International Journal of Multimedia and Ubiquitous Engineering
Vol. 10, No. 3 (2015), pp. 147-160
http://dx.doi.org/10.14257/ijmue.2015.10.3.15

A Framework for Secure NFC Applications

Jianchao Luo and Zhijie Qiu

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China
luojc@uestc.edu.cn, qzhijie@uestc.edu.cn

Abstract

Near Field Communication (NFC) technology provides a more natural and intuitive way for users to interact with intelligent environments within close proximity. It can be used in many scenarios including ticketing, advertising, controlled access, etc. However, NFC applications, especially those installed to the Secure Element (SE) for secure usage, have not yet been deployed widely enough in commercial environment. One reason is that the market is emerging and NFC-enabled smartphones are still very limited. Another reason is the life-cycle management of NFC applications on the SE is very complicated for application providers since the NFC business ecosystems have not yet taken off. In this paper, in order to establish a secure operating environment for third-party NFC applications, we propose a trusted NFC application management framework, which provides secure application provisioning and life-cycle management services. With the framework, NFC smartphone users can find and download interested NFC applications easily, and application providers can deploy NFC applications on users' SEs in a simple but secure way.

Keywords: *Near Field Communication, NFC Applications, Trusted Service Manager, Secure Element, Application Providers*

1. Introduction

Near Field Communication (NFC) [1] is an emerging technology that provides a more natural and intuitive way for users to interact with intelligent environments [2] within close proximity. As a subset of RFID technology, NFC is compatible with the existing contactless infrastructure and enables a simple and safe two-way interaction between electronic devices.

One advantage of this technology is its simplicity, as NFC is characterized by a natural and intuitive "just touch" movement. It just needs the NFC devices to be brought close to one another. There is no need to line up camera with barcode or input password.

Meanwhile, NFC has become a better choice for secure communication than other wireless connectivity protocols like Bluetooth, Wi-Fi, *etc.*, since NFC devices must be in close proximity to each other, typically 4 centimeters or less, where data communication is hard to be monitored by an unauthorized party. In addition, NFC can also be used to establish a Bluetooth or Wi-Fi connection in a simple and fast way for higher data transfer speed.

NFC is being included in more and more devices, particularly smartphones. The adoption of NFC into mobile phones has opened the door to various NFC applications including mobile payments, ticketing, advertising, controlled access, *etc*.

There are three main ways to use NFC:

- Reader/writer mode: the NFC device is active and reads or writes information from or to a passive NFC tag, such as an NFC tag embedded in

ISSN: 1975-0080 IJMUE

a smart poster.

- Peer-to-peer mode: two NFC devices communicate with each other to exchange information such as virtual business cards or digital photos.
- Card emulation mode: the NFC device behaves exactly like a contactless card, which can communicate with contactless reader for example to make a payment by touching a payment terminal.

Reader/writer mode and peer-to-peer mode are referred to as NFC open mode since they do not provide secure means of communication, whereas card emulation mode is a secure mode that supports secured NFC applications. There are two kinds of card emulation: Host-based Card Emulation (HCE) [3] and SE-based card emulation. HCE is a purely software-based solution that is simpler but less secure than SE-based card emulation, as it lacks secure storage and trusted execution environment. Hence the usage of a SE is recommended for turning NFC applications more secure. The SE is a tamper-resistant one chip secure microcontroller capable of securely hosting applications and their confidential and cryptographic data [4]. NFC applications like mobile payments and transit passes require to be installed to the SE, as they need to securely exchange and store sensitive data. SE can be available in multiple form factors like Universal Integrated Circuit Card (UICC), embedded SE, micro SD, *etc* [5]. Therefore, there are different business models to support the diversified choices.

However, NFC applications, especially those installed to SEs for secure usage, have not yet been deployed widely enough in commercial environment. One reason is that the market is emerging and NFC-enabled smartphones commercially available are currently very limited, although their shipments grow quickly. Another reason is the life-cycle management of a secure NFC application stored in the SE is very complicated for the Application Provider (AP) since the NFC business ecosystems have not yet taken off. As NFC technology can eliminate the need to carry bus passes, loyalty cards, house keys and event tickets, consumers will find out how much easier NFC can make their lives and demand more and more NFC applications. The NFC ecosystem, which consists of the entities collaborating to provide an overall NFC solution, needs to be in place to accommodate demands from NFC APs and NFC smartphone users.

Although there are some studies [6-8] on analyzing and defining the requirements of an NFC ecosystem, its architecture design and technical implementation are not explained in detail. In this paper, we present a secure and trusted NFC application management framework as well as its implementation, which manages NFC applications' life-cycle and provides secure data exchange and storage services. With the framework, NFC smartphone users can not only find and download interested NFC applications easily, but also view the card content directly on the smartphone, while the APs do not have to issue various smart cards, like membership cards, to users and can deploy NFC applications to users' SEs and alter the data stored in the SEs in a simple but secure way.

