NFC FEATURE BOX: AN OPEN, NFC ENABLER INDEPENDENT MOBILE PAYMENT AND IDENTIFICATION METHOD

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İSMAİL TÜRK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

JULY 2019

Approval of the thesis:

NFC FEATURE BOX: AN OPEN, NFC ENABLER INDEPENDENT MOBILE PAYMENT AND IDENTIFICATION METHOD

submitted by **İSMAİL TÜRK** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Prof. Dr. Nihan Kesim Çiçekli
Supervisor, **Computer Engineering, METU** _____

Assist. Prof. Dr. Pelin Angın
Co-Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Ahmet Coşar
Computer Engineering, Turkish Aeronautical Association Uni. _____

Prof. Dr. Nihan Kesim Çiçekli
Computer Engineering, METU _____

Prof. Dr. Halit Oğuztüzün
Computer Engineering, METU _____

Assist. Prof. Dr. Hande Alemdar
Computer Engineering, METU _____

Assoc. Prof. Dr. Tansel Dökeroğlu
Computer Engineering, TED University _____

Date: 30.07.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: İsmail Türk

Signature:

ABSTRACT

NFC FEATURE BOX: AN OPEN, NFC ENABLER INDEPENDENT MOBILE PAYMENT AND IDENTIFICATION METHOD

Türk, İsmail
Doctor of Philosophy, Computer Engineering
Supervisor: Prof. Dr. Nihan Kesim Çiçekli
Co-Supervisor: Assist. Prof. Dr. Pelin Angın

July 2019, 120 pages

The use of Mobile Devices for electronic payment has increased significantly in the last decade. Near Field Communication (NFC) mobile payment is gaining popularity and it is widely considered to be the technology that will turn smartphones into m-wallets. While a typical wallet contains identification, loyalty, public transport and credit cards, m-wallet solutions currently have well-defined standards for credit card enrollment and usage only. In this thesis, we explore and present the main reason for this limitation. Then, we introduce a new method for using the Secure Element (SE) inside the NFC Phones which allows standard enrollment and usage for proprietary payment and identification schemes. We also explain the practical use of our model by presenting well-known proprietary payment scenarios such as public transport payment. This study makes significant contributions to NFC technology by proposing a method for proprietary transaction flows that can be used in parallel with NFC credit card payment solutions. Furthermore, it provides an open protocol which can serve as an NFC payment and identification transaction execution standard.

Keywords: Near Field Communication, Mobile Transactions, Financial Transactions, Payment Protocols

ÖZ

NFC ÖZELLİK KUTUSU: BİR AÇIK, NFC SAĞLAYICI BAĞIMSIZ MOBİL ÖDEME VE KİMLİKLENDİRME YÖNTEMİ

Türk, İsmail
Doktora, Bilgisayar Mühendisliği
Tez Danışmanı: Prof. Dr. Nihan Kesim Çiçekli
Ortak Tez Danışmanı: Dr. Öğr. Üyesi Pelin Angın

Temmuz 2019, 120 sayfa

Mobil cihazların elektronik ödemeler için kullanımı geçtiğimiz on yılda önemli derecede artış gösterdi. Yakın Saha İletişimi (NFC) ile mobil ödeme giderek popülerlik kazanmakta ve kitleler tarafından akıllı telefonları mobil cüzdanlar haline dönüştüren teknoloji olarak adlandırılmaktadır. Bilindik bir cüzdan kimlik, sadakat, toplu taşıma ve kredi kartları barındırırken, günümüzdeki mobil cüzdan çözümleri sadece kredi kartı ekleme ve kullanımı için tanımlı standartlar sunmaktadır. Bu tez çalışmasında bizler bu kısıtın temelindeki sorunu araştırıp sunuyoruz. Ve ardından, NFC telefonlar içerisinde bulunan Güvenli Öğe (SE) kullanarak kişiye özel ödeme ve kimliklendirme sistemlerinde kullanılabilecek yeni bir yöntem tanıtıyoruz. Bunun yanı sıra önerdiğimiz yöntemin toplu taşıma ödemesi gibi sektörde yaygın olarak bilinen ödeme senaryoları üzerinde pratikteki kullanımını sunuyoruz. Bu çalışma NFC ile kredi kartı ödeme çözümleri ile eşzamanlı olarak kullanılabilecek kişiye özel işlem akışları tanımlası gereği NFC teknolojisine önemli katılarda bulunmaktadır. Buna ek olarak, NFC ödeme ve kimliklendirme işlemi gerçekleştirme standartı olarak kullanılabilecek bir açık protokol sunmaktadır.

Anahtar Kelimeler: Yakın Saha İletişimi, Mobil İşlemler, Finansal İşlemler, Ödeme Protokolleri

Dedicated to my sons to inspire them with science throughout their lives.

ACKNOWLEDGEMENTS

I sincerely thank to my advisors Prof. Dr. Nihan Kesim Çiçekli and Asist. Prof. Dr. Pelin Angın as this would not be possible without their great support.
I sincerely thank to Prof. Dr. Ahmet Coşar for his support and guidance throughout my Master and Doctorate programs.
I thank to TUBİTAK Bilim İnsanı Destekleme Daire Başkanlığı (BİDEB) 2211 Yurt İçi Doktora Burs Programı for supporting my PhD education.
I thank to examining committee members for providing their valuable feedbacks.

There is no violation of intellectual property rights of my past and current employers in this thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

AES	Advanced Encryption Standard
AID	Application Identifier of Chip Applet
APDU	Application Protocol Data Unit
API	Application Programming Interface
CDATA	Data field of APDU command
CI	Chain Index of Create Instance Command
CM	Card Manager
CLA	Class byte of the APDU command
$DEC_X(m)$	Decryption of m using key X
DES	Data Encryption Standard
EMV	Europay, Mastercard and Visa
$ENC_X(m)$	Encryption of m using key X
ETSI	European Telecommunications Standard Institute
FIM	Feature Instance Manager
GSM	Global System for Mobile Communications
GSMA	GSM Association
HCE	Host Card Emulation
INS	Instruction Byte of the APDU command
ISD	Issuer Security Domain
ISO	International Organization for Standardization
LCI	Last Chain Indicator of Create Instance Command
Lc	Length Byte of APDU command
Le	Expected Length Byte of APDU command
MAC	Message Authentication Code
MNO	Mobile Network Operator

NFC	Near Field Communication
NIST	National Institute of Standards and Technology
P1	Parameter-1 byte of APDU command
P2	Parameter-2 byte of APDU command
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PUB_X	Public Key of X
PRI_X	Private Key of X
RFID	Radio Frequency Identification
Rnd_X	Secure Random X
SE	Secure Element
SAM	Secure Access Module
SIM	Subscriber Identity Module
SSD	Supplementary Security Domain
SP	Service Provider
TSM	Trusted Service Manager
UICC	Universal Integrated Circuit Card

CHAPTER 1

INTRODUCTION

The earliest mobile handset devices were designed with two key capabilities: initiate and answer phone calls. Today, however, these devices have evolved, with smartphones serving as mobile devices which can easily perform many computing tasks in our daily life (Agar, 2013).

Near Field Communication (NFC) technology had been available on our mobile handsets for more than a decade. NFC is the technology that enables secure and contactless communication between mobile devices that are in physical proximity, within a few centimeters (Curran, 2012). Because of the unique features that are possible with NFC the number of NFC enabled phone models had increased tremendously over the years. The most remarkable advantage of NFC technology was its capability to turn a phone into a wallet because of the security it can offer (Andersson, 2011). Provided security comes from the Secure Element (SE) that an NFC phone has, which is similar to the chip that is available on credit cards. As a result, system designers preferred to use the SE to store secure credentials which will be used for payment and identification routines that the mobile phone will be used for (Tan, 2014).

The SE chip has multi-application capabilities; therefore, enabling it to host several applications at the same time. In this way, the mobile handset chip allows for numerous credit card application instances as well as proprietary applications such as loyalty cards, transport ticketing and others (Akram, 2010).

An SE requires secure content management that can be offered only by the SE owner because an SE is protected by Issuer Keys which are only known to the owner of the

device. In this way, only the owner of the SE has administrative capabilities to update chip content (Turk, 2015a).

Payment and Identification industry performs a transaction in two main flows; online and offline transactions. In an online transaction flow, user authenticity decision to perform the requested operation is given at the central systems of the Service Provider by utilizing online records and verification data of the user. However, in an offline transaction flow, user authenticity decision to perform the requested operation is given within the terminal without connecting to any online resource. This doesn't necessarily mean that offline transactions are not capable of or are not allowed to access online resources while they are idle. The main differentiation factor is the requirement of accessing online resources during the transaction execution.

Offline payment and identification approach is in the core of public transportation payments, loyalty systems, and gate/office access systems today. Performing these operations by mobile phone was one of the main advantages that NFC can offer (Aldershof, 2012; Dias, 2014), whereas still today there is no solution available on our NFC phones.

1.1. Problem Definition

The importance of digital payments projected nearly 30 years ago, and agnostic payment flows started to be designed to protect user privacy (Chaum, 1985). NFC Technology is seen as the ideal candidate to boost digital payment as it brings the convenience of paying with the mobile handset that we carry all the time (Choudhary, 2006). Once the NFC functionality became an available feature on mobile phones, payment industry specifications were updated to cover mobile payment schemes that incorporate NFC capabilities (EMV, 2008; EMVCo, 2007). Card specifications have also been updated to include the mobile residence of tamper-resistant chip Secure Element (SE) and the management of secure information stored in it by the manufacturer, but not directly by the phone user (GlobalPlatform, 2012). Consequently, the mobile transaction industry adopted specifications to include

payment schemes over NFC (Raina, 2017) and payment through NFC phones had been promoted by industry players with financial incentives as it is proven to accelerate NFC adoption (Zhao, 2019).

The preparation of a physical contactless card differs slightly from the preparation of an instance within a mobile handset. Physical contactless cards are produced in secure environments and the target user information is already known when they are being prepared (Morse, 2008). However, when the secure element of a mobile handset is manufactured, the final user is unknown; along with the applications it will contain. All the issuances are performed while the mobile handset is actively in use. Therefore, NFC issuance requires remote access to the SEs to handle any contactless application activity. As a result of this fact, NFC issuance is dependent on the NFC Enabler.

This method of execution was intentional, the Mobile Payment Industry Players sought to dominate mobile payment platforms and monopolize payment technology during its infancy (Staykova, 2015).

We believe that dependency on the NFC Enabler is the biggest problem in NFC technology preventing this technology from worldwide adoption. The following scenario is an example which is used for explaining our proposal.

"Business B has between 250-500 personnel who use contactless cards for secure access to the main building and their offices. The building also has facilities, a café, restaurant and vending machines in the main building all of which use a proprietary payment system. Since many employees own NFC enabled phones, the company decides to allow their employees to use their phones for building access and payments in the café and restaurant. As there are several brands of phones used by employees and some of the phones even have their SEs stored in the SIM card; the company cannot possibly sign deals with so many phone manufacturers and mobile phone network operators."

Although above scenario is a simple application of NFC technology, realization of such a system is very difficult because we need approval and operational involvement

of mobile phone manufacturers which are beyond the ability and financial power of a small business, this task achievable only by large companies/businesses which are attractive/profitable-enough to phone manufacturers.

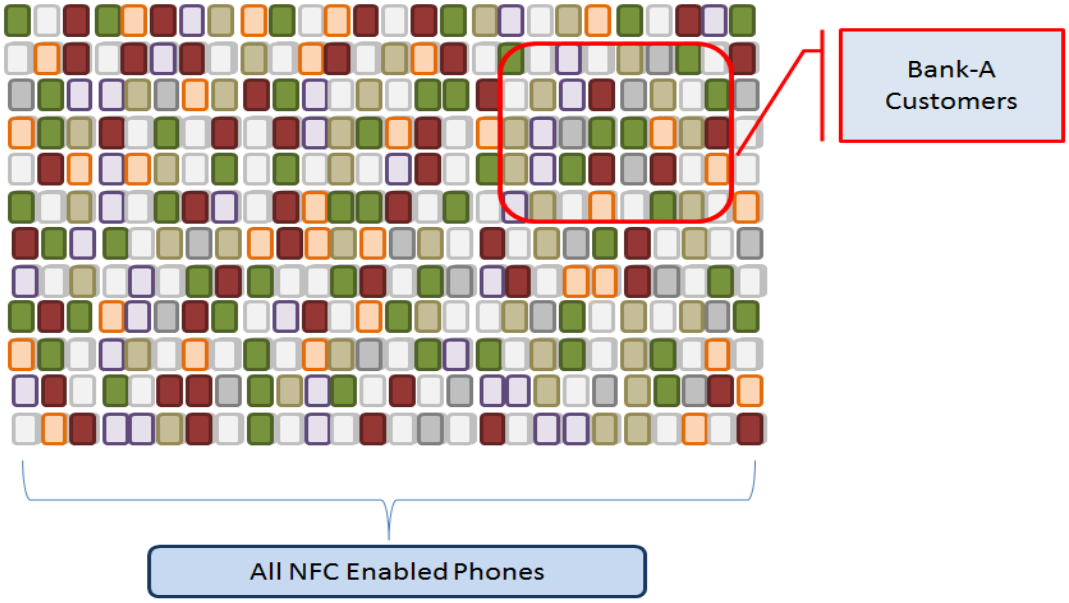

Figure 1.1. Different NFC enablers versus Service Provider's target

Figure 1.1 illustrates all available NFC phones in the market – each color denotes different NFC-enabler – that requires different service issuance schemes. Whereas, a Service Provider – illustrated as a bank in the figure – is only interested in a portion of those but very likely having all kinds of NFC-enablers. This results in, no matter the Service Provider customer base size is, the current state of the NFC usage activation is not without being forced to deal with all major enablers.

Finally, the major problem in the current state of the NFC issuance and usage infrastructure can be summarized as follows:

- There are too many NFC-Enablers,
- There is no interoperability in between,
- The entry barrier for small businesses is too high to afford,

- There are no identification standards

1.2. Main Goal and Contributions

It is an obvious fact that the NFC ecosystem will not succeed without having an enabler independent NFC issuance and usage mechanism. Thus, the credit card payment industry updated the standards to adapt credit card application issuance to the NFC phones. However, the lack of a standard for proprietary payment and identification application issuance is still a blocker.

Our main goal is to solve the dependency problem in NFC payment and identification systems and to demonstrate it can easily be implemented by any payment industry player, potentially offered as an additional service to their existing customers.

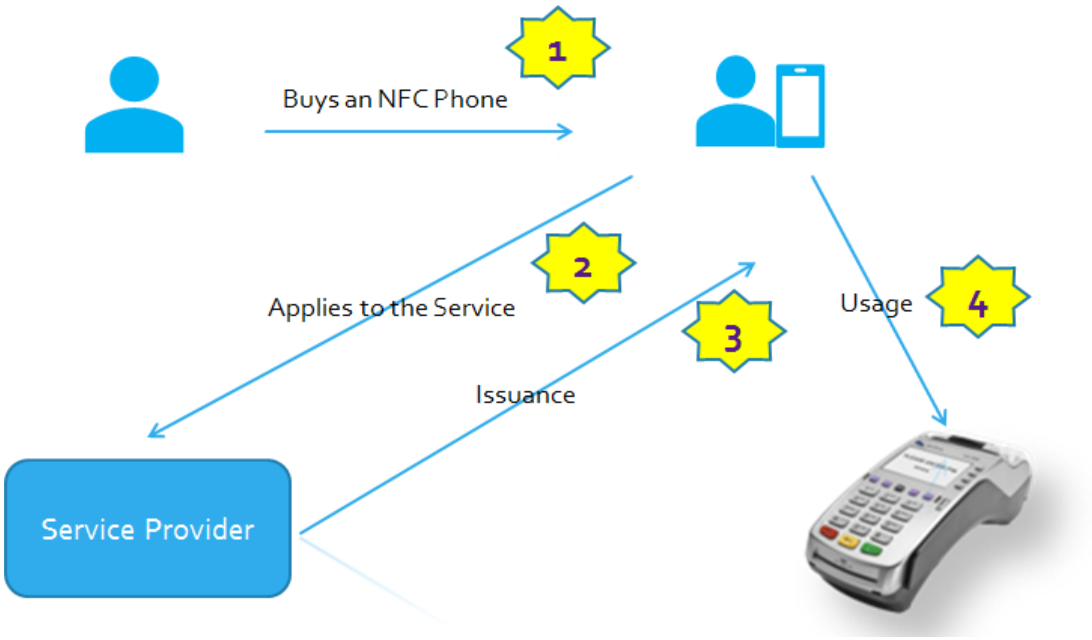

Figure 1.2. Targeted NFC issuance and usage flow

Figure 1.2 depicts the main contribution of this thesis. As shown in the flow, our goal is to allow a user to own any NFC phone, apply to any service, get desired service issued by the Service Provider without needing NFC Enabler involvement and

eventually, allow Service Provider to prepare issued service for making transaction without requiring NFC Enabler's secure transaction channel to the phones.

We propose that NFC Phones install an NFC Feature Box application in their SEs that communicates with the card system terminals to perform card-based transactions. This thesis explains the details of our proposed method and its implementation protocol. We show how NFC Feature Box application can easily be implemented by any card-based system owner and offered to all their customers. Thus, we can replace all the contactless cards we use for many different tasks with a single mobile phone.

This thesis contributes to NFC mobile transactions field by committing to achieve the following objectives:

Objective 1: Identify major problems in the NFC ecosystem that is blocking mass adaption of the technology and limiting the usage to a few payment schemes only.

Objective 2: Develop a User-Centric solution for the NFC ecosystem that users can register and start using NFC services at their will, without requiring third-party involvement.

Objective 3: Design and develop open NFC feature issuance and activation protocols that Service Providers can implement and join the NFC ecosystem on their own.

Objective 4: Besides solving identified problems, contribute to enhancements of the NFC ecosystem by introducing features and flexibilities that are improving NFC transaction user experience.

Objective 5: Discuss and evaluate security and performance aspects of proposed solutions to ensure goals are achieved with a secure and efficient protocol.

1.3. Structure of the Thesis

The remainder of this thesis is organized as follows.

In Chapter 2 we provide information about wireless communication principles behind NFC technology. Then, we explain the main factors that are allowing secure

transactions over NFC communication. We identify the entities playing an active role in the management of secure components. NFC technology had been tailored to custom scenarios by involving dedicated protocols that are explained in this section. Physical card and NFC issuance dynamics and differentiators are explained that helps to clarify the enabler dependency problem in the field. Finally, we list studies that had been contributing to solving the NFC Enabler dependency problem from different aspects.

In Chapter 3 we explain our studies that we had conducted to solve the NFC enabler dependency problem in online mobile transactions. Different approaches to perform an online NFC transaction and corresponding challenges are explained in this chapter. Our solutions in this field are explained from security and performance perspective.

In Chapter 4 we provide a detailed explanation of our proposed model for offline mobile transactions together with its high-level design, technical applicability to the field, protocol details and inter-party secure handshake to create secure data transfer channel, remote secure issuance of services, activation and management lifecycle of the instances and details of transaction protocol. All the command structures and transaction command/response formats are provided in this section.

In Chapter 5 we evaluate our protocol from the security and performance perspective. We list possible attack scenarios in the field and explain how NFC Feature Box is protected against those. NFC Feature Box is proven both theoretically and practically. Details about our experimental setup to show the practical applicability of our solution in the field are explained in this section.

Chapter 6 presents the results of our studies and gives details about our achievements. We explain how NFC Feature Box and our other contributions to NFC mobile transaction field solves the NFC Enabler dependency problem. Then, we investigate how further can NFC Feature Box be improved, and we list possible future studies that can use NFC Feature Box as the core solution.

CHAPTER 2

BACKGROUND

2.1. Near Field Communication

Near Field Communication (NFC) is a wireless communication technology that lets data transmission between two NFC media in a contactless way (Turk, 2015b). NFC technology is derived from RFID technology, therefore, we will explain RFID principles before explaining the details of NFC technology.

2.1.1. RFID

Radio Frequency Identification (RFID) is a wireless communication technology that lets data transmission between an RFID Reader and an RFID tag. In general, RFID tags have no power source and harvest the power it needs to operate from the RF field generated by the RFID reader (Kitsos, 2016). After the invention of RFID technology, there had been other RFID tags, called active RFID tags, which are connected to a power source, such as a battery. Main purpose of having an active RFID is to increase the range of data transmission distance. Besides active tags, common usage of RFID technology is to generate a field in front of the reader that passive RFID tags can harvest the power when they are in the range and use it to operate. A typical RFID system is depicted in Figure 2.1.

