. Langer, C. Kantner, J. Scharinger, and I. Schaumüller-Bichl, "Risk Analysis of Over-the-Air Transactions in an NFC Ecosystem," in *Proceedings of the 2009 First International Workshop on Near Field Communication*, ser. NFC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 87–92.