The rest of the paper is structured as follows. Section 2 reviews the related works. Section 3 describes the requirements of the proposed framework. In Section 4, we propose the framework's architecture and describe the main components of the framework in detail. Section 5 evaluates the approach via uploading, downloading and installing the NFC applications based on the framework. In Section 6, we conclude the paper.

2. Related Works

As NFC simplifies the way of interaction between devices and guarantees a fast and secure information exchange, many NFC applications have been developed in the recent years.

Borrego-Jaraba *et al.*, [9] propose an NFC-based contactless location and surfing system oriented to help the user to find the location of interest points within the city and navigate through them. Fernández *et al.*, [10] present an attendance control system based on NFC technology. Students can sign-in correctly by bringing their NFC smartphones close to an NFC tag in their classroom, following the "Tap & Go" mode. Hardy *et al.*, [11] present MyState, a novel application that enables users to create personalized physical interfaces that can be used to share information with the social community through quick, explicit touch interactions whilst providing complete control over when this information is shared. All of these NFC applications use the reader/writer mode.

In peer-to-peer mode, two NFC devices can establish a network connection to exchange data. Monteiro *et al.*, [12] propose a peer-to-peer based application that demonstrates the usage of NFC and Bluetooth technologies for money transaction between mobile devices. All the necessary data exchanged between both devices is sent through Bluetooth previously enabled by the NFC touch. In [13], the WingBonus application provides users with an easy way to acquire, store, manage and use vouchers. By using the peer-to-peer mode, the NFC-enabled device sends the information of the voucher to the NFC reader for redemption and users can also exchange vouchers between them. However, the application is not secure enough since the confidential information of the voucher is not stored in the SE.

Card emulation mode is useful for payment and ticketing applications, as it gives NFC devices smart card capability. GlobalPlatform has provided a card specification [14] that describes secure multi-application card content management functionality. In [15], Widmann *et al.*, propose an NFC ticketing system, which offers features for accessing the SE to view and store electronic tickets.

However, as more and more card emulation applications from different APs are to be developed and deployed on the same SE, a trusted NFC application management framework is needed for providing secure deployment and management services. Although there are some studies [6-8] on analyzing and defining the requirements of an NFC ecosystem, its architecture design and implementation are not explained in detail.

In this paper, in order to establish a secure operating environment for third-party NFC applications, a trusted NFC application management framework is proposed, which manages NFC applications' life-cycle and provides secure data exchange and storage services. Especially, the framework opens card emulation applications to the third-party developers who can deploy them to users' SEs in a simple but secure way.

3. System Description

An NFC Ecosystem is the collection of business entities that collaborate to provide NFC services. The main roles involved in the proposed framework include Trusted Service Manager (TSM), User, AP, Certification Authority (CA) and SE Issuer. In this section, we describe the system requirements by presenting user and AP's typical interaction scenarios with the system.

The main goal of the system is to build a secure and trusted environment for third-party NFC applications. In application submission phase, it provides APs with services including AP registration, application check, application publishing, *etc*.

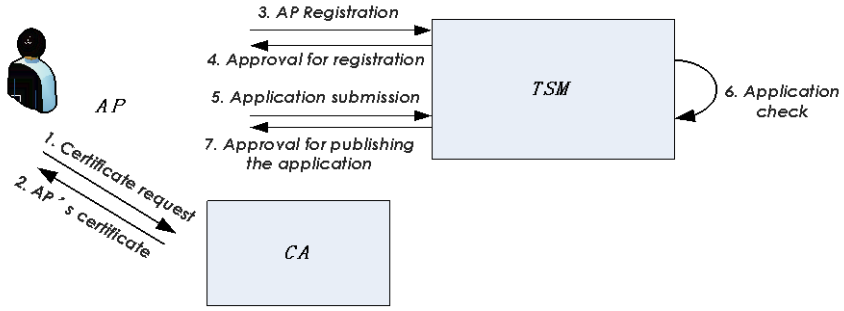

Figure 1. The Typical Interaction Process in Application Submission Phase

Figure 1 shows the typical interaction between AP and the system in application submission phase:

1. Requesting a digital certificate: In order to realize identity authentication, AP generates an RSA key pair and sends the public key to CA for requesting its digital certificate if it does not have one. Likewise, TSM also needs to request its digital certificate from CA.
2. Obtaining a digital certificate from CA: CA issues a newly generated digital certificate to AP.
3. AP registration: Before submitting an NFC application to the system, AP has to send its company or personal information to the system for check and record.
4. Approval for AP registration: Once AP registration is approved, the system will send a unique developer ID to AP for subsequent services like application submission.
5. Submitting an NFC application: AP logs into the system by using the developer ID and submitting an NFC application.
6. Application check: The system reviews the submitted application to ensure it is valid.
7. Approval for publishing the application: The valid application is published into the application store of the system for user's browsing and downloading. And AP will receive the notification of the publishing status.