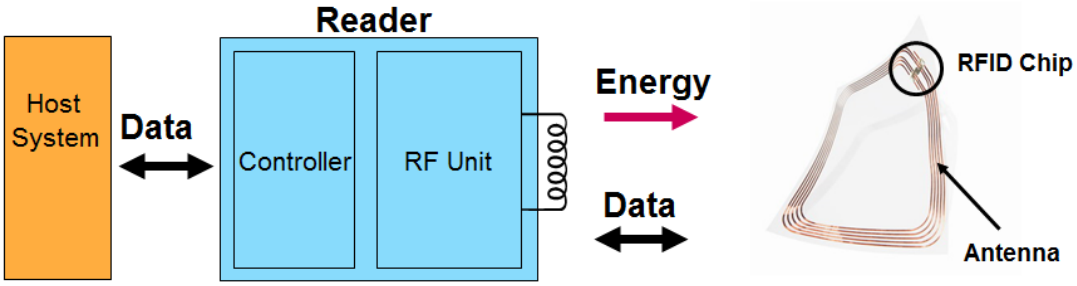

Figure 2.1. RFID Communication

Depending on the purpose, different radio frequencies are chosen to communicate between RFID components. Therefore, RFID technology is categorized according to the frequency involved; 125 kHz is referred to as "Low Frequency", 13.56MHz is referred to as "High Frequency" and 840-960MHz is referred as "Ultra-High Frequency" (Mayes & Markantonakis, 2008). Ultra-High Frequency (UHF) is used for long-range communications. The maximum range is calculated according to Figure 2.2.

$$R_{max} = \sqrt{\frac{P_{EIRP} \cdot G_{Label} \cdot \lambda^2}{(4 \cdot \pi)^2 \cdot P_{Chip}} \cdot \vartheta_{Matching} \cdot \vartheta_{Polarisation} \cdot \vartheta_{Antenna}}$$

Figure 2.2. Maximum range calculation formula

In order to give an example of maximum range calculation, we can use values below that are defined under the United States and European Union regulations;

Table 2.1. *Range Calculation Attributes*

Variable	*US Regulations*	*EU Regulations*
P_{EIRP}	4W	3.28W
G_{Label}	1.64	1.64
f	915MHz	869MHz
P_{CHIP}	35µW	35µW

$\vartheta_{MATCHING}$	0.8	0.8
$\vartheta_{Polarisation}$	1	1
$\vartheta_{Antenna}$	0.5	0.5

The maximum range Rmax under the United States and European Union regulations are given in Figure 2.3 and Figure 2.4, respectively.

$$R_{max} = \sqrt{\frac{4W \cdot 1.64 \cdot 0.33m^2}{(4 \cdot \pi)^2 \cdot 35 \cdot 10^{-6} W} \cdot 0.8 \cdot 1 \cdot 0.5} = 7.19m$$

Figure 2.3. Rmax in US

$$R_{max} = \sqrt{\frac{3.28W \cdot 1.64 \cdot 0.35m^2}{(4 \cdot \pi)^2 \cdot 35 \cdot 10^{-6} W} \cdot 0.8 \cdot 1 \cdot 0.5} = 6.90m$$

Figure 2.4. Rmax in EU

An RFID Reader (also called as Transponder) powers the field, transmits energy and data by inductive coupling. A passive tag, having a coil inside, generates the current out of the electromagnetic field generated by the reader. The tag uses this current to power up its Operating System and starts demodulating the data transmitted by the reader. And then responds to the commands using load modulation (Mayes & Markantonakis, 2008).

2.1.2. Contactless Cards

The usage of contactless cards for short-range payment and identification is based on new form factors derived from RFID technology. A contactless card has a chip and an

antenna to make the contactless coupling and communication as shown in Figure 2.5. ISO 15693 based cards and ISO 14443 based cards are used for this purpose. It has the same RFID communication principles: a reader is powering the field and a passive tag (contactless card) uses this power to operate and responds to the reader. Today, there are many use cases of contactless cards such as door access, transport payment, loyalty cards, and contactless credit cards.

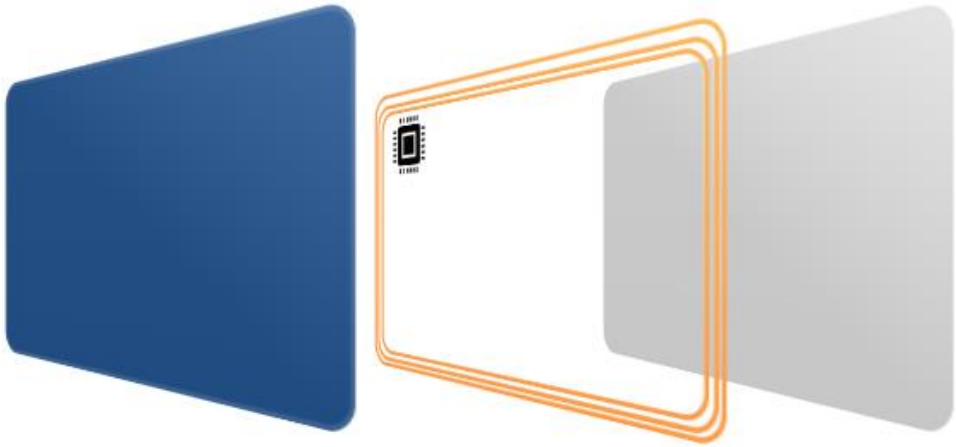

Figure 2.5. Contactless Cards

As any device having the chip and an antenna can function as a contactless card, different form factors of contactless payment and identification instruments are derived such as key fobs, wristbands, and watches as illustrated in Figure 2.6.

Figure 2.6. Illustration of an NFC-Enabled Watch

2.1.3. NFC Technology

NFC communication protocol is a subset of RFID communication protocol that is defined as a standard to ensure the operating distance, the communication protocol, and the security mechanism. NFC uses 13.56MHz frequency and the operating distance can be 10 cm maximum. Therefore, NFC technology offers a contactless, fast and secure communication channel between two NFC media (Coskun, 2012). The communication speed can be 106 kbps, 212 kbps, 424 kbps and 848 kbps that can be agreed between two NFC entities at the communication setup phase. This communication speed is low compared to other wireless communication technologies such as Bluetooth. However, NFC offers a quicker pairing than Bluetooth. Therefore, it had been considered for single-tap transactions such as door access/attendance (Fernandez, 2013), transportation payment (Widmann, 2012), loyalty (Ozdenizci, 2013), credential verification (Alpar, 2012), voucher payment (Van Damme, 2009), credit card payment (Kanniainen, 2010), proprietary payment schemes (Gronli, 2015), attendance control (Fernandez, 2013) and even for other verticals such as wireless power transfer systems (Biswas, 2018), airport baggage claim systems (Renardi, 2017), and game-based learning applications (Dzafic, 2017). Similar to RFID, NFC communication requires a reader to power up the field and a tag to respond to reader

commands. Besides, NFC technology offers a Peer-to-Peer communication in which both NFC entities can act as reader and tag at the same time for bidirectional communication.

NFC Technology standards are defined by SonyTM and Philips Semiconductors in 2004 in order to enable mobile devices functioning as Contactless Readers or Contactless Cards.

2.2. NFC Modes

2.2.1. Reader Mode

Reader Mode allows a mobile device to read and write an NFC tag (Hang, 2010) as shown in Figure 2.7. This mode is similar to RFID transponder mode as the reader plays an active role in powering up its electric field so that any tag in front of it can be booted using this power. The communication always starts from the reader's side; it sends a command and receives a reply from the tag. An NFC device in Reader Mode can be either a reader in ISO14443 Type A, ISO14443 Type B, ISO15693 or FeliCa mode. The device can be configured to poll all these modes consecutively or can be configured to accept only one type. NFC Reader Mode is the ideal mode to trigger a task at a mobile device by just tapping the phone to a tag. This mode is also used to trigger a payment or identification transaction by NFC phone (Konidala, 2012).

Figure 2.7. An NFC Phone in Reader Mode reads NFC Tag

2.2.2. Peer-to-Peer Mode

NFC Peer-to-Peer mode is not possible with contactless cards because it requires both entities to have a power source and any of them can have the active role depending on the communication initiation. This mode is introduced by NFC standards in ISO18092. Peer-to-Peer mode allows two NFC devices to exchange information between one another (ISO/IEC 18092, 2004). In this mode, both sides can play the active role. Generally, this mode is initially preferred for non-secure applications such as the exchange of information related to gaming, contact card information, etc. as shown in Figure 2.8 (NFC Forum, 2013). However recently, researchers also tried to utilize this mode to establish a mobile transaction flow in which entities exchange public information over NFC Peer-to-Peer interface, then communicates to their banks through secure communication channel (Bojjagani, 2019).

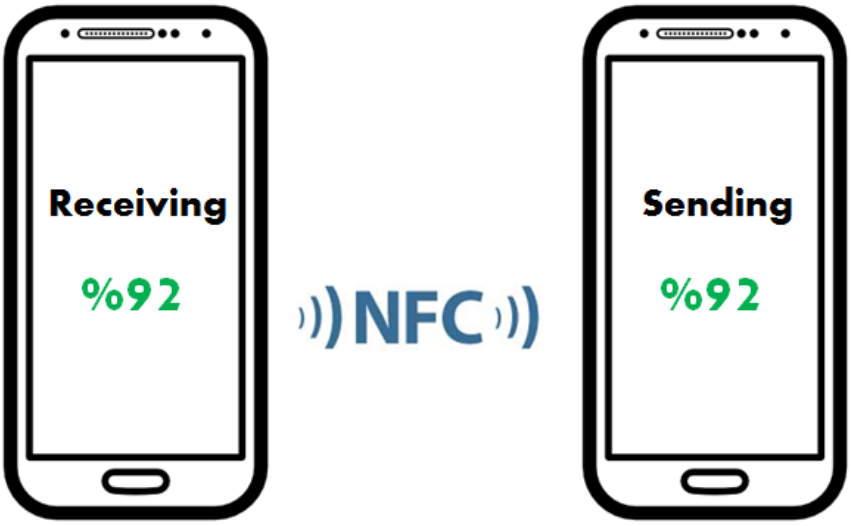

Figure 2.8. Phone-to-Phone file transfer initiated by NFC P2P Mode

2.2.3. Card Emulation Mode

Card Emulation Mode allows a mobile device to function as a contactless card. In this mode, the device needs to have a secure element to perform actual card-related operations (Coskun, 2012). The device emulates the card using its antenna which acts as the antenna of a contactless card. In this mode, the device is in the passive mode which means an active device is necessary to initiate a communication.

Although each option has its advantages, Card Emulation gained interest in the secure identification industry because this mode allows the users to convert their mobile handsets into a mobile wallet which contains credit card information, offline payment cards, loyalty cards, et cetera; all of which are stored inside the SE (Liu, 2015). More than a decade ago, mobile payments are started to be seen as the feature of the digital payment industry (Hassinen, 2006) and today it is possible using NFC Technology.

Figure 2.9. NFC Phone as a Payment Instrument

2.3. Secure Element

Main motivation behind the invention of NFC technology was using it for applications requiring security such as payment and secure access (Benyo, 2007). Using mobile handset for payment applications reduces the operational costs of a regular card-based system as such mobile device is already issued to the User (Lehdonvirta, 2009). Therefore, the initial applications were targeting making a digital wallet using an NFC phone. Today, there are many mobile wallets based on NFC technology but mainly driven by global players such as Apple Pay, SamsungPay and Google Wallet as the reputation of payment system provider plays an important role in payment industry (Koster, 2016). All of these applications used Secure Element (SE) to store confidential credentials of the user as the SE is proven to be secure by the certification authorities. In the payment industry, it is a common practice to use certified hardware to store user credentials as the leakage of such data may result in fraud in the payment system. Secure Element is dedicated hardware that is tested against known hardware and logical attacks and proven that it is secure. Secure Elements are certified under Common Criteria (CC) Evaluation Assurance Level (EAL) or Federal Information Processing Standard (FIPS) publications. Secure Element (SE) term is referred to the secure hardware and the secure Operating System running on it. Commonly used

Operating Systems are MULTOS (Multi-Application Smart Card Operating System) and Java Card OS.

After the invention of NFC Technology, several different hardware that are available or can easily be attached to a mobile handset are used as Secure Element. Figure 2.10 lists available options for having a Secure Element on a mobile device (Reveilhac, 2009).

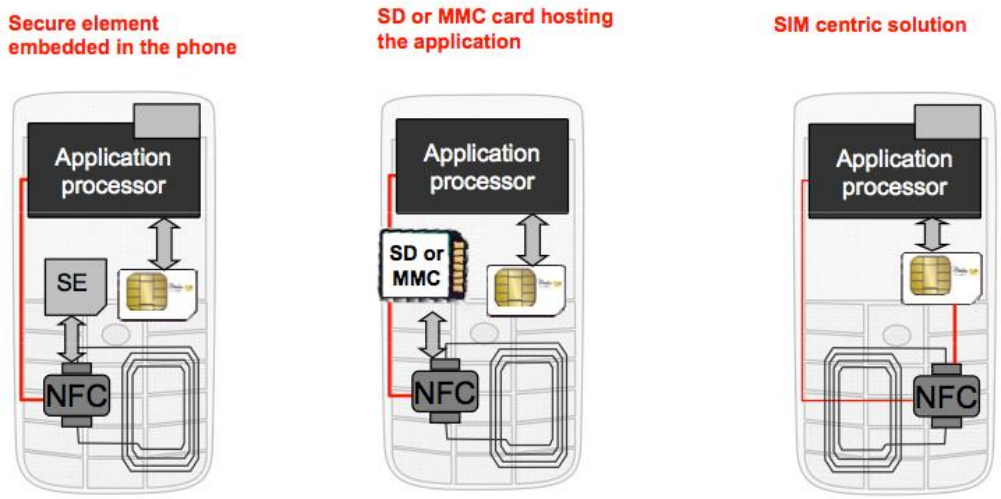

Figure 2.10. Secure Element options on a Mobile Device

Option 1 (Embedded SE): A Secure Element that is already assembled on Mobile Handset is used as the SE of NFC. This solution is not without Mobile Handset manufacturer dependency as they have the exclusive access to their own hardware.

Option 2 (Attached SE): A Secure Element that is available in a memory card can be used as the SE of NFC. This solution brings more flexibility but not practical as many smartphones do not allow memory cards.

Option 3 (SIM-based SE): The SIM cards that are available in each phone are having the same security requirements of a Secure Element. SIM cards are secure hardware components that can be used for secure applications (Ok, 2016). Therefore, they are good candidates to be used as an SE of NFC. However, this solution is not without

MNO dependency as the SIM cards are under exclusive control by GSM Operators (Eberspaecher, 2008).

2.3.1. SE Structure

A Secure Element is a multi-application smart card which can contain several chip applications (applet) inside by ensuring memory separation between them. SE contains a master application called Card Manager in order to manage further application installation, deletion and assigning access rights. Figure 2.11 shows the architectural overview of Java Card based Secure Element (Java Card Platform Specification, 2009).

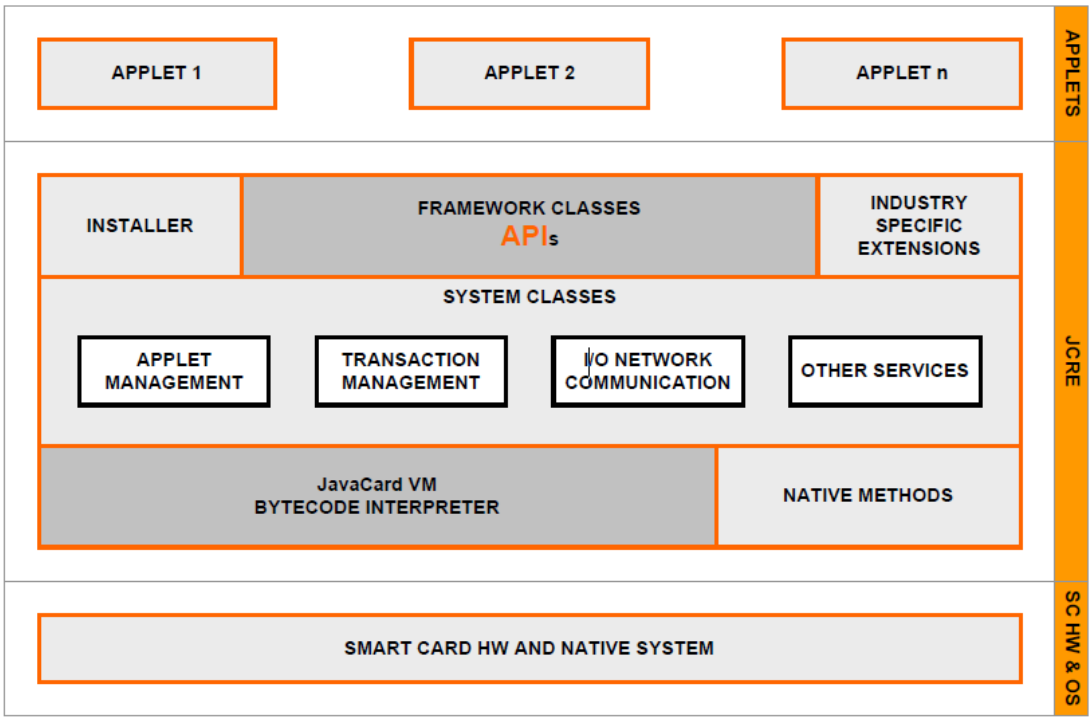

Figure 2.11. Architectural Overview of Java Card OS

On a Secure Element, only one applet can be active at a time and the Operating System directs the commands to the activated applet. Each applet is identified by its Applet

Identifier (AID) and it is used to select the desired applet. A sample reader to applet communication is shown in Figure 2.12.

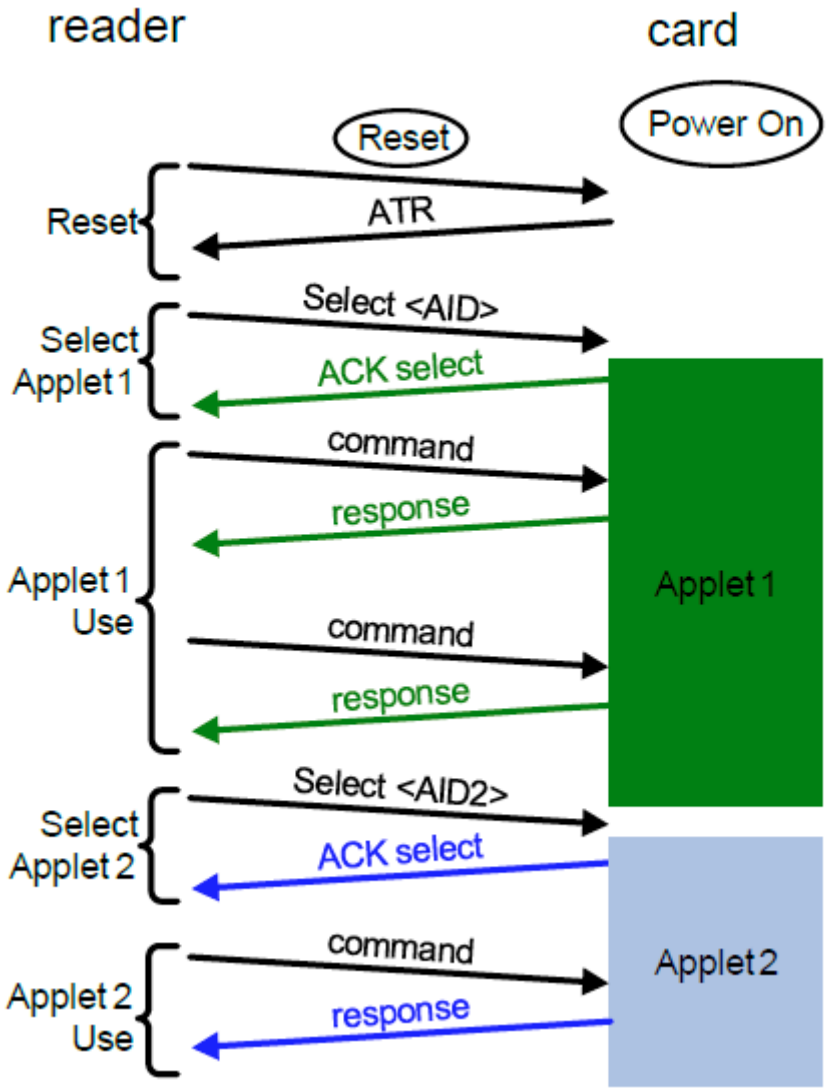

Figure 2.12. Reader to SE applet communication

The SE content is managed by the Card Manager based on the Global Platform guidelines. An SE is issued by the Issuer and it has its own Security Domain on the SE. Card Manager (CM) allows Issuer to authenticate the SE using the Issuer Keys and allows the authenticated user to update the SE content. Applet loadable files (CAP

File) are loaded to the SE through CM and become available for usage by the following installation commands. This scenario is depicted in Figure 2.13 (GlobalPlatform, 2006).