In application installation and service phase, as the neutral and trusted party that all entities of the ecosystem can trust, the system provides users and APs with OTA application provisioning and lifecycle management services including application downloading, installation, personalization, key generation, *etc*.

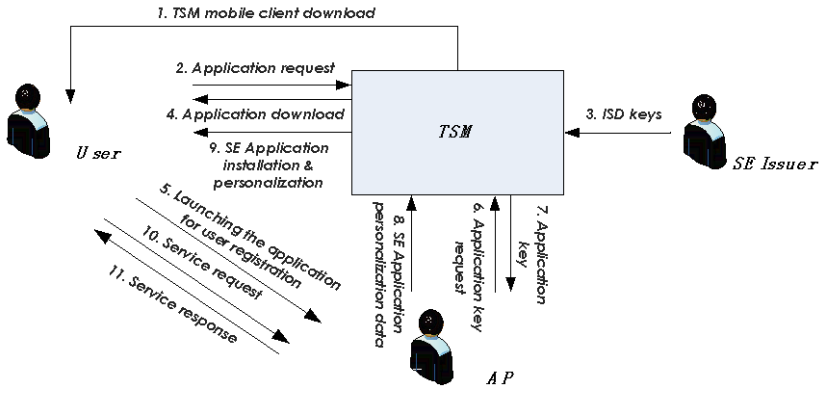

Figure 2. The Typical Interaction Process in Application Installation and Service Phase

Figure 2 shows the user and AP's typical interaction with the system in application installation and service phase:

1. Downloading the TSM mobile client: As the mobile agent of TSM, the TSM mobile client has to be installed on the NFC smartphone in advance in order to communicate with the TSM server securely.
2. Requesting an NFC application: As the TSM mobile client has already been installed at step 1, NFC smartphone users can browse and find the interested application before requesting to download it.
3. Obtaining Issuer Security Domain (ISD) keys: In order to control and manage SEs and SE applications, TSM must obtain the ISD keys from the SE Issuer. This step is needed only once for one SE. The ISD key set contains three keys:
   - Secure Channel Encryption Key: used for mutual authentication between the card and the off-card entity.
   - Secure Channel Message Authentication Code (MAC) Key: used for integrity authentication of the data transmitted over the secure channel.
   - Data Encryption Key: used for ensuring confidentiality of the transmitted sensitive data.
4. Downloading and installing the NFC application: The TSM mobile client downloads the NFC application from the TSM server, after which the user can install it. The secure NFC application requires to store sensitive data on the SE, however, the SE application code and personalization data will be loaded in a secure way subsequently.
5. Launching the application for user registration: For NFC applications deployed on the SE, the AP needs to acquire the user's information through the registration process in order to generate the personalization data specifically for him/her.
6. Requesting application keys: The AP sends a request to the TSM for application keys. For secure communication between SE application and off-card entity like an NFC reader which connects to the application backend system, a secure channel session has to be established using the application keys. The AP provides the TSM with key generation parameter such as user ID for generating application key set, which also contains three keys (Secure Channel Encryption Key, Secure Channel MAC Key and Data Encryption Key) for entity authentication, data integrity authentication and data confidentiality protection respectively.
7. Obtaining application keys: The TSM generates the application keys for the current user, encrypts them using TSM's RSA private key and then sends the encrypted application keys to the AP. The AP obtains the application keys after decryption using TSM's RSA public key.
8. Transferring the SE application's personalization data to TSM: Before performing personalization of the SE application, application's personalization data should be encrypted using the AP's private key and transferred to the TSM by the AP, after which the TSM can obtain the personalization data through decryption using the AP's public key.
9. Installing and personalizing the SE application: Before installing the SE application onto the SE, a secure channel session should be established for secure data transmission. By using the AP's associated Security Domain services, the TSM loads and installs the SE application, as well as performs the application's personalization with the personalization data received at step 8.
10. Requesting service: As an NFC device can operate in three different modes, the user can request service from the intelligent environment by touching an NFC tag, an NFC reader or an NFC smartphone. For example, while

operating in card emulation mode, the NFC handset sends the application data, like coupon, to the NFC reader which connects to the AP's backend system for validation and redemption. To protect data confidentiality, The system uses the application-specific key obtained at step 7 for encrypting the coupon's sensitive information.

11. Receiving service response: The NFC smartphone receives the service response. To take the example described at step 10, after the AP finishes validating the coupon information, it will send the redemption result to the user.

As described above, the propose framework clearly defines the primary roles in the NFC ecosystem, simplifies the deployment of NFC applications for APs and provides a secure and convenient way for users to obtain NFC applications.

4. System Architecture

The system is TSM-centric, which provides secure and trusted end-to-end framework for deployment and management of NFC applications.