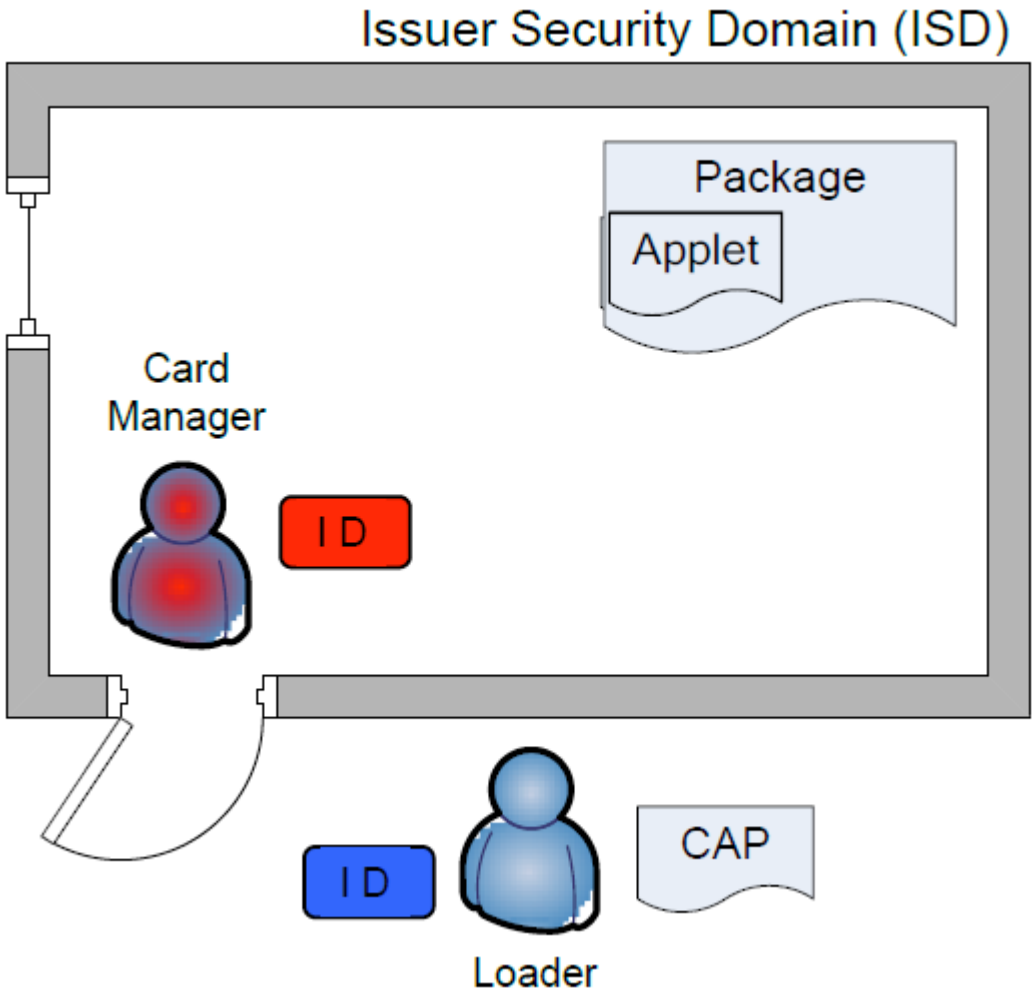

Figure 2.13. Secure Element Structure

Many SE today also offers Supplementary Security Domain (SSD) besides Issuer Security Domain (ISD) in order to organize the applets logically and to assign different access keys to different logical regions on the chip (GlobalPlatform, 2006).

2.3.2. SE Authentication

Secure Element Operating Systems accepts all of the commands sent from the reader and determines the action using its decision tree. If there is an active applet (selected prior to this command), the OS forwards the command to the applet for handling. If there is no selected applet, OS executes the command and responds if it is a generic read command which does not require authentication. Any other content change commands require an upfront authentication using the relevant Security Domain Keys. This authentication is performed using Initialize Update and External Authenticate commands that are explained in Global Platform specifications.

2.3.3. Host Card Emulation (HCE)

As Secure Element content management is dependent on the SE Issuer, an alternative usage called Host Card Emulation (HCE) is introduced in order to solve dependency problems. HCE is based on the NFC controller to direct the received command to the Mobile Handset CPU which will then be directed to the listening Mobile Application (Svitok, 2014). And the command result will follow this route in the reverse direction to be delivered to the reader. This method is called HCE, also known as Software Card Emulation. Although this method can be considered as a solution to the SE dependency problem, it comes with its share on security. As stated in the previous sections, the payment industry firmly depends on hardware components to ensure the security and confidentiality of the stored user credentials. Furthermore, researchers started to focus on improving transaction security of NFC EMV transactions after identifying vulnerabilities in EMV transaction processing flow even if hardware components are involved (Al-Haj, 2018). Therefore, we do not consider HCE as an alternative solution to the SE dependency problem.

2.4. NFC Communication Standards

The communication of an NFC Enabled Device is defined by several international standards as listed below (ISO14443, 2011; ISO18092, 2004; NFC Forum, 2016; ISO7816, 2013);

2.4.1. ISO14443

This is the communication standard coming from Proximity Coupling Smart Cards. The standard defines the card characteristics, methods and format of the data transmission and also the anti-collision method to select a single card for communication when several cards are available at the same time in the range of the reader. ISO14443 is the focus of this thesis as our NFC Feature Box applet communicates to the reader using this standard.

2.4.2. ISO15693

This standard is included in NFC devices in order to stay compatible with Vicinity Coupling Smart Cards. Vicinity cards offer longer read/write distance compared to Proximity cards. As the main focus of NFC is on secure applications, offering longer distance on the mobile device did not gain as much interest as proximity. Therefore, this standard is not the focus of this thesis.

2.4.3. ISO18092

As explained in previous sections, this standard is introduced with NFC Technology to let bidirectional communication between two NFC entities. The NFC Feature Box is based on Reader/Card communication of NFC therefore; this standard is not in the focus of our thesis.

2.4.4. NFC Forum

NFC Forum is a non-profit industry association of the NFC ecosystem and defines standards of NFC devices for interoperability. NFC Forum categorizes the NFC tags into 4 types depending on the used technology (NFC Forum, 2016). NFC Feature Box applet explained in this thesis can be implemented as an NFC Forum Tag Type 4.

2.4.5. ISO7816

This standard defines the contact communication of a smart card. A smart card having a contactless interface also connected to the same chip inside the card. Therefore, a contactless smart card has the same command/response execution of contact-based smart cards except for the communication channel creation and activation steps. As a result, ISO7816-4 is part of NFC communication.

2.4.6. APDU

Application Protocol Data Unit (APDU) is the command and response frame standard used in ISO7816-4 data transmission between NFC devices. Command and Response APDUs are having the structure shown in Figure 2.14.

COMMAND APDU						
MANDATORY HEADER				OPTIONAL BODY		
CLA	INS	P1	P2	Lc	DataField	Le
1byte	1byte	1byte	1byte	1byte	DETERMINED by Lc	1byte

RESPONSE APDU		
OPTIONAL BODY	MANDATORY TRAILER	
DATA FIELD	SW1	SW2
VARIABLE LENGTH	1byte	1byte

Figure 2.14. APDU Structure

2.5. Card Issuance

Card Issuance is a very critical step in the Payment Industry because it requires the transmission of confidential information of the user and the security keys to the chip. Therefore, this step requires a secure environment or a secure channel to ensure that the transmitted data is not copied by third parties.

2.5.1. Regular Smart Card Issuance

The payment industry has been using plastic cards for decades. In their infancy, these cards only included the card number and owner's name engraved on the card. Then, cards were equipped with a magnetic stripe which contains the engraved information also as digitally coded in it. However, this magnetic stripe was physically exposed making the information it contained vulnerable to stealing and/or making unauthorized copies. Today, we use modern, tamper-resistant, chip-based smart cards capable of digitally identifying the card itself, its user and the validity of the card via cryptographic engines stored electronically inside the chip (Akram, 2015; Li, 2013). In short, card technology has evolved greatly over time, while its issuance sequence has essentially remained unchanged.

The regular Smart Card Issuance sequence is depicted in Figure 2.15. The following steps depict this Credit Card Scenario:

Step 1: Cardholder completes a credit application from a bank.

Step 2: Bank creates and sends the cardholder's information to the card maker.

Step 3: Card maker prepares and personalizes the card according to cardholder's application information. This step is performed in a secure facility allowing credentials to be securely loaded onto the card.

Step 4: The card is shipped to the cardholder and activated (over the telephone or a visit to the local bank branch) by the user.

Step 5: The cardholder is ready to make payments with the card.

Regular smart card issuance is fairly simple and all roles within the flow are well defined by international standards (Francis, 2010). Specific payment industry players such as Europay, Mastercard and Visa (EMV) define international rules to credit card and debit card preparation and personalization.

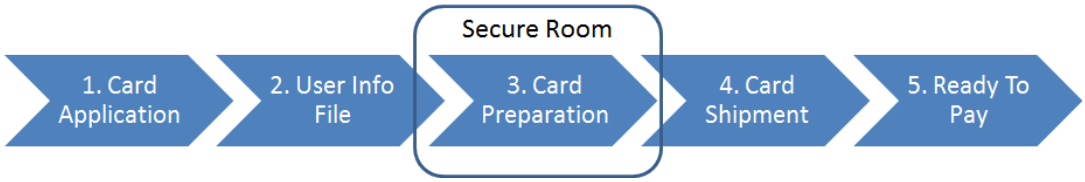

Figure 2.15. Regular Smart Card Issuance Flow

2.5.2. NFC Issuance

After the development of NFC Technology, card emulation mode introduced as the option to store the NFC phone owner's smart cards inside the phone's SE. The SE inside the NFC phone is reachable only by the SE owner (Enabler). In the case of SIM-based NFCs and embedded-NFCs, the Mobile Network Operator and the phone manufacturers have this capability, respectively (Alliance, 2011). For this reason, NFC issuance is dependent on the NFC Enabler because they have exclusive access to the authentication keys of SEs.

While NFC Enablers were capable of providing NFC issuance services in their own right, the introduction of Trusted Third Parties to the system intends to reduce the complexity of the NFC Enabler and Service Provider integrations (Madlmayr, 2008). Referred to Trusted Service Managers (TSM), these integrations serve as a proxy provider to Service Providers, allowing them to reach the SE of the target user. For this reason, the NFC Enabler and TSMs play an active role in NFC issuance.

The NFC Issuance Sequence is depicted in Figure 2.16 and the following steps depict this Credit Card scenario:

Step 1: User owns NFC Phone featuring either SIM-Based or Embedded Secure Element.

Step 2: User submits an application with the bank to activate Credit Card on the mobile device.

Step 3: Bank detects the SE owner of the User's phone.

Step 4: Bank detects the TSM that is serving the detected SE.

Step 5: Bank prepares card information and generates remote issuance data.

Step 6: Bank establishes secure communication with TSM servers.

Step 7: TSM creates a remote secure channel to the SE.

Step 8: TSM performs the remote installation.

Step 9: User is ready to make payment with the mobile device.

These steps can vary, depending on the SE Owner or proprietary NFC schemes that have already been deployed in the field. However, in all the cases, NFC issuance requires a remote secure connection to the SE to securely transfer confidential credentials.

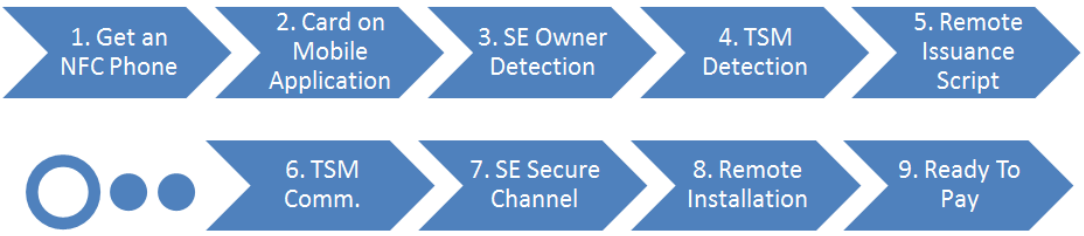

Figure 2.16. NFC Issuance Flow

2.6. Related Work

Dependencies in mobile systems cause several problems in the mass adoption of available services therefore, this area had been targeted by many academic studies. Furthermore, corporations in the financial industry contributed to this field by researches and even protected resulting intellectual properties with international patent publications. Here below, we list some of them which are related to our research area.

2.6.1. Ruiz-Martinez et al.'s Method

Mobile Network Operator (MNO) dependency in mobile signature services is a blocking factor that users can only get this service from their operator and there is no interoperability (Ruiz-Martinez, 2011). Ruiz-Martinez et al. define a new method for users that they can use their mobile handsets to perform m-signatures that are not dependent on any MNO. The method is derived from European Telecommunications Standards Institute (ETSI) specifications for mobile signature services on SIM (ETSI, 1996) but developed as to make it MNO independent. Although this study is an important step in resolving enabler dependency problems in the field, the study is limited to a single function and not possible to extend for global payment and identification challenges that NFC Feature Box is solving.

2.6.2. Pourghami et al.'s Protocol

An alternative method in order to solve enabler dependencies on NFC payments is to use a Cloud-Based payment approach (Pourghami, 2014). Cloud-based payment processing is linked to the NFC Phone of the User in a secure way and a new payment execution protocol is generated out of the advised transaction execution flow. Together with the rise in cloud computing solutions, re-defining payment execution by utilizing cloud resourced had gained interest. However, offline transactions

especially close-circuit proprietary payment schemes have hard restrictions over completing a financial transaction in a limited time that can only be achieved by offline transaction processing. Therefore, NFC Feature Box solves NFC Enabler dependency without requiring transaction acceptance devices to access online resources while performing the transaction.

2.6.3. Suryotrisongko et al.'s Method

Another alternative way to remove dependencies and reduce the infrastructure cost is to use a feature that is available on all smartphones. Reading Quick Response (QR) codes to identify the digital wallet and making a payment can be utilized by only using the camera of the smartphone (Suryotrisongko, 2012). This smart solution is applicable to several proprietary payment and identification systems in which users are not in a rush to perform a transaction and can operate their phones to capture QR codes in the required angle and perspective. However, in global payment and identification system space usually, offline transactions are preferred because such systems perform a high volume of transactions per minute. Therefore, expecting the end-user to capture an image to perform a transaction is not practical. Furthermore, from a user experience point of view, just tapping a phone to a terminal has an incomparable advantage over trying to capture a QR code in the right perspective to perform a daily life transaction.

2.6.4. Lyne et al.'s Method

After having NFC technology as a function available in mobile handsets another improvement observed in the financial industry was utilizing mobile handset as NFC transaction terminal. This allows replacing over-the-counter payment terminals with mobile applications that can be downloaded and deployed on to NFC-enabled mobile handset that merchants already have. However, as payment terminals require having

a secure element to conduct secure communication to the payment media – a contactless card – converting an NFC phone into a mobile point of sale device is not without NFC enabler dependency as it requires issuing point of sale application/credential into the SE. An international patent publication that had been filed by Cubic Corporation, Inc. San Diego, CA, US provides details of having Secure Element independent personal point of sale application on a mobile device that can read and write contactless memory cards (Lyne, 2016). The invention suggests communicating to the smart media over NFC interface by establishing the secure communication channel from a remote server to the smart media directly, and so bypassing the for an SE on a mobile device. Although this invention is a great improvement in the NFC enabler dependency domain, it is only solving the problem when the mobile handset is positioned as the entity that is collecting the payment. Whereas, the vast majority of users having an NFC-enabled phone are using it as a media to make payment.

CHAPTER 3

NFC ONLINE TRANSACTIONS

NFC financial transactions are categorized into two depending on the connectivity need for transaction decisions; online and offline transactions. In both cases, the physical presence of the mobile handset in front of the terminal is required. A transaction is initiated when the user physically taps the phone to the transaction acceptance terminal. However, keeping the phone on the terminal during the whole transaction execution timeframe may not be necessary; it varies based on the transaction execution flow. Some online transaction processing flows initially exchange some data between components and then terminal proceeds with online execution in order not to enforce the user to hold the phone on the terminal. Throughout our studies in NFC mobile transactions field, we have identified and provided solutions to blocking barriers of performing enabler independent and effective online NFC transactions as listed below:

3.1. Online Secure Element

As stated earlier, a practical solution to the enabler dependency problem is carrying the transaction decision process to cloud resources which can be controlled by the service provider. “Having 4G, Enabling Cloud-Based Execution for NFC based Financial Transactions” contributes to this effort by positioning NFC mobile device as a transmitter between transaction acceptance device and the cloud services when transaction flow reaches offline cryptographic processing (Turk, 2015a). Given the fact that cryptographic operations require more processing power than other operations, they also require a larger execution timeframe when they are executed in a Secure Element rather than being executed in a dedicated hardware security module (HSM).

Table 3.1. *Cryptographic operation execution times*

Device	*1024-bit Sign*	*2048-bit Sign*
Secure Element A	86 ms	437 ms
Secure Element B	121 ms	600 ms
HSM Device	0,14 ms	0,83 ms

Table 3.1 shows the performance variation of executing a cryptographic operation that is common in the financial industry, especially in online transactions. As it can be observed from the table, there is around 600 ms difference between performing an RSA 2048-bit signing operation on SE and HSM. In this study, we state and prove with experimental studies that the gain from carrying cryptographic operations to cloud resources can compensate for the loss on network latency especially on LTE network. Therefore, online secure element offers a solution to the dependency problem for financial transactions that rely on heavy cryptographic operations.

3.2. Preventing Connection Loss

Deployment of unattended devices in the field has an important share in NFC transaction space, such as access management solutions that are mounted on the wall of buildings and payment acceptance devices on vending machines. These devices require online connectivity either to perform the transaction or to synchronize with central systems to continue to accept transactions. Unattended devices can lose Internet connection and will remain non-operational until a field service technician reaches the physical location and solves the connectivity problem.

With NFC card-based solutions, when such an unattended device is facing connectivity problems it would mean all users will not be able to use services offered

by those devices. However, in today's world, we can use our NFC-enabled mobile phones as a replacement for our physical cards.

"Internet Connection Sharing Through NFC for Connection Loss Problem in Internet-of-Things Devices" introduces a novel connection sharing protocol over NFC interface to utilize the connectivity of the mobile device to allow the offline device to access online sources it needs and eventually allowing the execution of the transaction (Turk, 2015b). This study provides all the details about creating a connectivity sharing handshake before the transaction if one party needs to use the other party as a proxy to access online sources it needs. And then the transaction is executed by securely transmitting data packages over the sharing party.

3.3. Read Only NFC

After NFC technology became an available function in mobile handsets, card emulation mode gained the most interest in the financial industry besides Reader and Peer-to-Peer modes. Because this mode is the one that turns the mobile handset into a digital wallet that contains digital copies of our physical payment cards. Card emulation mode allowed users to tap their phones to payment terminals and the transaction is executed as if the physical contactless payment card is tapped.

In order to have an NFC phone in card emulation mode to function as a physical card, corresponding card application – also known as chip application – to be loaded into the Secure Element which is under the exclusive control of its owner; NFC Enabler. Therefore, starting from the early days of NFC technology, utilizing card emulation mode had been dependent on NFC enabler of the mobile handset, and it is still today.

On the other hand, besides bringing dependency to its enabler, Card Emulation mode in NFC technology is an optional function while Reader Mode is mandatory in all NFC protocols that are explained earlier in this thesis.

In order to solve the dependency problem in online mobile transactions through NFC devices, we leveraged upon Reader Mode which is available in all NFC devices. Reader Mode simply allows NFC handset to read any NFC device without relying on its enabler. We have introduced "RONFC: A Novel Enabler-Independent NFC Protocol for Mobile Transactions" (Turk, 2019). RONFC stands for Read-Only NFC that emphasizes the reader mode utilization approach of our solution.

In a secure payment transaction using secure hardware to generate a transaction cryptogram is crucial. Transaction data that is encrypted or signed by secure hardware which is proven to store cryptographic keys securely is a mandatory requirement in the payment industry. As Secure Element is certified hardware to store cryptographic keys securely it was the choice of processing NFC transactions with mobile devices.

The payment industry also mandates the usage of certified secure hardware in payment terminals which is required to create secure transfer between the terminal and its provider. RONFC proposes using secure hardware attached to the payment terminal to create a secure transaction cryptogram and to ensure the privacy and integrity of the executed transaction.

Terminals are under control of their providers – also known as Terminal Providers – therefore, the Secure Element attached to the terminals are accessible by Terminal Providers without requiring any third-party involvement such as enablers or Trusted Service Managers (TSMs).

All these facts allowed us to invent a novel mobile financial transaction protocol over NFC phones without relying on the Secure Element attached to the NFC device, thus removing the dependency to its enabler. We could achieve this goal by changing the transaction flow and generating a new mobile transaction processing protocol.

Like any other payment system our proposed model requires a Central Authority (CA) to be present in the ecosystem for managing communication and settlement between Card Providers and Terminal Providers.