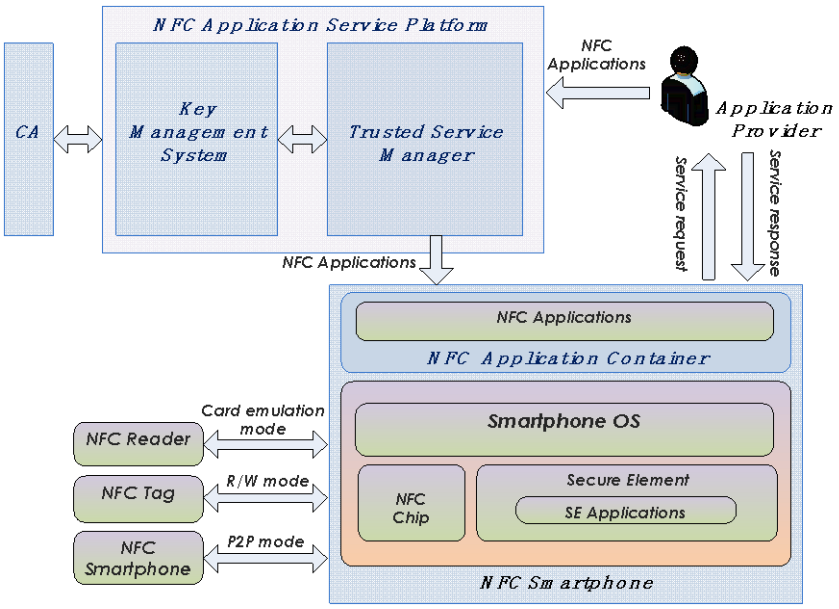

Figure 3. Architecture of the Proposed Framework

The framework is composed of two parts as shown in Figure 3:

- NFC Application Container: It is installed on the NFC smartphone in advance and is mainly responsible for showing the available NFC applications and managing NFC applications' life-cycle. As the TSM client, the NFC Application Container is delegated to establish secure channel sessions between the TSM and the SE and perform application loading, installation and personalization operations by sending APDU (Application Protocol Data Units) commands to the SE (APDUs are packets of data that are exchanged between the card and the off-card applications).
- NFC Application Service Platform: It is deployed on the remote server and provides secure and trusted services including AP management, key management, OTA provisioning, and application lifecycle management.

4.1. NFC Application Container

Before downloading and installing an NFC application, the user has to install the NFC Application Container on his/her NFC smartphone. As the mobile agent of the TSM, it facilitates loading and managing the third-party NFC enabled applications in a secure way, removing the need for the APs to publish and manage multiple versions of the same application all by themselves, and providing friendly user interface for the users to search and install NFC applications. Figure 4 shows the interaction between the user and the NFC Application Container.

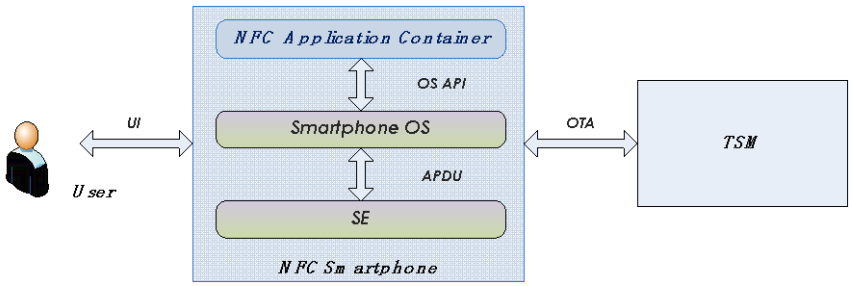

Figure 4. Interaction between the User and the NFC Application Container

Being an NFC application portal, the NFC Application Container provides a central place for users to discover and download trusted NFC applications securely. The APs can take it as an effective channel for promoting their NFC applications.

We can classify the NFC applications into two categories: open NFC applications and secure NFC applications. The open NFC applications are those that do not need to store sensitive information on the SE, like the Smart Poster application which reads the URL of the poster from the NFC tag and displays its detailed information on user's mobile phone. However, the secure NFC applications are those that require to save sensitive information onto the SE, like the Ticketing application, which operates in card emulation mode and stores the ticket voucher onto the SE in order to prevent its being read in an unauthorized manner, since the SE can provide secure storage and trusted execution environment. The proposed framework supports managing both open NFC applications and secure NFC applications.

The installation process of the open NFC applications is simple as they are actually mobile applications whereas that of the secure NFC applications is more complicated as they usually consists of two parts: mobile application and SE applet. The SE applet is a Java Card application running on the SE and mainly responsible for controlling access to the sensitive information stored in the SE. However, since the SE does not provide any graphic user interface, a mobile application running on the mobile device is needed to provide effective graphical interaction for the user to communicate with the SE applet. For the secure NFC application, its SE applet as well as personalization data have to be loaded and installed onto the SE in a secure way, after its mobile application has been installed on the mobile device. However, thanks to the NFC Application Container, the complicated installation process of a SE applet is transparent to the user.

The key functionality of the NFC Application Container is to manage the NFC applications' life-cycle in a secure and efficient way via cooperating closely with the NFC Application Service Platform. It can manage multiple NFC mobile applications as well as multiple SE applets. The application management functionality includes installation, personalization, deletion, lock, unlock, *etc*.

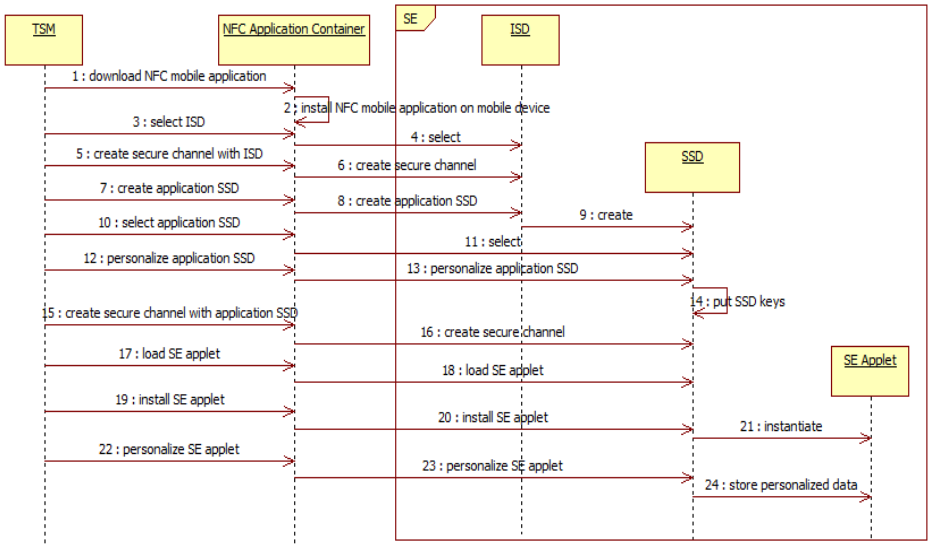

Figure 5. The Installation and Personalization Process of a Secure NFC Application

As shown in Figure 5, the NFC Application Container acts as a bridge between the TSM server and the SE (The open NFC application are not installed to the SE as described in Section 4.1.1). It receives APDU commands from the TSM and transfer them to the SE for processing. Figure 5 demonstrates the installation and personalization process of a secure NFC application in detail:

1. Downloading an NFC mobile application from TSM: The user downloads an NFC mobile application from the TSM's application store after finding it from the NFC Application Container. The NFC mobile application can provide graphical user interface for content management of the SE applets. However, download of a SE applet is performed separately in a more secure manner.
2. Installing the mobile application on the mobile device: The user accepts the application business agreement and installs the mobile application on the mobile device via the NFC Application Container.
3. Requesting to select the ISD: After finishing installing the mobile application of the secure NFC application, its SE applet should be installed subsequently. The Security Domain is mainly responsible for loading and installing the SE applet. The Issuer Security Domain (ISD) is the primary and mandatory on-card representative of the SE Issuer while the Supplementary Security Domain (SSD) is the additional and optional on-card representative of the AP. In the proposed framework, each AP requires to have its own SSD for managing its applications in order to securely isolate its own SSD and applications from those of other APs. Because the SSD is created by the ISD, the TSM should communicate with the ISD by selecting it.
4. Selecting the ISD: Upon receiving the "SELECT" command from the TSM, the NFC Application Container sends it to the SE and the ISD is then selected.
5. Requesting to create a secure channel with the ISD: For secure data exchange, a secure channel session should be established through mutual authentication between the card and the off-card entity. The TSM initiates the request to create a secure channel with the ISD.
6. Creating a secure channel with the ISD: In the secure channel initiation phase, the NFC Application Container sends the "INITIALIZE UPDATE" command to the ISD to determine whether it is communicating with an authenticated card entity, and then it sends the "EXTERNAL AUTHENTICATION"

command to enable the ISD to determine whether it is communicating with an authenticated off-card entity. During the mutual authentication process, both the card and the off-card entity send the newly generated challenge data encrypted by using the secure channel encryption key of the ISD to the other party, and the other party verifies the cryptogram.