RONFC mobile transaction executions steps are as follows:

1. Transaction flow starts with entering transaction details (i.e. amount) at the terminal as it is the common practice today in all credit card payment scenarios
2. The terminal connects to Secure Element attached to it (all payment terminals have Secure Elements for secure communication handling with their providers)
3. Terminal SE which contains RONFC application creates the transaction cryptogram
4. Terminal goes into Card Emulation mode to advertise transaction data and starts waiting for a notification of transaction completion
5. Mobile handset user launches Card Provider's mobile application that he wants to pay this transaction with
6. User taps the NFC phone to payment terminal as we are doing it today for legacy NFC payment, so user experience and transaction payment easiness remains the same
7. User sees transaction details on the phone screen, which is unique to RONFC as legacy NFC transaction processing is not capable of providing transaction details to the handset
8. User approves the transaction after performing user authentication factors forced by its card provider; such as fingerprint matching, face recognition verification, PIN verification or a combination of those. This is also unique to RONFC as legacy NFC processing cannot intercept the transaction to perform user verification after the phone is tapped to the payment terminal
9. Card Provider's mobile application sends transaction data and cryptogram to Card Provider through its proprietary secure communication channel.
10. Card Provider checks the user's available credit versus the requested transaction amount and approves/rejects the transaction. Transaction response data is generated and secured by Card Provider's Hardware Security Modules (HSMs) and this information is sent to the CA to be relayed to corresponding

Terminal Provider of the transaction. This flow is similar to inter-bank switch services to make credit card payment available to all users around the world.

11. Terminal Provider verifies the transaction response cryptogram. Then generates its own secure transaction result cryptogram and sends it to the corresponding Terminal which initiated the transaction.
12. Upon receipt of the transaction response, terminal connects to the SE which is the only component can verify the cryptogram of the response.
13. Depending on approval or reject status Terminal screen shows related messages and prints customer copies if necessary, for the transaction.

The aforementioned RONFC transaction flow is depicted in Figure 3.1.

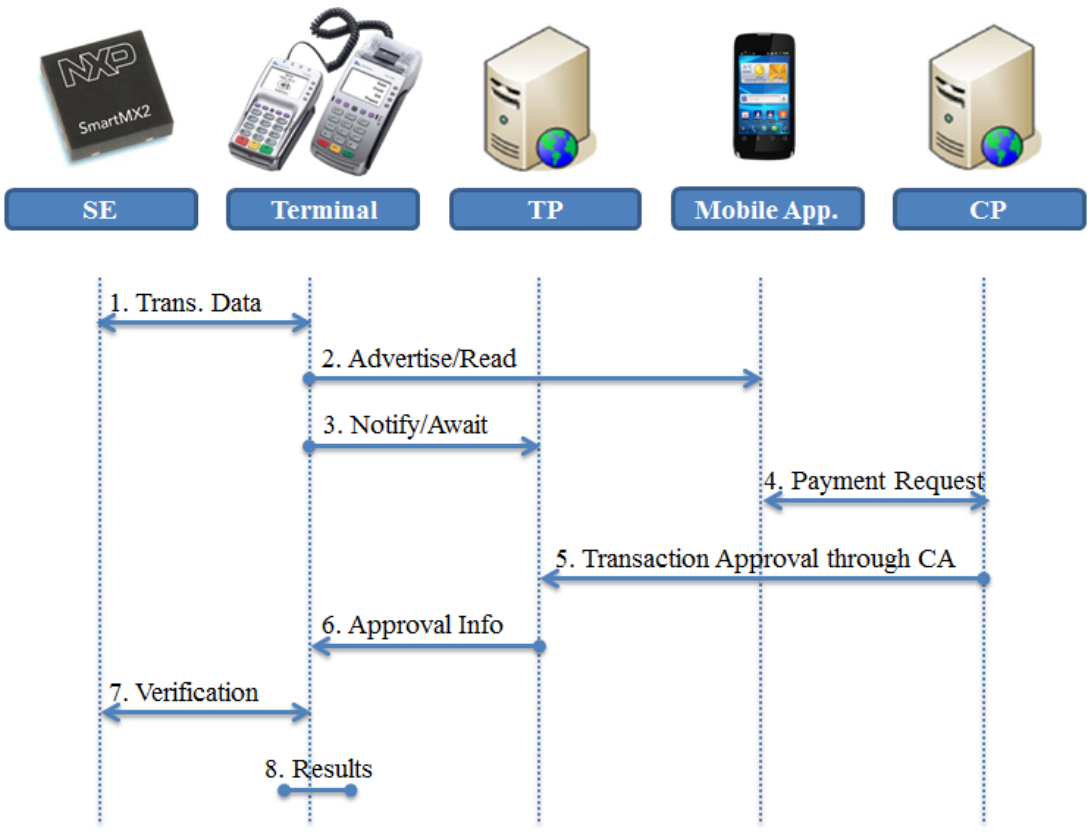

Figure 3.1. RONFC Transaction Flow

NFC enabled device coverage is increasing every year and people prefer paying with their phones because of the smooth payment experience it offers. RONFC allows Card Providers to enable NFC transactions for their customers without depending on NFC device owners or technology enablers, more importantly without being forced to pay transaction fees to those enablers.

3.3.1. Security of RONFC

RONFC offers end-to-end security by leveraging upon hardware security components and secure communication channels that are already established in financial industry players as shown in Figure 3.2. RONFC offers both transaction security and integrity by its secure transaction cryptogram design.

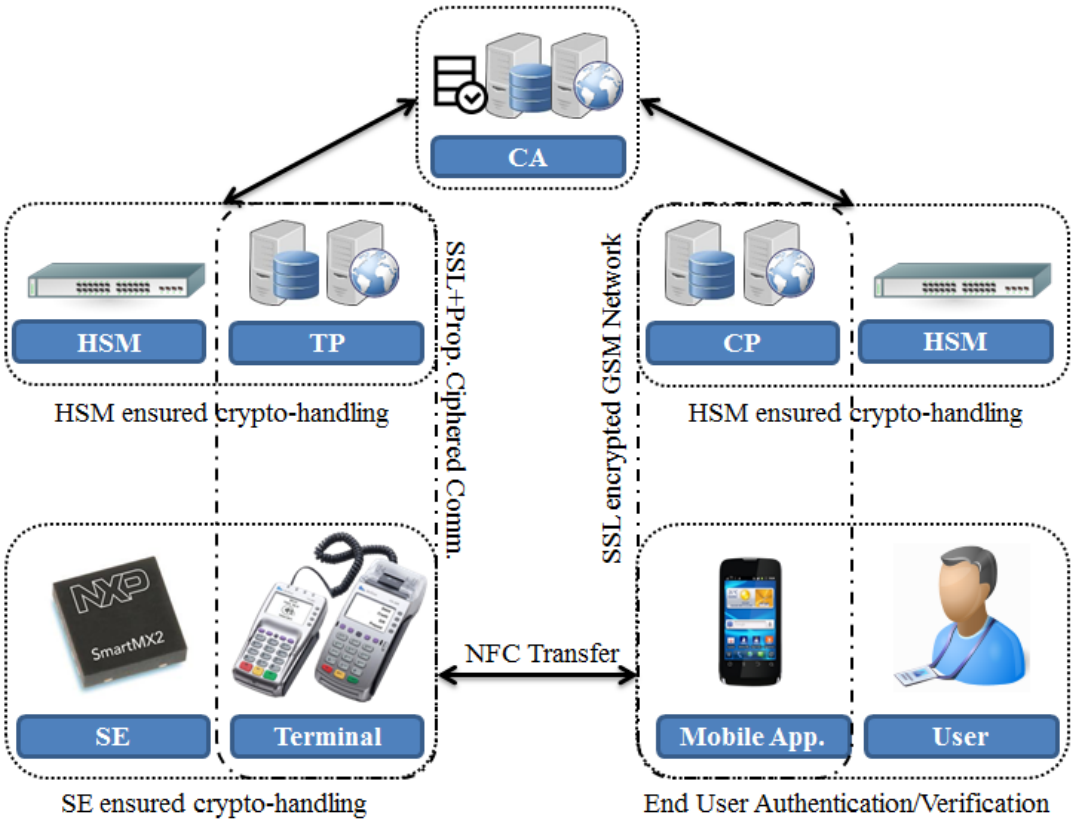

Figure 3.2. RONFC Components and Communication Channels

The Secure Element attached to the payment terminal plays an important role in ensuring the end-to-end transaction security of our proposed model. SE is the component that creates the secure transaction cryptogram at the beginning of the transaction, and it is again the component that performs final transaction result verification. Secure Elements are certified to be used in secure transactions after being tested against all known hardware and logical attacks. In our protocol, SE is used not only to store transaction encryption and signing keys securely but also to perform those crypto functions inside the chip. Thus, performing all sensitive operations within the boundaries of the certified secure area.

In RONFC the terminal is taking a transmitter role for delivering secure cryptogram to NFC device through NFC interface. Since the data transmitted over the NFC communication channel is protected by a hardware-backed cryptogram our protocol is protected against communication sniffing attacks. Furthermore, internal counters and hardware-backed true random data usage inside the chip application protects RONFC from logical attacks.

Given the fact that all financial industry players are already maintaining secure communication to their devices, Terminal-to-TerminalProvider and MobileApplication-to-CardProvider secure communication channels are already maintained by these entities. RONFC protocol plays an active role when transaction information is delivered to those entities. RONFC mandates Hardware Security Modules (HSMs) at Provider centers to protect transaction encryption and signing keys. Transaction approval cryptogram that is generated by Providers is defined within the RONFC protocol to ensure transaction security, integrity, and protection against logical attacks including dishonest Provider attacks.

Legacy NFC transaction flow only relies on hardware security when the user performs a financial transaction, whereas a mobile handset is capable of performing other user verification methods such as fingerprint matching, face recognition verification, et

cetera. A recent study extended legacy NFC transaction processing with fingerprint identification as users experience a smooth and secure transaction through biometric matching (Zhang, 2018). RONFC allows mobile applications to enforce any kind of user identification/verification method as part of the transaction execution flow besides carrying over hardware-backed secure transaction cryptogram.

3.3.2. Performance of RONFC

The most important performance metric in the financial industry is the transaction execution time. And a well-performing payment transaction is a result of efficient usage of resources and optimized execution flows. RONFC is optimized to keep command/response count at a minimum during a transaction while preserving the security at maximum. In order to prove the performance and to illustrate practical applicability of our protocol in real-life payment scenarios we had implemented RONFC components and created an experimental setup with the following components:

- Server applications that are simulating Terminal Provider, Card Provider and CA roles
- PC application that is attached to a Smart Card through the PC/SC interface to execute Terminal functions
- RONFC based SE application loaded into a Smart Card
- RONFC based Android mobile application to simulate User functions

We had executed end-to-end RONFC transactions over experimental setup and measured time spent at reference points shown in Figure 3.3.

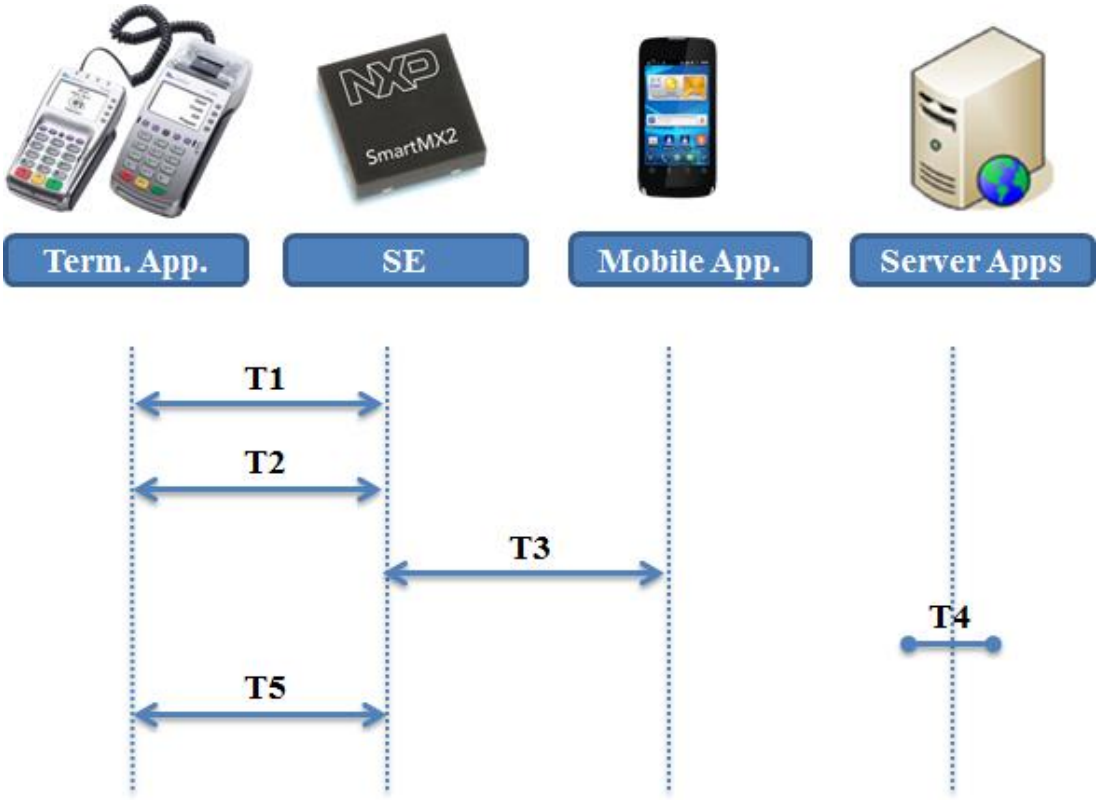

Figure 3.3. RONFC Performance measurement reference points

Roles of each reference point is as follows:

- T1: Is the time spent to calculate transaction cryptogram. This reference point is important as it includes the performance cost of creating a secure cryptogram that ensures the overall transaction security.
- T2: Is the time spent to bring SE calculated data into NFC interface so that become available for NFC phone to read it.
- T3: Is the time spent by NFC phone to read transaction data from the terminal. This reference point is important as it is the total duration that the user needs to hold the phone tapped to the terminal.
- T4: Is the time spent during the flow of transactions from phone application to the terminal provider. This reference point gives us the total cost of processing transaction data at the providers' network.

- T5: Is the time spent by SE to verify the transaction. It is important as a transaction completion is done by SE after transaction cryptogram verification. This reference point provides information about the cost of verifying a secure transaction within the chip.

Table 3.2. *RONFC Transaction Performance*

Step	*Min*	*Max*	*Avg*
T1	342 ms	588 ms	386 ms
T2	33 ms	70 ms	42 ms
T3	651 ms	2092 ms	874 ms
T4	280 ms	466 ms	321 ms
T5	306 ms	422 ms	345 ms
Total	1612 ms	3638 ms	1968 ms

Table 3.2 contains results of execution 20 RONFC transaction on our experimental setup. A transaction is completed within 2 seconds on average excluding the network communication overhead. As shown by the T3 reference point RONFC requires a user to hold the phone on the terminal for less than a second. All crypto calculations at each component took less than 500ms.

As a result, RONFC achieves NFC enabler independency by changing the transaction execution flow without losing security strength. The contributions of RONFC in the NFC mobile transaction field are explained in the Conclusions chapter.

CHAPTER 4

NFC FEATURE BOX

4.1. System Overview

Banking cards are provided by international players such as Master Card, Visa and American Express. Therefore, putting a banking application into NFC SE is simple as these players have world-wide standards for their payment applications. After having NFC functionality on mobile phones, these players updated the remote issuance schemes and having a credit card in our NFC phone became possible. However, non-banking payment schemes are not in a standard way and require proprietary transaction execution flow depending on the system provider. Likewise, card-based identification systems perform proprietary transaction execution for cardholder verification. Thus, enabling an NFC phone to be used in a proprietary payment or identification system requires adding the specific application into the SE. This then leads to the aforementioned NFC adaption problems. Besides, putting all proprietary transaction applications onto the SE is not practical as it has very limited memory.

In this thesis, we propose a container application – NFC Feature Box – to cover all non-banking payment and identification transaction schemes as instances underneath. It is a single application implementing a linked list of instances that dynamically extends and shrinks upon the instance creation and deletions. The user has full control over the features in the NFC phone, namely, can add or remove any feature at any time. As a result, NFC Feature Box is NFC Enabler-independent; regardless of its NFC being SIM-based, Embedded or Attached SE.

In Feature Box, each feature instance contains feature related information and a dynamic list of data files, value files, and the authentication keys. Each data file is mapped with one read access key and one write access key. Thus, a read or write

operation requires a prior authentication with the read or write key, respectively. Each value file is mapped with two keys that are authorized for increment and decrement operations like read/write authorization for data files. A single key can be authorized for many file operations.

A typical example of a non-banking payment system is public transport ticketing and a typical example of a card-based identification system is access management. In both cases, the majority of world-wide implementations are based on the symmetric cryptography; Triple Data Encryption Standard (TDES) and Advanced Encryption Standard (AES). Therefore, our proposed method supports TDES and AES options to be used as authentication keys. Although symmetric key cryptography is the common choice world-wide, our method can be extended with asymmetric key cryptography as well.

NFC Feature Box application on the SE offers a secure communication flow for data exchange channel creation between the SE and the Service Provider without having the necessity to have NFC enabler in between, thus achieving NFC enabler independency as shown in Figure 4.1. The secure channel between the SE and the Service Provider is based on Public Key Infrastructure (PKI) and a secure session key is generated for each channel creation. Service Provider uses this channel to create its instance in the phone SE securely.

After creating a feature instance on the SE, Service Provider updates its Transaction Acceptance Devices (TAD) to accept NFC transactions. The TAD implements feature selection steps to allow NFC Feature Box to activate the desired feature in its list. And then, a regular transaction execution flow is executed by the TAD to perform the desired transaction such as transport payment, allow access to the building, et cetera.

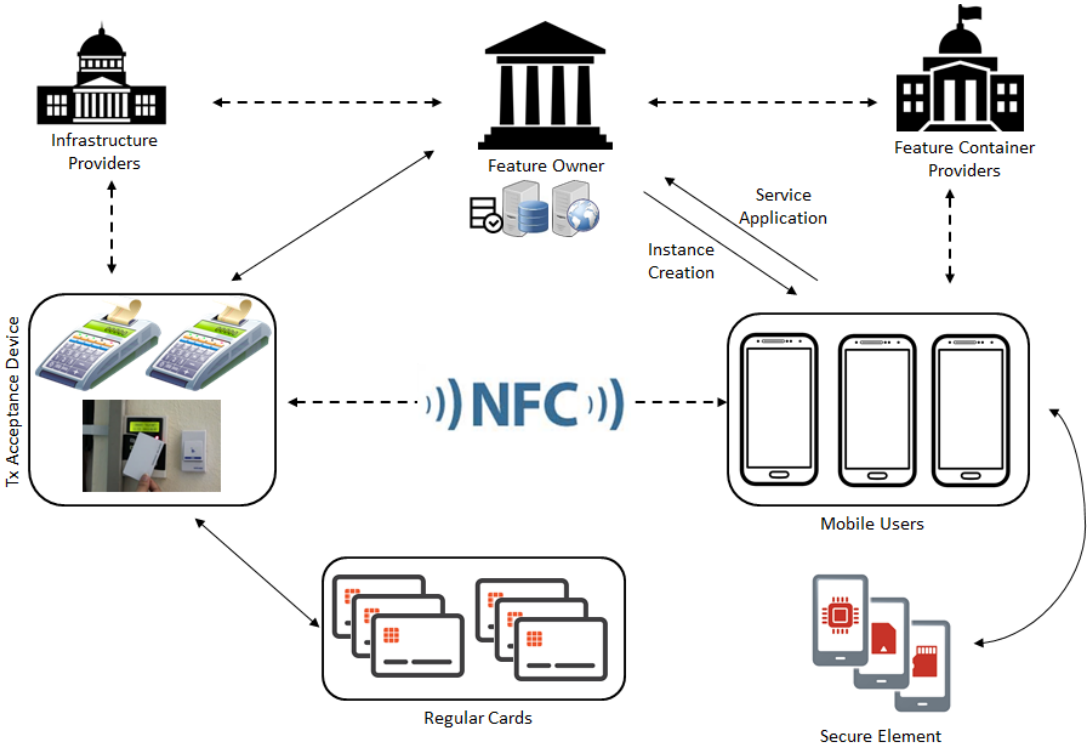

Figure 4.1. NFC Feature Box System Overview

4.2. System Stakeholders

4.2.1. User

NFC Feature Box utilizes User-Centric Approach; therefore, the User has full control over the NFC phone and also on the applications to be installed to and deleted from the phone. In a typical payment or identification system, a User is referred to any person having an instrument of the system to perform the contactless action. Therefore, a smart card holder is also the user of the system. In our proposed model, a User is referred to the person who wants to use the close-circuit payment or identification system using his/her NFC Phone. The user uses the mobile application of the Service Provider to activate the usage of NFC phone as a contactless smart card in the system that is offered by the Service Provider. In order to make a transaction using the NFC Phone user simply taps his/her phone to the Transaction Acceptance

Device and the rest of the transaction will be automatically performed between the TAD and the NFC Feature Box application in the SE.