7. Requesting to create a SSD for the AP: To isolate applications of the AP from those of other APs, the TSM should create a SSD for each AP.
8. Transferring the SSD creation command to the ISD: Upon receiving the SSD creation command from the TSM, the NFC Application Container transfers it to the ISD.
9. Creating a SSD for the AP: The ISD creates the AP's associated SSD.
10. Requesting to select the AP's associated SSD: The AP's associated SSD is a privileged application that holds cryptographic keys which can be used to support secure channel establishment operations and to authorize card content management functions. Before performing installation and personalization of a SE applet, the AP's associated SSD has to be selected and personalized.
11. Selecting the AP's associated SSD: Upon receiving the "SELECT" command from the TSM, the NFC Application Container transfers it to the SE and the AP's associated SSD is then selected.
12. Requesting to personalize the AP's associated SSD: During the personalization process, the AP's associated SSD loads cryptographic keys which can be used to support secure channel establishment operations and application management functions.
13. Transferring the personalization command to the SSD: The NFC Application Container transfers the "PUT KEY" command to the AP's associated SSD. To protect data confidentiality, the cryptographic keys to be loaded to the SSD are encrypted using the data encryption key of the ISD before being sent to the SSD.
14. Personalizing the AP's associated SSD: The SSD stores the key information supplied in the "PUT KEY" command.
15. Requesting to create a secure channel with the AP's associated SSD: For secure data exchange, a secure channel session should be established through mutual authentication between the TSM and the SSD. The TSM initiates the request to create a secure channel with the SSD. The secure channel encryption key used for off-card entity authentication has already been loaded to the SSD at step 14.
16. Creating a secure channel with the AP's associated SSD: The initiation of a secure channel session is the same as that described at step 6.
17. Requesting to load a SE applet: The TSM initiates the request to load a SE applet onto the SE.
18. Loading a SE applet: The AP's associated SSD is responsible for loading the SE applet. The NFC Application Container sends the "INSTALL [for load]" command to the SSD for loading. The data field of the command details the requirements regarding the load file. Multiple "LOAD" commands are then used to transport the applet's load file in blocks according to the size of the file and the communication buffer size of the SE.
19. Requesting to install the SE applet: The TSM initiates the request to install the SE applet loaded at step 18.
20. Installing the SE applet: The NFC Application Container sends the "INSTALL [for install and make selectable]" command to the SSD for installing the SE applet just loaded.
21. Instantiating the SE applet: The installation process includes the creation of applet instance and applet registration in the GlobalPlatform registry. The

GlobalPlatform registry stores applet management information including its Application Identifier (AID), associated Security Domain, etc. After the SE applet is made selectable, the off-card entity can select it for follow-up communication by sending the "SELECT" command in which the AID is specified.

22. Requesting to personalize the SE applet: After the installation, the SE applet requires to load its personalization data which may include its own keys and application-specific data. The application's personalization data is generated by the AP and usually user-related. For example, the personalization data of a payment application may contain information such as credit/debit card number, user name, card expiry date, etc. Likewise, the secure communication service of the applet's associated SSD is used to manage the secure loading of the personalization data.
23. Personalizing the SE applet: The applet's associated SSD is responsible for personalizing the SE applet. The NFC Application Container sends the "INSTALL [for personalization]" command and subsequent "STORE DATA" commands to the SSD for performing the personalization of the applet.
24. Storing personalization data to the SE applet: The SSD stores the personalization data to the SE applet.

However, steps through 3 - 9 and steps from 12 to 14 are not required if the SE applet's associated SSD has already been created and personalized before the installation and personalization of a secure NFC application.

4.2. NFC Application Service Platform

The NFC Application Service Platform is TSM-centric and provides secure and trusted services which mainly include AP management, key management, application OTA provisioning and lifecycle management.

As the core module of the platform, the TSM is a neutral and trusted party that all entities of the NFC ecosystem can trust. It consists of three modules (See Figure 6):

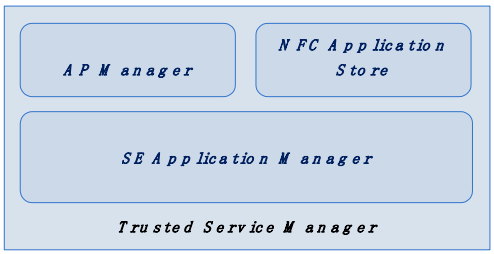

Figure 6. The Architecture of the TSM

- AP Manager

  Every AP has to apply for a developer ID before it can upload an NFC application to the system. The AP Manager provides AP registration and management functionality. The AP should send its company or personal information to the system for check and record. Once AP registration is approved, the AP will obtain a unique developer ID for logging into the system.
- NFC Application Store

  There are so many APs and NFC applications. To simplify the discovery and installation of NFC applications, the NFC Application Store provides a central place for housing them. It is in charge of receiving NFC applications from the APs and publishing them for browsing and downloading. To ensure that all the

NFC applications are trustworthy, it will check the security and usability of the uploaded applications before publishing them.

- SE Application Manager

  The SE Application Manager manages multiple SE applets as well as their associated SSDs on the SE through cooperating with the NFC Application Container. As shown in Figure 7, in order to isolate applications of one AP from those of other APs, the SE Application Manager needs to create different SSDs for different APs since different APs deploy applications on the same SE.