4.2.2. SE Issuer

SE Issuers are Mobile Handset manufacturers, Mobile Network Operators or NFC-Enabled memory card manufacturers depending on how the NFC Technology is enabled on the phone. Current NFC Phones feature credit card payments by adding credit card application instances to the available Secure Element. This is possible because of the availability of credit card applications on the NFC Secure Element. Likewise, NFC Feature Box requires SE Issuers to add the NFC Feature Box applet in the manufactured NFC Secure Elements. As explained in previous sections, Secure Elements are under control by their issuers. Thus, they have access to the Secure Elements of the NFC phones that are in use already. This allows them to remotely access and install NFC Feature Box application to existing Secure Elements in the field. As a result, NFC Feature Box can become an available feature for existing users once it is started to be used as a standard for close-circuit payment and identification systems. NFC Feature Box application is a container having several feature instances underneath. An SE Issuer can offer a Mobile Application to their users to explore the content of the NFC Feature Box application and also offer functions to discover available features for the User that will make feature installation easier.

4.2.3. Service Provider

NFC Feature Box is designed to allow Payment and Identification related Service Providers to enable NFC Phone usage besides card-based systems that they are already operating. A Service Provider (SP) needs to implement the Web Services defined in this thesis in order to enable NFC Phone usage in their systems. SP is responsible to offer a convenient way for users to explore the Feature offered by the SP. Obviously,

the easiest way to achieve this is by using Mobile Applications that are either owned by the SP or SE Issuer. Additionally, Service Providers are responsible to manage the User Instance keys that are exchanged during the Feature registration. Common practices for secure key handling can be found in (Barker, 2007).

4.2.4. Transaction Acceptance Device

Transaction Acceptance Devices (TADs) are interface devices that Service Providers use to interact with users of the system. For a payment system, a POS terminal can act as the TAD and for an access management system, a contactless reader installed on the wall act as the TAD. TADs are already operated by the SP or Infrastructure providers, namely, they are under control of the issuer only. Therefore, making updates on the TAD device only requires access by its owner without requiring third-party dependencies. A Service Provider who wishes to enable NFC Phone usage in their live platform also needs to update their TAD execution flow to accept NFC phones in tandem with contactless smart cards. This is because of two main reasons. Firstly, the existing device might be working with lower layers of ISO14443 as NFC Feature Box requires ISO14443-4. Or, the existing device might be working in different protocol standards other than ISO14443-4. In this case, a firmware update would be necessary on the TAD device. Secondly, if the existing device is working on ISO14443-4, a software update will be necessary on the TAD device to let the TAD device distinguish a regular contactless smart card and an NFC Phone. The TAD device software will need a decision tree to determine which application to select depending on the type of used instrument for the transaction. Contactless smart card transaction execution can continue after selecting the card application of the SP, whereas NFC Feature BOX transaction will require querying the availability of the Feature of SP on the tapped NFC Phone.

4.3. NFC Feature Box Applet

The main component of the proposed method in this thesis is the Chip Applet that utilizes feature container for NFC Feature Services offered by Service Providers. While implementing our proof of concept applets we had worked on Java Card Open Platform (JCOP) chip which is based on Java Card and Global Platform Standards. The described protocol in this thesis can be implemented in any Global Platform compliant Smart Card Platform which offers Symmetric and Asymmetric cryptography.

4.3.1. Applet Properties

Applet properties are important when accessing the applet through the mobile device or card terminal interfaces. Although an applet can work with any proprietary properties, we advise using the below properties in order to ensure worldwide interoperability.

Table 4.1. *Applet Properties*

Attribute	*Value*
Package AID	0x46 65 61 74 75 72 65
Module AID	0x46 65 61 74 75 72 65 42 6f 78
Applet Instance AID	0x46 65 61 74 75 72 65 42 6f 78 31
Global Platform Version (used in tests)	GP 2.1.1
Java Card Version (used in tests)	JC 2.2.1

NFC Feature Box applet lifecycle is managed internally by the applet based on executed Personalization and Registration steps. Applet can have lifecycle states listed below and the lifecycle indicator byte is replied to the select command that will be explained in the following sections.

INSTALLED: After the applet is installed by the SE Issuer it becomes selectable and starts offering limited functionalities until Feature Instances are created underneath. During installation, NFC Feature Box applet generates ECC Key Pair that will be used in cryptographic calculations between the SE and SP.

PERSONALIZATION: NFC Feature Box applet automatically enters into this state when a Feature Instance creation is triggered and leaves this state when the Instance Creation is committed. After entering into this state, applet deselects previously selected Instance and the Personalization state is aborted if a Feature Discovery command is received. This behavior is to ensure that a Secure Element is not blocked with an incomplete personalization attempt that may never be completed.

POST_ISSUANCE: NFC Feature Box offers remote content update function for Service Providers. Applet enters this state automatically after receiving Post Issuance Request Command. After entering this state, applet deselects previously selected Instance and the Post Issuance state is aborted if a Feature Discovery command is received. This behavior is to ensure that a Secure Element is not blocked with an incomplete Post Issuance attempt that may never be completed.

OPERATIONAL: NFC Feature Box applet becomes Operational after the first Feature Instance creation. This state is the default state of the applet and it automatically falls back to this state if any error occurs in Personalization or Post Issuance states.

INSTANCE_SELECTED: Applet identifies that a Feature Instance is selected and active using this state. Applet returns into the OPERATIONAL state after receiving a deselect command or Transaction Commit command.

4.3.2. Feature Instance Manager (FIM)

Since the NFC Feature Box applet is a container of Feature Instances, Feature Instance Manager is the main component of NFC Feature Box to manage secure channel creation, feature instance creation and deletion.

Initially, the NFC Feature Box applet comes with an empty feature list and it allows Service Providers to add their features in it with a dynamic memory structure. In order to achieve this, Feature Instance Manager keeps a linked list of Feature Instances as shown in Figure 4.2.

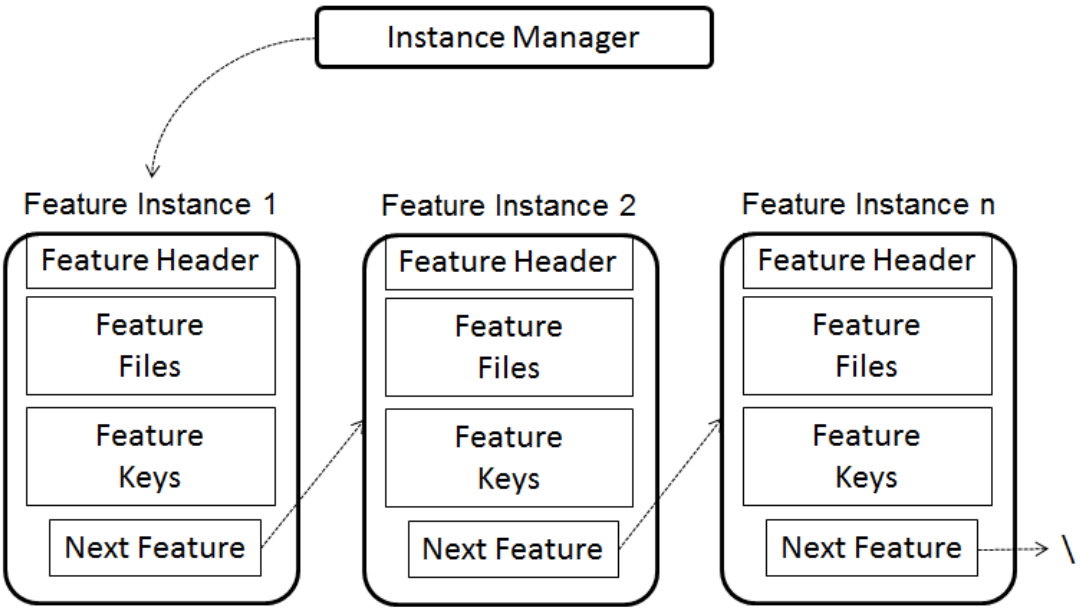

Figure 4.2. Feature Instances Linked List

Each Feature Instance keeps feature related information and a pointer that is pointing the next Feature Instance. One of the major problems in NFC Secure Element is the limited memory space it offers. In the current state of the technology, NFC Secure Elements contain credit card applications from various providers such as MasterCard and Visa. If the user uses his/her phone for credit card payment, the SE also contains credit card application instances that have personalized data and cryptographic credentials. As a result, an important part of the SE is already reserved for credit card applications. Credit card applications and instance creation routines are standardized worldwide, therefore, reserving a memory area for this purpose makes sense from the SE Issuer perspective. However, close-circuit payment schemes are local applications in general and adding a proprietary application into the SE by SE Issuer does not make

sense as it will not be valid except the small region of that local payment scheme is used. NFC Feature Box solves this issue by offering a dynamic structure for Feature Instances. SE Issuer has the responsibility to add the NFC Feature Box applet into the SE. And then, the User will have full control over the applet to add/remove any local Feature Instance into his/her phone.

To sum up, NFC Feature Box offers a way for local payment and identification service providers to use NFC phones in an interoperable system that is valid worldwide.

Feature Instance Manager Authentication and Feature Instance Creation steps explained in detail in the following sections.

4.3.3. Feature Instance

A Feature Instance is a standard but flexible file structure for payment and identification schemes that are using Symmetric key authentication mechanism. The majority of the close-circuit payment and identification schemes in the world are based on symmetric authentication methods that are using TDES and AES as the crypto algorithms. Therefore, NFC Feature Box covers the majority of the existing applications and also it can be extended to PKI based payment and identification schemes as NFC Feature Box requires the availability of PKI algorithms on the chip to perform Service Provider authentication and secure channel creation.

As shown in Figure 4.3, a Feature Instance contains Feature Header, Static Data Files, Value Files, Authentication Keys and Next Feature pointer for the linked list of feature instances managed by the Feature Instance Manager.

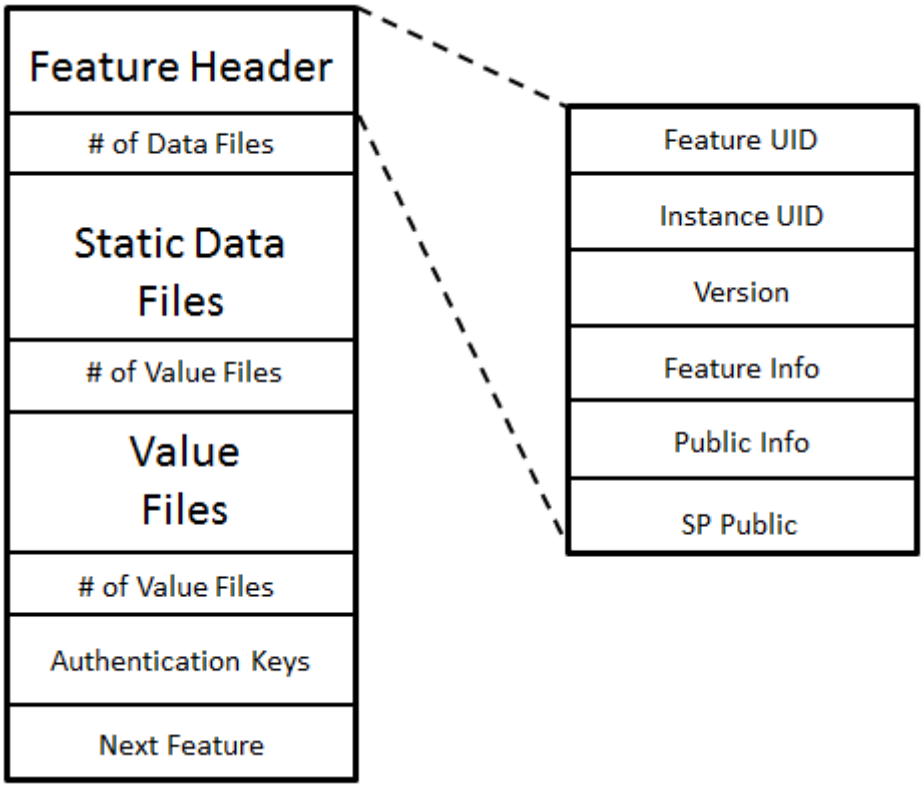

Figure 4.3. Feature Instance Structure

Feature Header and Next Feature variables in a Feature Instance are mandatory. Especially in non-critical access management systems, a card may not have any data inside other than card identifiers. Therefore, Static Data Files, Value Files and Authentication Keys are optional variables in NFC Feature Box.

Details of Feature Instance variables are explained below;

4.3.3.1. Feature Header

Feature Header is a set of mandatory variables that define the Feature Instance characteristics and its owner.

- Feature UID: This is the unique identifier of the feature offered by the Service Provider. Service Provider assigns the same Feature UID to all of the issued instances in NFC Phones so that it can be queried during the transaction to

determine existence in the NFC Feature Box. Feature UID has a length of 4 bytes.

- Instance UID: This is the unique identifier of the created feature instance. It is assigned by the Service Provider during the instance creation and it is under Service Provider's responsibility to assign true unique identifiers to each instance. Different Service Providers may assign the same UID under different features. However, Instance UIDs are associated with the Feature UID so uniqueness is guaranteed. Instance UID has a length of 4 bytes. After the instance selection, NFC Feature Box provides the Instance UID to the TAD to allow instance specific calculations. A typical example of this case is using a diversified key for each instance which is based on unique instance identifiers. Some system integrators keep a master key for their system, and they calculate diversified keys out of a unique input (can be instance UID) for each instance they issue allowing each instance authentication with different keys. TAD device only keeps the master key and performs the pre-defined key calculation using the Instance UID to derive the diversified instance key.
- Version: This is a two-byte version identifier for the Service Provider. Different versions of instances of a Service Provider may be active in the field at the same time which may require different execution flow on the TAD side. Version information is used to differentiate the corresponding versions of the instances that are available in the NFC Feature Box.
- Feature Info: This is the feature related public information that is shown to the mobile phone users in order to describe the service. This information is an important identifier when a mobile application is used to manage several feature instances in NFC Feature Box.
- Public Info: This is the information storage area that is provided to the TAD without requiring authentication. A set of non-confidential information about the instance and the User can be stored in this area to let TAD device easily identify the instance or the User before the authentication flow executed.

SP Public: This is the Public Key of Service Provider. The Elliptic Curve algorithm is used for Asymmetric crypto calculations between the NFC Feature Box and Service Provider as it offers higher security with lower key sizes compared to RSA cryptography (Bos, 2014). Therefore, SP Public information needs to be filled by the Service Provider during the Feature Creation. ECC algorithm of NFC Feature Box uses Prime Field, ANSI 192-bit keys.

4.3.3.2. Static Data Files

NFC Feature Box stores a variable to the keep total number of static data files used by this instance. A Feature Instance can store 255 static data files at maximum. The number of static data files is specified during the instance creation and cannot be changed later. Each static data file may have a different length. It is assigned by file creation command which is issued during the personalization of the instance. Static Data Files are the storage area of the instance to keep instance or User related information. Static Data Files are byte arrays and they are separated by their indexes under a feature instance. File index starts with zero and the last static data file has [(# of data files) -1] as the index value. File index is used as an identifier to identify the target file when a file related command is issued. NFC Feature Box offers partial read/write options to increase the applet performance and also to allow the Service Providers to combine several data files into a single data file. Each Static Data File has two key pointers to identify the read and write access on this file. Read access key pointer identifies the key to be used to authenticate in order to issue a read command on this file. The same applies to the write command.

4.3.3.3. Value Files

NFC Feature Box offers value files in order to easily store and operate numerical representations of values that are used in a payment or identification system. A typical example of a value stored in value files is the available credit of the User in an offline payment system. NFC Feature Box stores a variable to keep the total number of value files used by this instance. A Feature Instance can store 255 value files at maximum.

The number of value files is specified during the instance creation and cannot be changed later. Value files have 4-byte length to keep unsigned values. Therefore, the minimum value can be 0 and the maximum value can be 4.294.967.295 by default. Value File creation command allows setting minimum and maximum limits other than default values. Value files are identified by the index of the file. File index starts with zero and the value file has [(# of value files) -1] as the index value. The file index is used as an identifier to identify the target file when a file related command is issued. Value Files can be altered using increment and decrement commands. Each Value file has three key pointers to identify read, increment and decrement access on this file. Read access key pointer identifies the key which needs to be used to authenticate in order to issue a read command on this file. The same applies to increment and decrement commands.

4.3.3.4. Authentication Keys

NFC Feature Box allows the Service Provider to store symmetric authentication keys inside the Authentication Key area of the applet. NFC Feature Box stores a variable to the keep total number of Authentication Keys used by this instance. A Feature Instance can have 14 Authentication Keys at maximum. The number of authentication keys is specified during the instance creation and cannot be changed later.

Authentication keys are identified using their indexes. The authentication key index starts from 1 and stored sequentially up to the total number of keys. NFC Feature Box reserves the key index 0 for “forbidden” and the index 15 for “public” classification of the access right. This approach will be explained in detail within the following sections.

NFC Feature Box allows the Service Provider to choose a key type for each created key. NFC Feature Box supports below symmetric key types;

- 2KTDES (2 key Three DES implementation)
- 3KTDES (3 key Three DES implementation)
- AES128 (AES with 128-bit key length)

- AES192 (AES with 192-bit key length)

An instance under NFC Feature Box can contain different authentication key types at the same time. When an authentication command is issued, NFC Feature Box determines the key type using the key identifier sent in the authentication command and performs the rest of the crypto calculations based on this key type.

The structure of a created instance is depicted in Figure 4.4.

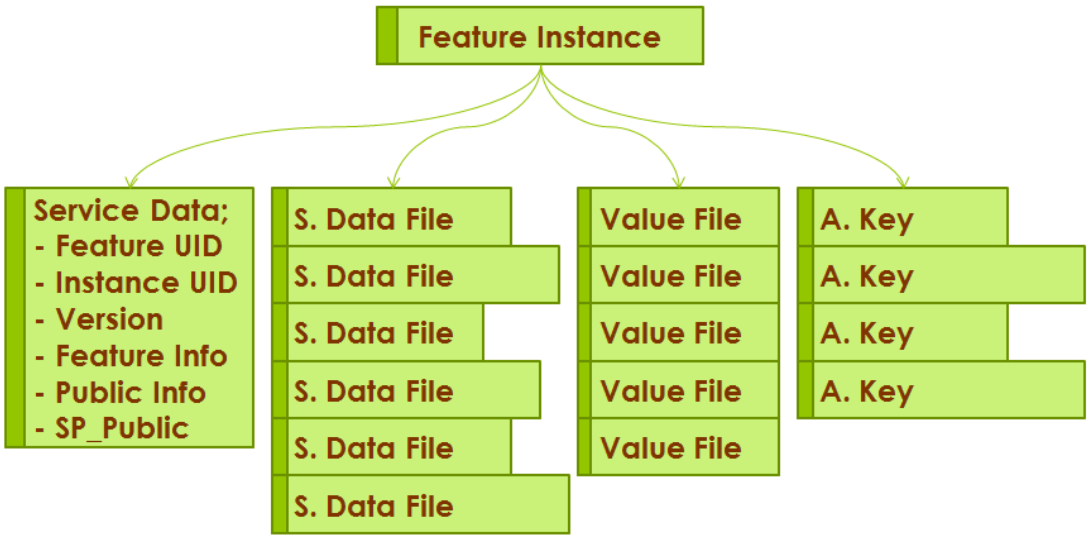

Figure 4.4. Created Feature Instance Structure

4.3.4. Key and Access Right Handling

NFC Feature Box requires each Static Data File to have two key pointers for read and write access and requires each Value File to have three key pointers for read, increment and decrement access. Keys are not exclusively reserved for one pointer. Several access pointers can point a single key. To illustrate, in a typical transaction system there might be a read key which is authentic to read any file in the system. NFC Feature Box allows assigning key identifiers to files to define the authorized key. Then, each received command is checked against the authentication requirement of the command if an authentication is required. NFC Feature Box

applet checks if the authenticated key identifier matches the assigned access right of the file. Figure 4.5 shows an example of access right assignment of four files using three authentication keys.