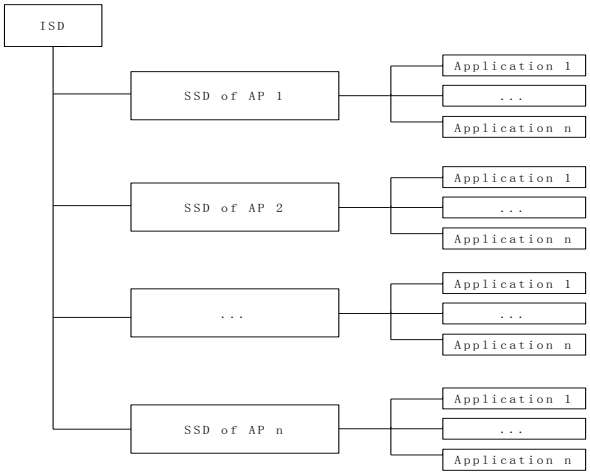

Figure 7. Different SSDs are Created for Different APs to Deploy their Applications on the SE

The SE Application Manager is required to obtain the ISD keys from the SE Issuer in order to create secure communication channel with the ISD, before it can create a SSD for the AP. The SSD can provide the application with cryptographic services to ensure confidentiality and integrity during personalization and runtime, therefore it has to hold cryptographic keys. The SE Application Manager puts keys into the SSD during the personalization process.

The SE application life-cycle management includes installation, personalization, deletion, lock, unlock, *etc*. To perform these application management operations, a secure channel session should also be established between the SE Application Manager and the SSD.

The Key Management System is mainly responsible for generation and storage of security domain keys as well as application keys.

To implement secure communication between the AP's backend system and the NFC terminal application during the NFC service usage phase, the AP should apply for application keys generated by the Key Management System. In the proposed framework, the application's master key set contains three 16-byte DES keys: secure channel encryption key, secure channel MAC key and data encryption key, which are used for mutual entity authentication, data integrity protection and confidentiality protection respectively. However, application's sub key set has to be dynamically generated for each user using application's master key set and each user's personal information, like a user's membership card ID, because different users should hold different application keys. The application's sub key set also contains three 16-byte DES keys.

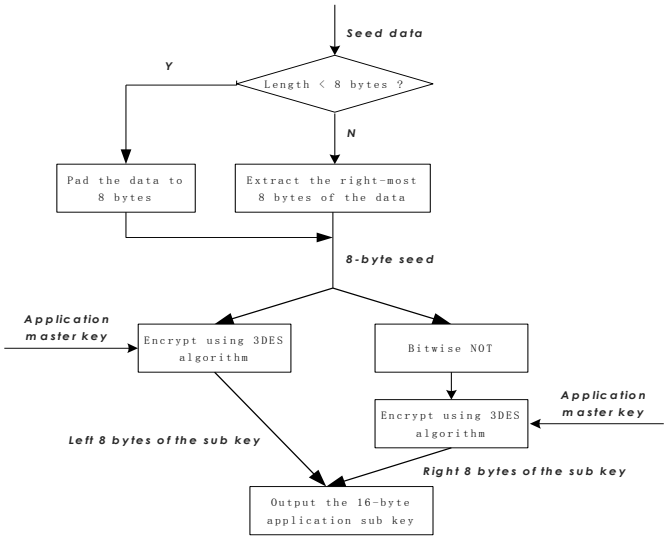

Figure 8. The Process of Generating Application's Sub Key

As shown in Figure 8, the Key Management System generates application's 16-byte sub key using the triple DES algorithm. The input seed data, *i.e.*, user's ID, is limited to 8 bytes and the application's master key is used as the cryptographic key for the computation.

As the TSM should hold the key set of the AP's associated SSD for SE application management, the Key Management System is also required to generate the SSD keys based on the same key generation method described above. The AP's ID is used as the input seed data while the ISD keys are used as cryptographic keys.

5. Evaluation

To validate the approach, two NFC applications are implemented and uploaded to the TSM for user download. As the two applications contain secret voucher information that needs storing in the SE, either of them includes a mobile application as well as a SE applet.

For the evaluation, the NFC Application Container should be installed on the NFC smartphone in advance. Figure 9 shows the UIs of downloading and installing the NFC applications via the NFC Application Container on the Android device.

(a) (b) (c)

Figure 9. Downloading and Installing NFC Applications via the NFC Application Container (a) The NFC Application that the User has Installed on the Phone. (b) The Application Store Where the User can Download New Applications. (c) The User Downloads and Installs the Movie Ticket Application

The NFC device where we run the NFC Application Container and the applications is a Samsung Galaxy S4 smartphone equipped with a micro SD card. The SD card has built-in SE and NFC antenna and is compatible with GlobalPlatform card specifications.

Through uploading, downloading and installing the NFC applications based on the framework, we can conclude that the proposed framework is easy to use for both end users and APs.

6. Conclusions

NFC is user-friendly and simple; however, the NFC business ecosystems have not yet taken off. Although the requirements of an NFC ecosystem have been widely analyzed, its architecture design and technical implementation are not explained in detail.

In this paper, we propose a secure and trusted NFC application management framework, which provides secure deployment and management services for third-party NFC APs and enables users to obtain NFC applications in a secure and convenient way.

We clearly define the primary roles in the NFC ecosystem and describe the system requirements by presenting user and AP's typical interaction scenarios with the system. A TSM is implemented and taken as the central entity that all party can trust, as it builds a trusted environment to enable secure communication and reduces the complexity of NFC services' business model. APs are thus relieved of large application management burden.

Security of application management and usage is well guaranteed by the proposed system. As more and more secure NFC applications from different APs are developed and deployed on the same SE, we create different SSD for each AP in order to securely isolate their applications. To provide cryptographic services for application management, the SSDs are personalized with secret keys which are dynamically generated by the proposed framework. In the meantime, to implement secure communication between the SE application and off-card entity during the NFC service usage phase, we provide each SE application with unique application key set which are used for entity authentication, data integrity protection and confidentiality protection respectively.

By providing an application store, the central place to house the NFC applications, the framework enables the NFC smartphone users to find and download interested NFC applications easily as well as view the card content directly on the phone. As the security and usability of the uploaded applications are checked before being published, they can be trusted by the users.

The SE can be incorporated on the UICC, embedded in the handset, or added externally as a micro SD card and the proposed framework is compatible with all these SEs. However, UICC and embedded SE are issued and managed by the MNOs or handset manufacturers while micro SD card is a third option where APs can deploy their secure NFC applications through the neutral TSM without MNO or handset manufacturer's involvement. To validate the framework, two NFC applications are implemented and deployed on the micro SD card's SE.

We think that the propose framework can effectively accelerate the practical application of NFC and help to form a harmonious NFC ecosystem. However, since it is still in the experimental stage, our future work will mainly focus on studying its scalability to support a large number of NFC applications and users.

Acknowledgements

The authors wish to acknowledge the support of the Mobile Payment Platform Research project (ID: 2013-265) of Important Science & Technology Projects of Chengdu Science and Technology Bureau for funding the research work presented in this paper.

References

[1] http://nfc-forum.org/what-is-nfc/.
[2] M. V. Bueno-Delgado, P. Pavón-Marino, A. De-Gea-García, and A. Dolón-García, "The smart university experience: An NFC-based ubiquitous environment", Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, **(2012)**.
[3] http://developer.android.com/guide/topics/connectivity/nfc/hce.html.
[4] http://www.globalplatform.org/mediaguideSE.asp.
[5] M. Reveilhac and M. Pasquet, "Promising Secure Element Alternatives for NFC Technology", Proceedings of the 1st International Workshop on Near Field Communication, **(2009)**, pp. 75-80.
[6] G. Madlmayr, J. Langer, and J. Scharinger, "Managing an NFC ecosystem", In Proceedings of the 7th international conference on mobile business, **(2008)**, pp. 95-101.
[7] B. Benyo, A. Vilmos, G. Fordos, B. Sodor, and L. Kovacs, "The StoLPan view of the NFC ecosystem", In Proceedings of the conference on wireless telecommunications symposium, **(2009)**, pp. 1-5.
[8] http://67.222.41.204/wp-content/uploads/2013/12/Stolpan-White-Paper-08.pdf.
[9] F. Borrego-Jaraba, I. Luque Ruiz, and M. A. Gómez-Nieto, "A NFC-based pervasive solution for city touristic surfing", Personal and Ubiquitous Computing, vol. 15, **(2011)**, pp. 731-742.
[10] M. J. L. Fernández, J. G. Fernández, S. R. Aguilar, B. S. Selvi, and R. G. Crespo, "Control of attendance applied in higher education through mobile NFC technologies", Expert Systems with Applications, vol. 40, **(2013)**, pp. 4478-4489.
[11] R. Hardy, E. Rukzio, P. Holleis, and M. Wagner, "MyState: Sharing social and contextual information through touch interactions with tagged objects", Proceedings of the 13th International Conference on Human-Computer Interaction with Mobile Devices and Services, **(2011)**, pp. 475-484.
[12] D. M. Monteiro, J. J. P. C. Rodrigues and J. Lloret, "A secure NFC application for credit transfer among mobile phones", Proceedings of 2012 International Conference on Computer, Information and Telecommunication Systems, **(2012)**.
[13] F. Borrego-Jaraba, P. C. Garrido, G. C. García, I. L. Ruiz, and M. A. Gómez-Nieto, "A ubiquitous NFC solution for the development of tailored marketing strategies based on discount vouchers and loyalty cards", Sensors (Switzerland), vol. 13, **(2013)**, pp. 6334-6354.
[14] http://www.globalplatform.org/specificationscard.asp.
[15] R. Widmann, S. Grünberger, B. Stadlmann and J. Langer, "System integration of NFC ticketing into an existing public transport infrastructure", In Proceedings of the 4th International Workshop on Near Field Communication, pp. 13-18, **(2012)**.

Authors

Jianchao Luo, he is a Ph.D. candidate in Computer Science at University of Electronic Science and Technology of China. He has been a lecturer in School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include mobile computing, context-aware systems and e-commerce security.

Zhijie Qiu, he received his B.S. and M.S. degrees in Computer Science and Engineering from University of Electronic Science and Technology of China, in 2001 and 2004, respectively. He has been an associate professor in School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include wireless communication, e-commerce security and ubiquitous computing.