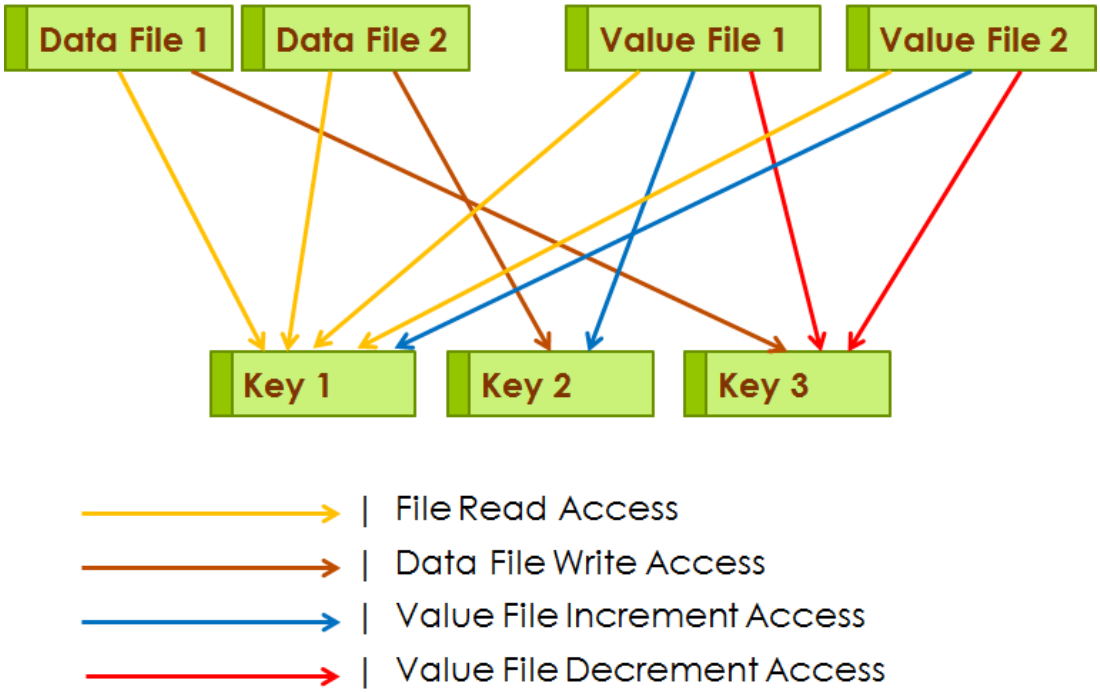

Figure 4.5. File Access Rights

In the given example Key 1 is authorized for Read Commands on all files and authorized for incrementing the Value File 2. Key 2 is authorized to write on Data File 2 and increment Value File 1. And Key 3 is authorized to write on Data File 1 and decrement both value files.

NFC Feature Box resets the authentication status after each instance selection or applet deselect.

4.3.5. Forbidden and Public Access Rights

In a payment or identification system, it might be possible to give a free access right to an action on a file. In general, it is required to increase transaction performance by performing non-critical actions without making an authentication and so without spending time on crypto calculations. To demonstrate, transaction history records of an offline payment system may contain some identifiers only that Service Provider can easily determine the pricing policy to be applied to the instance. As the transaction will involve authentication steps after calculating the price the TAD will be able to determine if the instance is genuine. Therefore, Service Provider may feel confident to store history record identifiers publicly on the chip to reduce transaction timing. NFC Feature Box offers "Public" access right for such needs. If index15 is assigned as a key identifier of any access right, NFC Feature Box accepts the command of such action no matter if the instance is authenticated with any key or is not authenticated at all.

On the other hand, Service Providers may want to block an access right of an action on a file. In general, it is required to increase the security of the system to ensure some value remains unchanged. To demonstrate, a service provider may issue a feature that is valid for limited usage. After each usage, the internal counter stored in a value file is decremented and the service provider wants to ensure that the value file cannot be incremented with any key or authentication. NFC Feature Box offers "Forbidden" access right for such needs. If index 0 is used as a key identifier of any access right, NFC Feature Box rejects all the commands of such action no matter if the instance is authenticated or not. Figure 4.6 shows an example of access right assignment of four files using three authentication keys and Public and Forbidden keys.

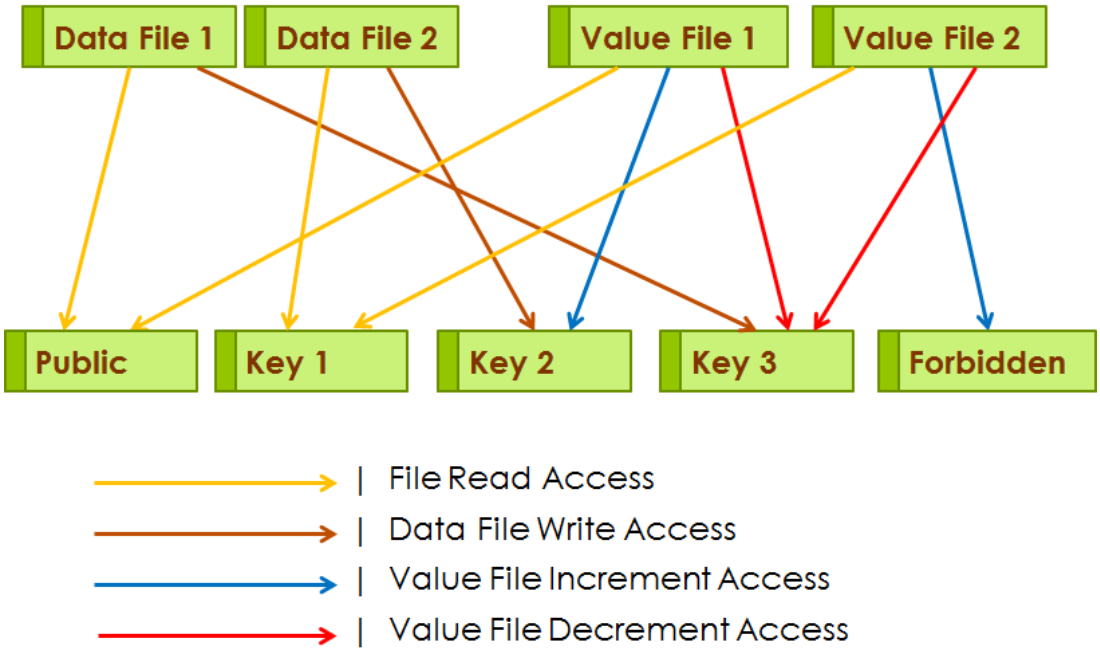

Figure 4.6. File Access Rights with reserved options

In the given example, Data File 1 and Value File 1 can be read without any authentication or after an authentication with any key. And Value File 2 can never be incremented.

4.4. Feature Creation

In order to create a Feature Instance in NFC Feature Box applet, there are three main steps. Initially, user needs to get a Registration Code from the Service Provider that will be used to map the User when the phone triggers Instance Creation. This allows Service Provider to load correct credentials on the correct phone. After establishing the registration channel, Service Provider requests Feature Instance Manager to create an instance for the SP. After a successful Instance creation, Service Provider issues Personalization commands to the created instance through the created secure channel. Feature Creation Flow is depicted in Figure 4.7.

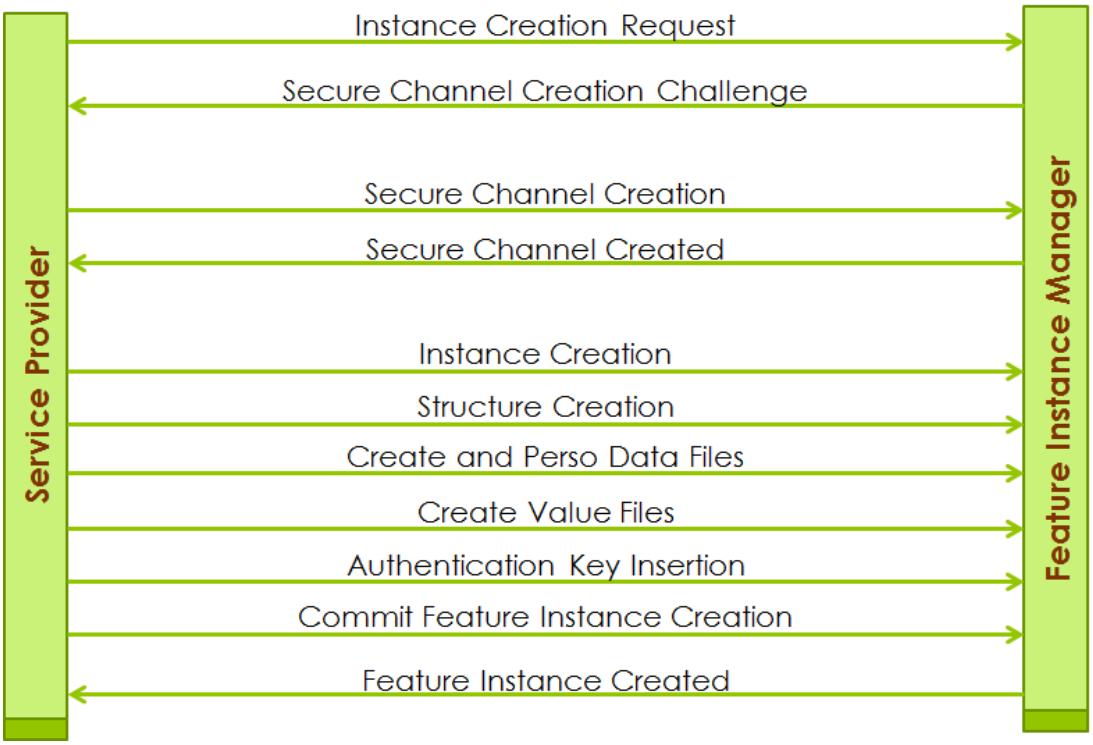

Figure 4.7. Feature Creation Flow

4.4.1. Registration Code

NFC Feature Box offers an Enabler Independent registration and usage model of NFC Mobile Payment and Identification. Having this goal achieved, there will be no trusted third party between the User Phone and the Service Provider. Thus, Service Provider needs to identify its user when it is attempting to download the feature into his/her phone. NFC Feature Box advices the usage Registration Code that is given by Service Provider to each applicant and this information is asked when the User initiates Feature Creation so that the SP can map the digital application with the actual user records in its database. Registration Code is an alphanumeric value which is exactly 8 characters long. Service Providers are advised to issue a unique and non-consecutive Registration Codes to each applicant. Providing this information during the Feature

Creation is the unique identifier of the actual user who applied for the service. Registration Code verification brings the second-factor authentication in NFC Feature Box which ensures feature installations to genuine users.

4.4.2. Feature Instance Creation

Feature Instance Creation is performed between the Service Provider and the Secure Element with the help of Phone Application (PA) that acts as a proxy between them. Instance creation routine starts from the PA where the user initiates the Feature Installation in his/her phone. PA establishes a secure connection to the Service Provider and also phone application connects to the Secure Element and establishes a connection to the FIM and the instance creation routine continuous as explained in Figure 4.8.

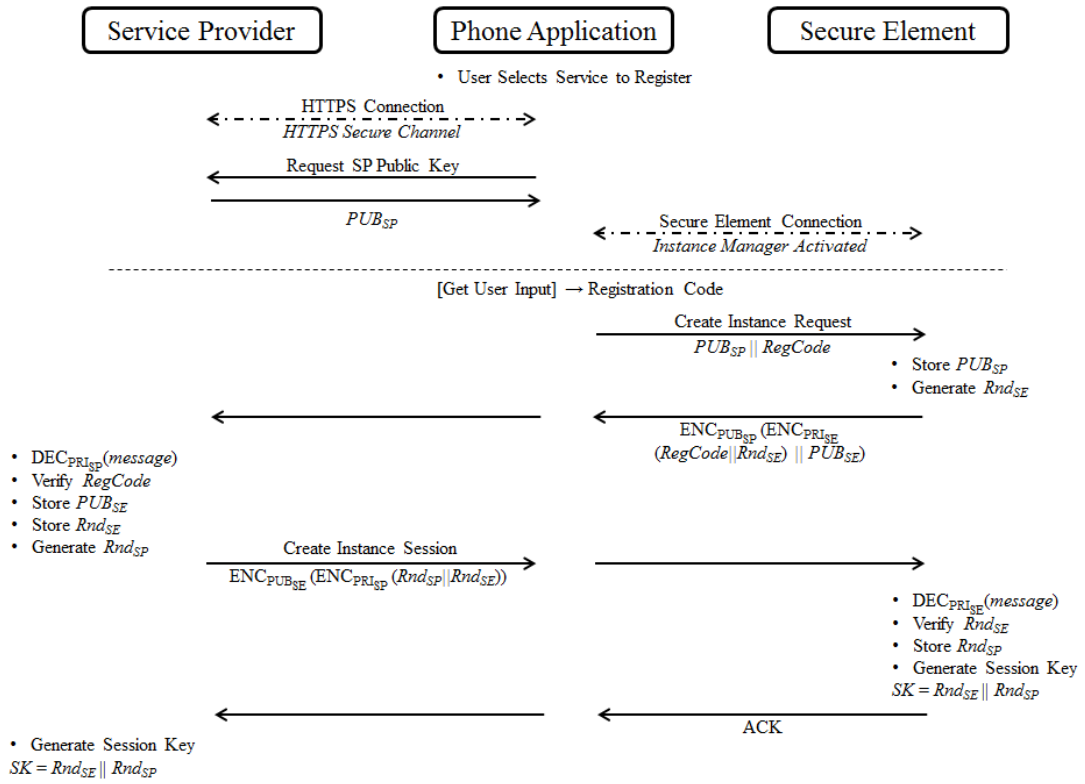

Figure 4.8. Feature Instance Creation

After creating SP and SE communication channels, PA retrieves the Public Key of Service Provider (PUB_{SP}) if it is not already stored in the PA, i.e. in SP's mobile application that is already installed on the user's phone.

The user enters Registration Code (*RegCode*) using User Interface of the phone and sends Create Instance Request Command to the SE. FIM stores the PUB_{SP} and generates a secure random with 12-bytes (Rnd_{SE}). FIM encrypts *RegCode* and Rnd_{SE} using its own private key (PRI_{SE}) and then encrypts the result with PUB_{SP}. This ensures that the message sent to SP can only be decrypted by that SP and also SP verifies the command is generated by the SE. In order to check the validity of the received message SP verifies the *RegCode* and also uses it to retrieve User information/credentials from its database. SP generates a 12-bytes secure random with (Rnd_{SP}). Secure Elements contain dedicated hardware functions to generate secure

random arrays. Therefore, it is a reliable component in session key generation between two parties. Service Provider web services are software running on the Server but generating a random array on the software is not considered to be secure (Subashini, 2011). Therefore, NFC Feature Box requires Service Providers to use dedicated hardware to perform crypto calculations and also to generate secure random arrays.

SP prepares Create Instance Session Command by encrypting Rnd_{SP} and Rnd_{SE} first with PRI_{SP} and encrypting the result with PUB_{SE}. FIM decrypts the received message using PRI_{SE} and decrypts the result with PUB_{SP}. This ensures that the prepared message can only be decrypted by the target SE and also verifies that the message is prepared by the Feature Owner SP. FIM sends the acknowledge message if received Rnd_{SE} matches internally generated Rnd_{SE}.

At this step, both FIM and SP generate the Session Key (SK) by appending Rnd_{SE} and Rnd_{SP}. The result of this concatenation is 24-byte length and it is used as 3KTDES key for further session communication encryption and decryption operations.

4.4.3. Instance Personalization

After creating a secure communication channel between the SP and the FIM, Personalization routines start. In this phase of Feature Instance Creation, SP requests the physical instance creation in the SE and fills newly created instance with User data as shown in Figure 4.9.

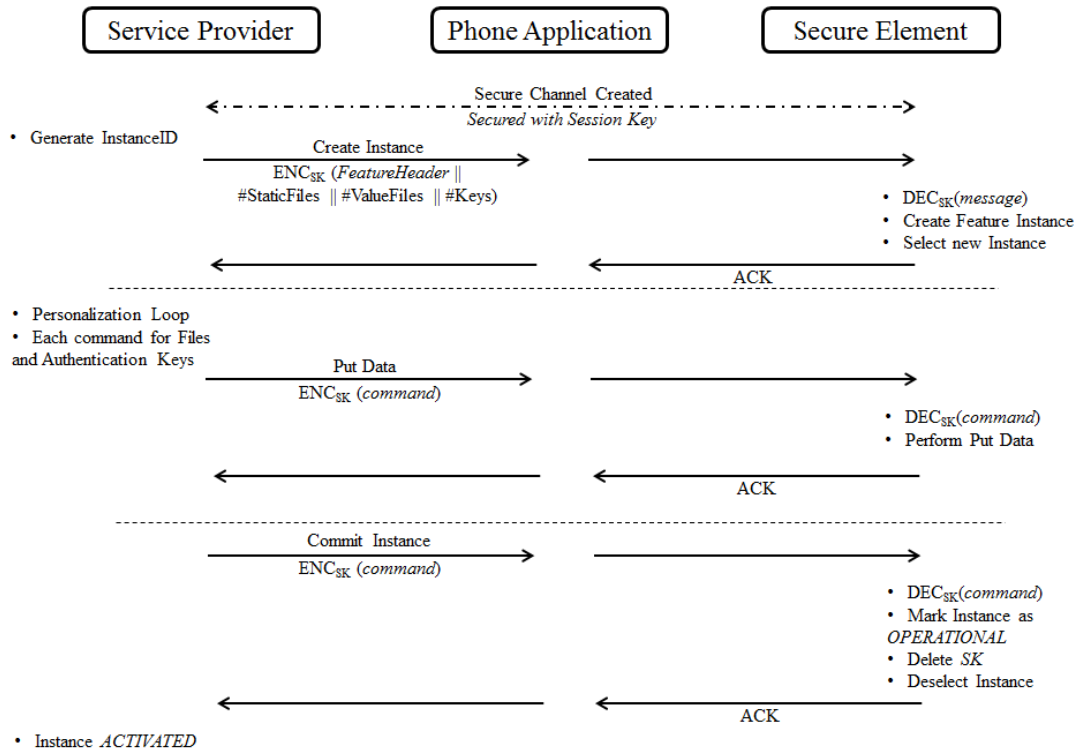

Figure 4.9. Feature Instance Personalization

Before starting personalization routines, SP internally generates a unique Instance UID for the new registration. This UID will be the unique identifier of the Feature Instance in NFC Feature Box that belongs to this Feature Owner.

SP prepares the Feature Header of the generated instance and sends Create Instance Command with Feature Header, # of Static Data Files, # of Value Files and # of Authentication Keys parameters that are encrypted with SK. FIM decrypts the message using SK, creates new Feature Instance, adds this instance to its linked list and selects newly created Feature Instance.

After receiving the Instance Creation ACK message, SP starts Put Data Command loop to create the file structure of the Feature Instance and populates User related information into files. SP uses Put Data Command also for writing Authentication Keys inside the instance. NFC Feature Box applet reserves required memory area for

each received Put Data Command and stores sent data in it. During personalization, the instance cannot be used as it does not have complete information set yet. Therefore, SP marks the completion of the personalization by issuing Commit Command. NFC Feature Box marks the newly created instance as OPERATIONAL after receiving Commit Instance Command. NFC Feature Box or FIM does not perform redundancy or validity check over the instance file structure while executing the commit. Therefore, it is under Service Provider's responsibility to make a complete personalization to ensure instance is properly created and the data is correctly populated.

4.4.4. Post Issuance

Smart cards are offline devices, this makes content update after issuance harder. A huge advantage of using mobile devices for payment and identification is remote connectivity. It gives Service Providers the flexibility to update the chip content remotely. However, existing NFC models require SE Issuers or their TSMs to access the chip and perform content updates. Thus, creates the aforementioned enabler dependency.

NFC Feature Box also offers Post Issuance function to Service Providers to allow remote content updates on the instances that were previously created. NFC Feature Box achieves this by requiring neither SE Issuer's nor TSM's involvement. During the Feature Instance Creation both SP and SE stores each other's public keys which are then used for creating a secure channel in between. Users can trigger content updates using the PA of the SP whenever a remote update is necessary. Post issuance execution flow is shown in Figure 4.10.

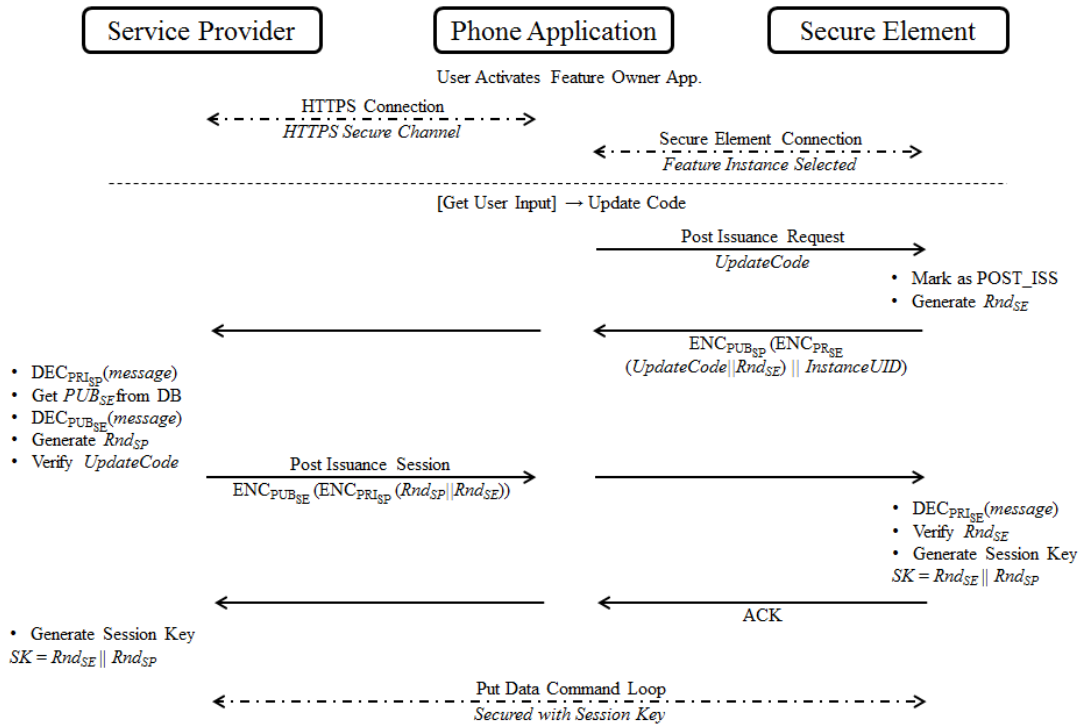

Figure 4.10. Post Issuance Flow

The PA of the SP makes an HTTPS connection to the SP and connects to the SE to select the FIM. And then selects the Feature Instance of the SP that is already available in the NFC Feature Box applet. At this state, PA asks the User to enter the Update Code which is not a mandatory step for content updates. Therefore, Update Code can be empty if a Service Provider wishes to skip the second-factor authentication for remote update as it is already capable to verify its instance using the previously agreed secure keys. PA sends Post Issuance Request Command to initiate the Post Issuance routine.

FIM marks the Feature Instance as POST_ISS and generates the Rnd_{SE} with 12-bytes. *UpdateCode* and Rnd_{SE} are encrypted with PRI_{SE} and the result is encrypted with PUB_{SP} after *InstanceUID* is concatenated to it. SP decrypts the message and uses the *InstanceUID* to retrieve PUB_{SE} from its database. Then SP decrypts the message with PUB_{SE} to reach *UpdateCode* and Rnd_{SE}.

SP prepares Post Issuance Session Command by encrypting Rnd_{SP} and Rnd_{SE} first with PRI_{SP} and encrypting the result with PUB_{SE}. FIM decrypts the received message using PRI_{SE} and decrypts the result with PUB_{SP}. This ensures that the prepared message can only be decrypted by the target SE and also verifies that the message is prepared by the Feature Owner SP. FIM sends acknowledge message if received Rnd_{SE} matches with internally generated Rnd_{SE}.

At this step, both FIM and SP generate the Session Key (SK) by appending Rnd_{SE} and Rnd_{SP}. The result of this concatenation is 24-byte length and it is used as 3KTDES key for further session communication encryption and decryption operations.

After having a successful secure communication channel, SP starts issuing Put Data Commands to make remote content update. At the end of the update, SP issues Commit Instance Command to finalize the remote update and to put the Instance into OPERATIONAL mode again.

NFC Feature Box or FIM does not perform redundancy or validity check against instance file structure while executing the commit. Therefore, it is under Service Provider's responsibility to perform a valid content update that does not affect the execution of the Instance.

4.5. Transaction Flow

A transaction with NFC Feature Box applet consists of three parts as shown in Figure 4.11. In the first part, a feature discovery is performed to determine if the NFC Feature Box applet contains the feature for this transaction. In the second part, TAD authenticates to the NFC Feature Box applet instance and this step may be repeated as many times as the transaction requires. And finally, TAD issues file related commands to perform the transaction.

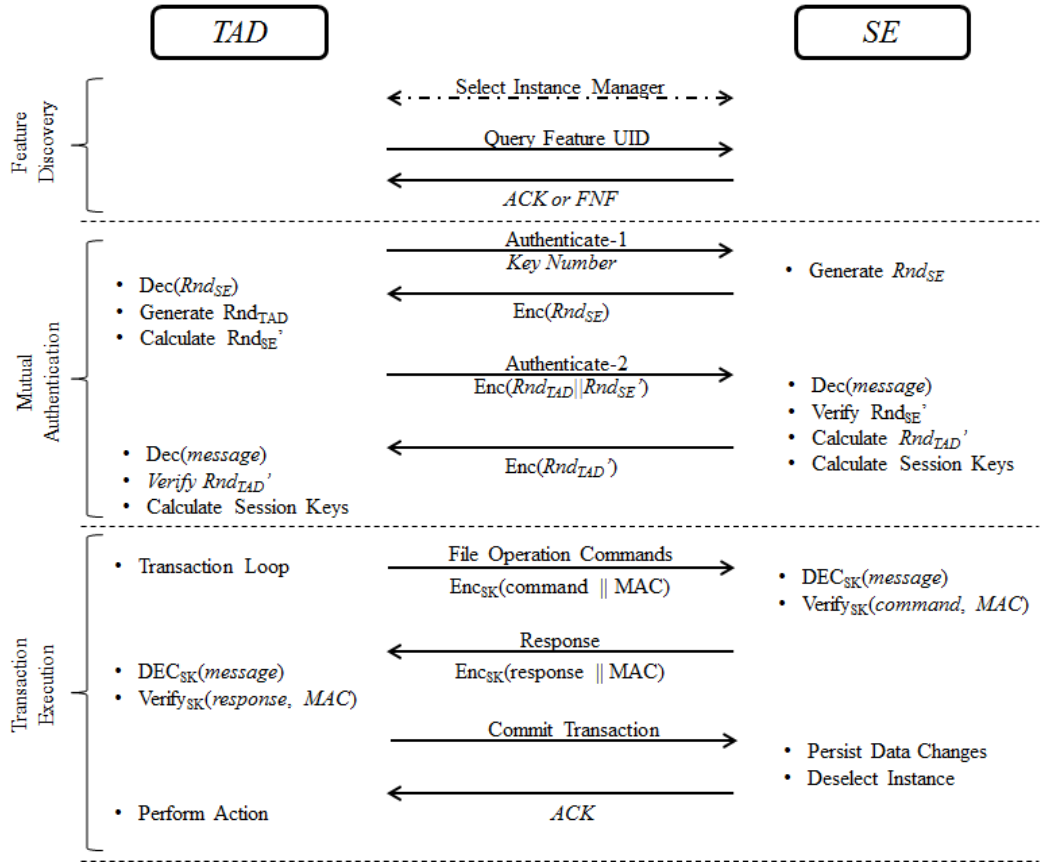

Figure 4.11. Transaction Flow

An NFC Secure Element is a multi-application platform and may contain several applications together with the NFC Feature Box applet such as credit card applets. Therefore, before starting the Transaction Execution, TAD needs to select FIM using its AID in the way it is explained in the Commands section. FIM replies to select command with FIM version number which can be used to distinguish the command protocol that might be changed with further improvements of the NFC Feature Box that is explained in this thesis.

4.5.1. Feature Discovery

After selecting the FIM, the first step of executing a transaction between the TAD and NFC Feature Box loaded NFC Phone is performing a Feature Discovery as Feature Box may contain many other Feature Instances. Each Feature Instance in NFC Feature Box has a *FeatureUID* which is the identifier of the Service Provider.

TAD sends Feature Discovery Command to the FIM with *FeatureUID* of the SP. FIM queries provided UID in the Linked List of Feature Instances. If UID is found, FIM marks the Feature Instance as SELECTED and returns the *InstanceUID* to the TAD. Otherwise, FIM returns Feature Not Found (FNF) error code.

4.5.2. Instance Authentication

NFC Feature Box implements three-pass mutual authentication mechanism to make a dual authentication between the TAD and the Feature Instance. This allows TAD to verify the genuineness of the Instance as well as allowing Instance to verify the genuineness of the TAD. This authentication is performed without sharing or transmitting the authentication key in between.

Authentication flow is initiated by the TAD by sending an Instance Authentication-1 Command. TAD sends the key index that will be used for this authentication as the parameter of the command. The Instance generates a secure random (Rnd_{SE}) using dedicated hardware of the SE and encrypts it using the Authentication Key at the index requested by the TAD. The length of Rnd_{SE} should be the same as the key length of the requested Authentication Key.

TAD decrypts the message using the same Authentication Key to access the Rnd_{SE}. And then rotates Rnd_{SE} one by left to calculate Rnd_{SE}'. TAD generates a secure random (Rnd_{TAD}) which has the same length with Rnd_{SE}. TAD concatenates these values and encrypts with the Authentication Key to send the Instance Authentication-2 Command.

The Instance decrypts the message using the Authentication Key and verifies Rnd_{SE}'. If it is valid, it means the TAD has the correct Authentication Key so it was able to decrypt the message and could rotate the Rnd_{SE}. The Instance calculates Rnd_{TAD}' in the same way the Rnd_{SE}' is calculated by rotating the array and replies to the command. TAD decrypts the received message and verifies Rnd_{TAD}'. If it is valid, it means the Instance is valid and has the same Authentication Key so it was able to decrypt the message and could rotate the Rnd_{TAD}.

After successful execution of authentication commands, both entities calculate two session keys called Session Encryption Key and Session MAC Key as shown in Figure 4.12. Lengths of Session Keys are depending on the length of the Authentication Key that is used for this authentication.

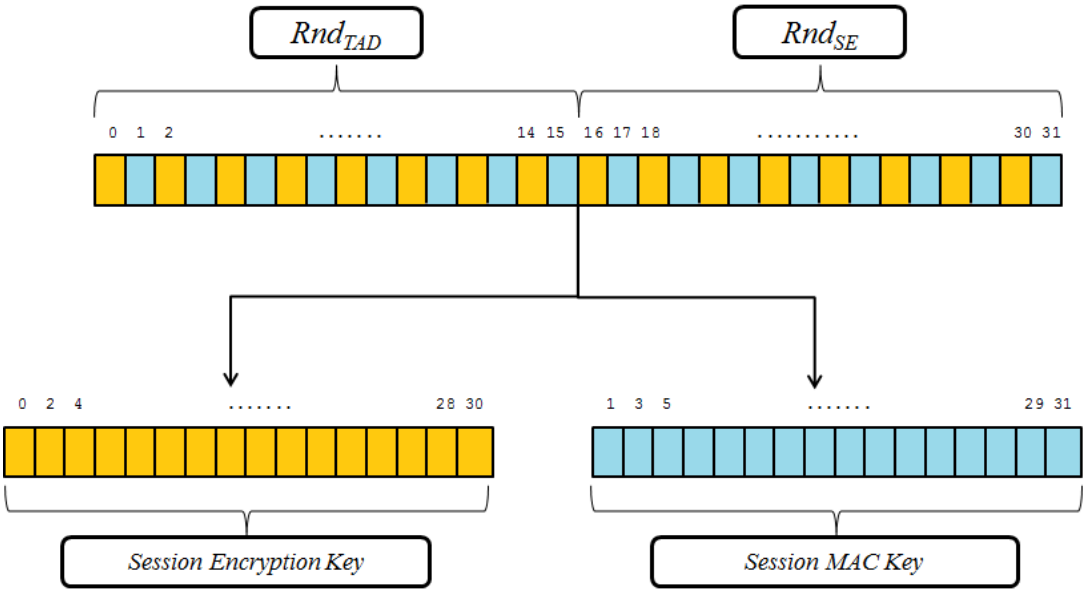

Figure 4.12. Session Key Generation

In the given example, 2KTDES or AES128 authentication usage is shown as their key length is 16 bytes. After successful authentication, Rnd_{SE} and Rnd_{TAD} are concatenated.

Even indexed bytes are used to build the Session Encryption Key and odd indexed bytes are used to build the Session MAC Key.

Session Encryption Key is used to encrypt messages transmitted between TAD and SE to ensure confidentiality. And Session MAC Key is used to calculate cryptographic MAC of the command to ensure integrity.

In order to encrypt the command data with Session Encryption Key, data needs to be padded with a trailer to have the input data length multiple of the key length. ISO 9797-1 padding is applied for this purpose (ISO9797, 2011).

Cryptographic MAC calculation of NFC Feature Box is an encryption operation with the Session MAC Key that is applied to the command header and the command data to ensure full integrity of the command. Cryptographic MAC of the command is calculated according to the NIST Special Publication 800-38B (Dworkin, 2005). In order to reduce the command length, the result of Cryptographic MAC calculation is truncated to 8 bytes. If authentication key length is 16 bytes then even indexed bytes of the resulting array is used to reach 8-byte length. If authentication key length is 24 bytes then the bytes at the index of multiples of three are used to reach 8-byte length. The MAC is added to the command data before applying the command encryption.

In a typical offline transaction system, TAD is equipped with a Secure Access Module (SAM) card which is proven to store authentication keys securely. NFC Feature Box advices the usage of SAM at TAD devices not only for storing the keys securely but also for generating secure random numbers and handling the Session Keys on secure hardware.

4.5.3. Transaction Execution

Transaction Execution step in NFC Feature Box is a combination of Authentication and File Operation Commands depending on the transaction flow of the required action. A typical payment application may require several authentications, file read

and write operations and also may require value file operations if the User balance is stored in the chip. On the contrary, typical door access may only require a single authentication and file read operation.

4.6. Command and Response Structures

NFC Feature Box applet communication between TAD and PA is based on ISO7816-4 APDU frame format as shown in Figure 2.14. NFC Feature Box applet does not check the CLA byte of the command APDU. Thus, any value can be assigned to CLA byte in the command sent to the applet. 0x80 is used as the CLA value in sample commands provided in this thesis.

List of Instruction and Parameter byte values of NFC Feature Box applet is given in Table 4.2.

Table 4.2. *Command Parameter List*

Command	*INS*	*P1*	*P2*
Create Instance Request	0x61	0x01	0x00
Create Instance Session	0x61	0x02	0x00
Create Instance	0x62	<CI >	<LCI>
Post Issuance Request	0x71	0x01	0x00
Post Issuance Session	0x71	0x02	0x00
Commit Instance	0xC1	0x00	0x00
Delete Instance	0xDD	0x00	0x00
Put Data	0xDA	<Field Index>	0x00
Feature Discovery	0xFD	0x00	0x00
Instance Authentication-1	0xA1	<Key Index>	0x00
Instance Authentication-2	0xA2	0x00	0x00
Read Static Data File	0x91	<File Index>	0x00
Write Static Data File	0x92	<File Index>	0x00
Read Value File	0x41	<File Index>	0x00
Increment Value File	0x42	<File Index>	0x00

Decrement Value File	0x43	<File Index>	0x00
Commit Transaction	0xC2	0x00	0x00

Details of the command data and response structure are explained in the following sections.

4.6.1. Select Commands

NFC Feature Box applet requires Global Platform communication, therefore, the applet selection command will be performed according to Global Platform specifications using the AID of the applet.

Table 4.3. *Select Command*

Attribute	*Value*
Command Header	0x00 0xA4 0x04 0x00
CDATA	Applet AID: 46 65 61 74 75 72 65 42 6f 78 31
Error Codes	0x9000 – Success
Response Data	<Applet State Indicator><Version>

Applet State Indicator byte that is returned in Response Data represents that Applet Lifecycle of NFC Feature Box as shown below;

- 0x01 – INSTALLED
- 0x02 – PERSONALIZATION
- 0x03 – POST_ISSUANCE
- 0x04 – OPERATIONAL
- 0x05 – INSTANCE_SELECTED

NFC Feature Box applet returns two-byte version information in response to select command to allow early detection of the applet version during transaction execution.

This selection command will activate the FIM in the NFC Feature Box applet which will then dispatch all the following commands. FIM internally maintains the selected Instance when TAD or PA is communicating directly to the Instance. Dedicated Feature Instance selection is performed automatically during the Feature Discovery. Thus, GP select command is not used for Instance selection.

4.6.2. Instance Structure Commands

Below commands are executed by FIM that are used to alter the Instance Structure and Instance Header.

Table 4.4. *Create Instance Request*

Attribute	*Value*
Command Header	0x80 0x61 0x01 0x00
CDATA	TLV coded tags: 0x01 <length> <PUB_{SP}> 0x02 0x08 <*RegCode*>
Error Codes	0x6A86 – If P1 is not valid 0x6700 – If the Tag lengths are not valid 0x6984 – If the Public Key format is not valid 0x9000 – Success
Response Data	Encrypted text

Table 4.5. *Create Instance Session*

Attribute	*Value*
Command Header	0x80 0x61 0x02 0x00
CDATA	Encrypted text

Error Codes	0x6A86 – If P1 is not valid 0x6700 – If the text length is not valid 0x6985 – If Rnd_{SE} cannot be verified 0x9000 – Success
Response Data	No Data

Table 4.6. *Create Instance*

Attribute	*Value*
Command Header	0x80 0x62 <CI> >LCI>
CDATA	TLV coded tags: 0x01 0x04 <*FeatureUID*> 0x02 0x04 <*InstanceUID*> 0x03 0x02 <Version> 0x04 <length> <Feature Info> 0x05 <length> <Public Info> 0x06 0x01 <#of Data Files> 0x07 0x01 <#of Value Files> 0x08 0x01 <#of Authentication Keys> All the data is encrypted with Session Key
Error Codes	0x6A86 – If Parameter bytes are not valid 0x6700 – If the Tag lengths are not valid 0x6A80 – If the Tag Values invalid or out of range 0x6982 – If the message format is not valid 0x6985 – If not authenticated 0x9000 – Success
Response Data	No Data

Parameters bytes of Create Instance command is used to manage the chaining of this command. Feature header is long enough to not fit in a single command. Therefore, proprietary command chaining is applied to transmit all data to the SE. NFC Feature Box stores retrieved data until the last block is sent and then executes the command.

Command data is divided into blocks each having 250 bytes at maximum. Each block is assigned with the Chain Index (CI) number sequentially starting from 0. P2 parameter is used to indicate chaining is continuing. Last Chain Indicator (LCI) value indicates if the received command is the last command of the chain or not. 0xFF indicates the last block. All other values indicate that there are more blocks to follow.

Table 4.7. *Post Issuance Request*

Attribute	*Value*
Command Header	0x80 0x71 0x01 0x00
CDATA	<*UpdateCode*>
Error Codes	0x6A86 – If P1 is not valid 0x9000 – Success
Response Data	Encrypted text

Table 4.8. *Post Issuance Session*

Attribute	*Value*
Command Header	0x80 0x71 0x02 0x00
CDATA	Encrypted text
Error Codes	0x6A86 – If P1 is not valid 0x6700 – If the text length is not valid 0x6985 – If Rnd_{SE} cannot be verified 0x9000 – Success
Response Data	No Data

Table 4.9. *Commit Issuance*

Attribute	*Value*
Command Header	0x80 0xC1 0x00 0x00
CDATA	No Data
Error Codes	0x6883 – If Create Instance chaining not completed 0x6985 – If not authenticated 0x9000 – Success
Response Data	No Data

Table 4.10. *Delete Issuance*

Attribute	*Value*
Command Header	0x80 0xDD 0x00 0x00
CDATA	No Data
Error Codes	0x6985 – If not authenticated 0x9000 – Success
Response Data	No Data

Put Data Command can be used for several purposes, the command structure is the same for all but the command data is slightly different for each purpose. Therefore, each Put Data command is explained separately.

Table 4.11. *Put Data for Static Data File*

Attribute	*Value*
Command Header	0x80 0xDA **0x01** <*File Index*>
CDATA	TLV coded tags: 0x01 0x02 <*length of the file* > 0x02 0x01 <*Read Access Key Index*> 0x03 0x01 <*Write Access Key Index*> All the data is encrypted with Session Key

Error Codes	0x6A86 – If Parameters are not valid 0x6700 – If the Tag lengths are not valid 0x6984 – If the index values are not valid 0x6985 – If not authenticated 0x9000 – Success
Response Data	No Data

Table 4.12. *Put Data for Value File*

Attribute	*Value*
Command Header	0x80 0xDA **0x02** <*File Index*>
CDATA	TLV coded tags: 0x01 0x04 <*Initial value of the file* > 0x02 0x04 <*Allowed Min value of the file* > 0x03 0x04 <*Allowed Max value of the file* > 0x04 0x01 <*Read Access Key Index*> 0x05 0x01 <*Increment Access Key Index*> 0x06 0x01 <*Decrement Access Key Index*> All the data is encrypted with Session Key
Error Codes	0x6A86 – If Parameters are not valid 0x6700 – If the Tag lengths are not valid 0x6984 – If the index values are not valid 0x6985 – If not authenticated 0x9000 – Success
Response Data	No Data

Table 4.13. *Put Data for Authentication Key*

Attribute	*Value*
Command Header	0x80 0xDA **0x03** <*File Index*>

CDATA	TLV coded tags: 0x01 0x01 <*Key Type*> 0x02 <length> <*Key Value* > All the data is encrypted with Session Key
Error Codes	0x6A86 – If Parameters are not valid 0x6700 – If the Tag lengths are not valid 0x6A80 – If the Key Type is not valid 0x6985 – If not authenticated 0x9000 – Success
Response Data	No Data

The Key Type tag in the Put Data command for Authentication Key is encoded according to the below list;

- 0x01 – 2KTDES
- 0x02 – 3KTDES
- 0x03 – AES128
- 0x04 – AES192

Table 4.14. *Put Data for Public Info (Post Issuance)*

Attribute	*Value*
Command Header	0x80 0xDA **0x04** 0x00
CDATA	Public Info encrypted with Session Key
Error Codes	0x6A86 – If P1 is not valid 0x6700 – If the text length is not valid 0x6985 – If not authenticated 0x9000 – Success
Response Data	No Data

Table 4.15. *Put Data for Feature Info (Post Issuance)*

Attribute	*Value*
Command Header	0x80 0xDA **0x05** 0x00
CDATA	Feature Info encrypted with Session Key
Error Codes	0x6A86 – If P1 is not valid 0x6700 – If the text length is not valid 0x6985 – If not authenticated 0x9000 – Success
Response Data	No Data

Table 4.16. *Put Data for Version (Post Issuance)*

Attribute	*Value*
Command Header	0x80 0xDA **0x06** 0x00
CDATA	New Version encrypted with Session Key
Error Codes	0x6A86 – If P1 is not valid 0x6700 – If the text length is not valid 0x6985 – If not authenticated 0x9000 – Success
Response Data	No Data

Table 4.17. *Feature Discovery*

Attribute	*Value*
Command Header	0x80 0xFD 0x00 0x00
CDATA	Feature UID
Error Codes	0x6A82 – If Feature does not exist 0x9000 – Success
Response Data	Instance UID

4.6.3. Instance Authentication

Instance authentication is performed with two commands as explained in the previous sections.

Table 4.18. *Instance Authentication-1*

Attribute	*Value*
Command Header	0x80 0xA1 <*Key Index*> 0x00
CDATA	No Data
Error Codes	0x6A86 – If Key Index is not valid 0x9000 – Success
Response Data	Encrypted Text

Table 4.19. *Instance Authentication-2*

Attribute	*Value*
Command Header	0x80 0xA2 0x00 0x00
CDATA	Encrypted Text
Error Codes	0x6700 – If the message length is not valid 0x6985 – If Rnd_{SE} cannot be verified 0x9000 – Success
Response Data	Encrypted Text

4.6.4. File Structure Commands

NFC Feature Box offers two types of files which are explained in the previous sections. Below commands are used for data access/update.

Table 4.20. *Read Static Data File*

Attribute	*Value*
Command Header	0x80 0x91 <*File Index*> 0x00
CDATA	<*Start Index*> <*# of bytes to read*> <MAC>
Error Codes	0x6A86 – If Parameters are not valid 0x6700 – If the Index values are not valid 0x6982 – If Security conditions not satisfied 0x6985 – If the MAC is not valid 0x9000 – Success
Response Data	Requested Data and the Data MAC is encrypted with Session Key

Table 4.21. *Read Value File*

Attribute	*Value*
Command Header	0x80 0x41 <*File Index*> 0x00
CDATA	<MAC>
Error Codes	0x6A86 – If Parameters are not valid 0x6982 – If Security conditions not satisfied 0x6985 – If the MAC is not valid 0x9000 – Success
Response Data	The Value of the file and the MAC is encrypted with Session Key

Table 4.22. *Write Static Data File*

Attribute	*Value*
Command Header	0x80 0x92 <*File Index*> 0x00
CDATA	<*Start Index*> <*End Index*> <data> <MAC> All the command data encrypted with Session Key

Error Codes	0x6A86 – If Parameters are not valid 0x6700 – If the Index values are not valid 0x6982 – If Security conditions not satisfied 0x6985 – If the MAC is not valid 0x9000 – Success
Response Data	No Data

Table 4.23. *Increment Value File*

Attribute	*Value*
Command Header	0x80 0x42 <*File Index*> 0x00
CDATA	<*Value*> <MAC> All the command data encrypted with Session Key
Error Codes	0x6A86 – If Parameters are not valid 0x6A82 – If the Value is not valid 0x6982 – If Security conditions not satisfied 0x6985 – If the MAC is not valid 0x6A80 – If increment exceeds allowed max value 0x9000 – Success
Response Data	New Value of the file and the MAC is encrypted with Session Key

Table 4.24. *Decrement Value File*

Attribute	*Value*
Command Header	0x80 0x43 <*File Index*> 0x00
CDATA	<*Value*> <MAC> All the command data encrypted with Session Key

Error Codes	0x6A86 – If Parameters are not valid 0x6A82 – If the Value is not valid 0x6982 – If Security conditions not satisfied 0x6985 – If the MAC is not valid 0x6A80 – If decrement result is lower than allowed min value 0x9000 – Success
Response Data	New Value of the file and the MAC is encrypted with Session Key

Table 4.25. *Commit Transaction*

Attribute	*Value*
Command Header	0x80 0xC2 0x00 0x00
CDATA	No Data
Error Codes	0x6982 – If there is no change to committed 0x6985 – If not authenticated 0x9000 – Success
Response Data	No Data

4.7. Service Provider Requirements

NFC Feature Box is designed for Service Providers who want to accept NFC Phones in their existing systems in tandem with contactless cards. NFC Feature Box allows Service Providers to implement the system requirements on their own without requiring NFC Enabler involvement. As a result, NFC Feature Box requires Service Providers to implement the Web Services that will serve for the Feature Instance lifecycle management. Additionally, Service Providers are requested to update their existing Transaction Acceptance Device network to accept NFC Phones in their system.

This thesis explains in detail the transaction execution flow and feature registration flow. This provides the required guidelines for the Web Service implementation. However, Web Service implementation details are out of scope for this thesis.

According to the aforementioned facts of using hardware components for secure key handling and random number generation, we strongly recommend Service Providers to use Hardware Security Modules (HSM) in their systems.

CHAPTER 5

EVALUATION AND EXPERIMENTS

5.1. Security Analysis

Security is an important requirement in many systems. However, it plays a crucial role especially in payment systems as a non-secure system may result in financial loss. NFC Feature Box offers secure transaction flow based on the hardware components and by the secure protocol design. Here, we discuss the physical and logical attacks that our system might face and then we explain how NFC Feature Box is protected against those attacks.

5.1.1. Hardware Attack Resistance

NFC Feature Box protocol is based on three main entities that have access to the authentication keys; Service Provider, Transaction Acceptance Device and the NFC Phone. NFC Feature Box applet on the NFC Phone is installed in the Secure Element which is dedicated secure hardware. Since storing the keys on dedicated hardware is the only way to ensure key confidentiality, we advise storing authentication keys on dedicated secure hardware for TAD and SP as shown in Figure 5.1.

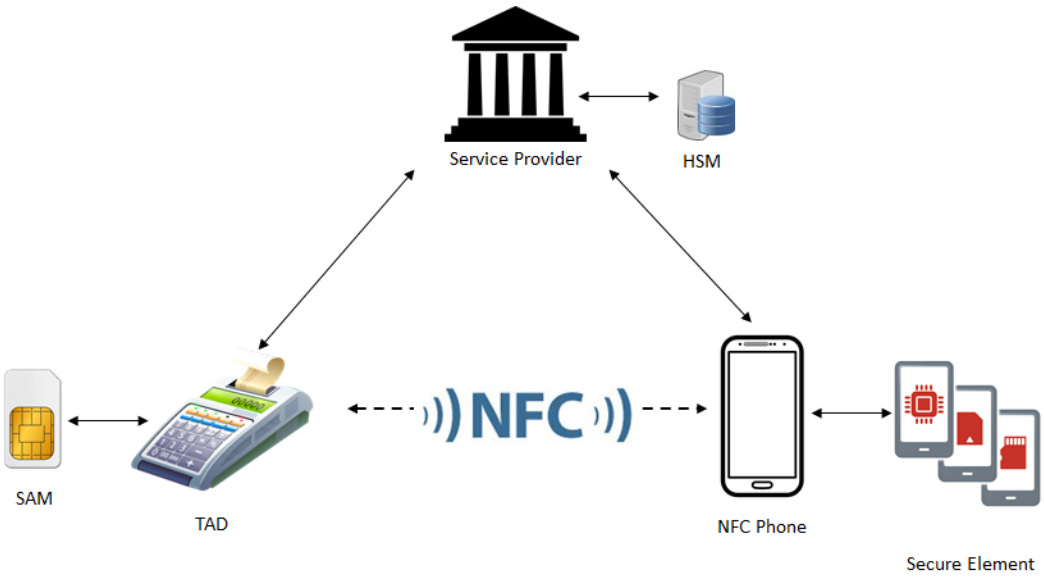

Figure 5.1. Hardware Usage for Key Handling

The usage of SAM card and HSM will ensure the Hardware Attack Resistance of the system from the Security Key confidentiality perspective.

5.1.2. Communication Channel Attack Resistance

As it is shown in Figure 5.1, there are three communication channels between three entities of the system; Service Provider to NFC SE, Service Provider to TAD and TAD to NFC SE.

5.1.2.1. Service Provider – NFC SE

The communication channel between the SP and the SE is established through the phone application. However, phone application plays an agnostic transceiver role and therefore has no access to the data transmitted between these entities. The communication channel between SP and the phone application is secured by the HTTPS connection in order to block man-in-the-middle attacks. The data transmitted between SP and SE is secured by the NFC Feature Box protocol which incorporates PKI infrastructure. NFC Feature Box uses double encryption of messages using ECC

keys of the entities which ensure confidentiality and integrity at the same time. Thus, the communication channel between SP and SE is considered to be secure and it can easily be improved with additional network layer authentication techniques (Lu, 2016). Possible logical attacks on this channel are explained in the following subsections.

5.1.2.2. Service Provider – TAD

Transaction Acceptance Devices are managed by their Service Provider which already has a proprietary communication channel to communicate. NFC Feature Box doesn't add an additional layer to that communication and therefore, it is under Service Provider's responsibility to maintain secure communication on that channel.

5.1.2.3. TAD – NFC SE

The communication channel between TAD and SE is created based on ISO14443 as explained in previous sections. ISO14443 protocol does not incorporate security by design as this protocol is not only targeting secure applications. Therefore, the confidentiality and integrity of the transmitted data over this channel are ensured at the application layer. NFC Feature Box offers three-pass mutual authentication mechanism in which the TAD and the SE can authenticate each other without sharing the key over the contactless interface. Each authentication creates two session keys which are based on the secure random arrays that are created during the authentication. As a result, NFC Feature Box authentication results in different session keys after each successful authentication. Further communication between TAD and SE is encrypted by the Session Encryption Key which ensures the confidentiality and also each file operation command contains the cryptographic MAC which ensures the integrity of the received message.

TAD and NFC Phone is communicating over the contactless interface and so communication can easily be sniffed. However, as per NFC Feature Box authentication design, copied data cannot be used to reverse engineer accessing authentication keys. Additionally, mutual authentication between TAD and NFC

Phone can be improved by additional layers (Ceipidor, 2012) to improve authentication security.

5.1.3. Logical Attacks

An attacker may try logical attack factors to gain an advantage over NFC Feature Box enabled systems. We list below the dishonest parties that might be attacking the system.

5.1.3.1. Dishonest Phone Application

A phone application in NFC Feature Box has no visibility over the messages transmitted between SP and the SE. The message confidentiality and integrity are ensured end-to-end by the communication between the SP and the SE. Therefore, a dishonest phone application can only damage the system by not delivering the message between entities which can easily be recognized by the User.

5.1.3.2. Dishonest SE application

A proprietary SE application that is trying to act as NFC Feature Box applet can be considered as a dishonest SE application. Such an application can try registering to SP service in order to get the SP keys that are sent into an Instance. *RegCode* in NFC Feature Box design brings the second factor authentication therefore; this attack is not possible without knowing the *RegCode* that SP assigned to genuine Users.

In addition to the two-factor authentication mechanism, SP is advised to use the Key Diversification mechanism to issue different keys to each Instance they have so that a key leakage may result in key leakage of that specific instance only.

5.1.3.3. Dishonest TAD application

A dishonest TAD application cannot have the authentication keys that are used in transaction authentication, therefore, they cannot act to gain an advantage.

5.1.3.4. Dishonest SP application

A dishonest SP application can try forwarding SE communication to their services in order to change Instance data for their purpose. SP and SE communication is double encrypted based on PKI infrastructure so that a dishonest SP application cannot authenticate to SE.

5.2. Performance Analysis

Performance figures for a payment or identification system is an important acceptance criterion in the industry. Especially in crowded cities, long transaction timing may result in long queues at the place where payment happens. Therefore, system owners require tight transaction timing limits to avoid queues. NFC Feature Box approach that is explained in this thesis mainly focuses on the registration and usage protocol that allows enabling independent technology usage. Therefore, Service Provider's existing transaction execution flow that is already running in TAD devices will not be affected by enabling NFC Phone acceptance. The only difference will be adding a decision tree at the beginning of a transaction to determine if the used instrument is a contactless card or an NFC Phone. Here in this section, we discuss the performance figures of enabling NFC phone acceptance at TAD devices.

A Service Provider will need to add the following steps in their existing TAD network to start accepting NFC Phone as a payment or identification media in their infrastructure;

5.2.1. Determine Transaction Media Type

In a regular transaction, TAD immediately starts selecting its target application in the chip if the transaction media is a multi-application platform such as Java Card. Application selection is performed using the AID of the application. Exiting the application of the SP will have a different AID than the AID of NFC Feature Box

applet. Therefore, TAD will issue its AID selection initially. NFC Phone SE will return File Not Found error to the issued AID selection. Then, TAD will issue NFC Feature Box applet AID selection which will return success.

As a result, enabling NFC phone acceptance in a system that is already working with multi-application transaction media will bring additional AID selection (T_{Select}) overhead.

If the existing infrastructure of the SP works with contactless memory cards such as MiFare™, then the TAD is able to determine the transaction media type at ISO 14443 Anti-collision loop using the SAK value of the chip. Contactless memory cards and multi-application chip platforms such as SE are giving different SAK values which can be used to distinguish these chip platforms. Therefore, TAD is able to determine the transaction media type at ISO14443 level without having a performance cost.

As a result, enabling NFC phone acceptance in a system that is working with contactless memory cards has no performance overhead.

5.2.2. Feature Discovery

NFC Feature Box applet requires the TAD to select its target Feature Instance that is available under FIM linked list. Performance figures of Feature Discovery command are similar to AID selection in multi-application chip platforms as Feature Discovery also requires getting the UID of the Instance and querying it internally. The Feature Discovery step is mandatory after the FIM selection.

An SP can choose to accept NFC Phones only in their system through NFC Feature Box. In this case, SP will immediately select NFC Feature Box Applet AID when a transaction media is detected by the TAD. Therefore, there will not be transaction media type detection. However, as NFC Feature Box requires Feature Discovery as a mandatory step, SP will have Feature Discovery performance overhead compared to any other proprietary transaction system that SP can implement.

The below table summarizes the performance effect of enabling NFC Phone acceptance in an existing system.

Table 5.1. *Performance Effect*

Infrastructure	*Determine Type*	*Feature Discovery*	*Total*
Existing Multi-Application Infrastructure	T_{Select}	T_{Select}	2x T_{Select}
Existing Memory Card Infrastructure	0	T_{Select}	T_{Select}
New Multi-Application Implementation	0	T_{Select}	T_{Select}

T_{Select} execution time depends on the chip platform but in general, it is less than 10 ms for all chip platforms available in the market and a typical payment system requires more than 400 ms for the whole transaction. Therefore, adding NFC Phone acceptance by NFC Feature Box does not affect the overall performance of the transaction.

5.3. Experimental Setup

In order to prove the practical applicability of NFC Feature Box in real life offline payment scenarios, we had implemented the following applications.

5.3.1. NFC Feature Box Applet

We implemented our NFC Feature Box Applet on JCOP Chip Operating System. It has the root Feature Instance Manager (FIM) and keeps a linked list for the Feature Instances as explained in this thesis. We deployed our Chip Application onto the Secure Element of a Samsung Galaxy S4 developer phone.

5.3.2. Service Provider Application

We implemented a Java Application that accepts HTTP connections from clients and implements our protocol's features such as generating the Registration Code, Secure Channel Creation, Instance Creation and Instance Personalization.

5.3.3. Payment Application

In order to show that our protocol dynamically creates instances and those instances can be used in real schemes, we developed a payment terminal application that communicates to our Feature Instance Manager, queries for available Feature Instances and performs a sample offline payment transaction using the value file within Feature Instances. A sample screen of the application is shown in Figure 5.2. Payment Application creates the transaction after selecting the target Service Instance and deducting the transaction amount from the available offline credit amount in the Instance.

Figure 5.2. Payment Application Screen

5.3.4. Phone Application

We implemented an Android Application for our Samsung Galaxy S4 developer phone that communicates to our Java-based Service Provider application and also

communicates to the Secure Element inside the phone. It has a basic Graphical User Interface to show available Service Instances inside the Chip and also shows the Instance Data of each Instance. An example application screen is shown in Figure 5.3, application lists available Service Instances and shows Instance details of each Instance like Instance UID, Instance Info and available offline balance.

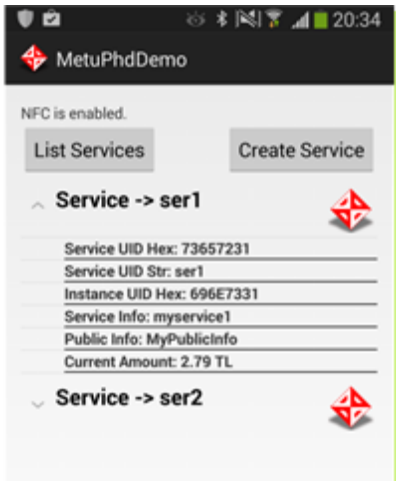

Figure 5.3. Phone Application Screen

Our experimental works have verified that our registration protocol works as targeted without requiring NFC Enabler involvement, and we could create an Instance in our NFC Phone. An offline proprietary payment scheme in our experimental work was just to show the flexibility of our design. It is possible to enhance this usage with generic smart card application use cases.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Using a mobile handset for various types of mobile transactions has been gaining tremendous popularity during the last decade. The availability of NFC technology within our mobile devices had played an active accelerator role in this traction. NFC technology brought secure transaction capability to a mobile phone which then boosted the financial transaction adaption into the mobile payment industry. Today, millions of people are using their NFC-enabled mobile handsets to perform payment transactions. This resulted in creating issuance and usage standards for credit card payment schemes. However, as early adopters of this technology, Mobile Network Operators and Mobile Handset Manufacturers played a leading role in adapting this technology according to their benefits. The resulting design of NFC-based mobile transactions forcing Service Providers to comply with rules set by those enablers and putting a large barrier in front of small businesses and organizations as this compliance requires a tremendous effort that is not affordable by the majority. Therefore, besides regular credit card payment scenarios no other NFC transaction could gain widespread adoption.

This thesis study aimed at solving NFC Enabler dependency to allow all players in this field to contribute to the improvement of NFC mobile transactions by adding services without depending on any entity. This goal is achieved for both online and offline physical payment schemes as they had been described in previous chapters.

6.1. Contributions

Throughout our studies we had invented two major protocols as a solution to the NFC Enabler dependency problem in the NFC Transactions field; RONFC solves the enabler dependency problem in online physical mobile transaction field and NFC

Feature Box solves enabler dependency problem in offline physical mobile transaction field. Contributions of these two studies in mobile transaction industry are as follows:

6.1.1. RONFC

- NFC-Enabler independent: RONFC gives full control to the users for mobile transactions by just requiring the use of the corresponding mobile application on NFC-enabled devices.
- No application limitation: In regular NFC, the SE on the NFC mobile device – whether SIM-based or embedded – has limited space for the chip applets, which can typically hold only a few. For this reason, an NFC mobile device owner can only use the device for a limited number of applications. However, with RONFC, the NFC mobile device can enable transactions through mobile applications and as a result, offer many more NFC-based applications.
- No card emulation issues: The Card Emulation Mode of NFC is an optional feature in NFC specifications; therefore, some of the NFC mobile devices do not feature card emulation at all. Consequently, creating an NFC-based transaction scheme based on a regular NFC SE results in system malfunctioning for some devices, even if they have NFC capability. However, the RONFC approach works with all the NFC-enabled mobile devices, as the Reader Mode is mandatory within NFC specifications.
- No TSM required: In the RONFC approach, the Card Providers can enable the NFC transactions themselves, simply by updating their mobile application. Also, the Terminal Providers have the ability to accept NFC transactions independently, simply by updating their terminals. As a result of this, to enable NFC transactions, NFC Enabler and their TSMs are not needed.
- Interoperability: Regular NFC transactions require proprietary integration schemes for each NFC Enabler or their TSMs. And there is no interoperability between the specs of each enabler. However, within our open protocol design,

service (e.g. payment) providers only need to integrate with the Central Authority, which ensures the connection to the worldwide NFC ecosystem.

- On-device authentication: Entering PINs or other authentication means on an unknown device is a security concern for most people, whereas our design offers the possibility to authenticate users on their mobile device. Moreover, multiple authentication methods that are available on mobile devices (such as face recognition, fingerprint scanning, etc.) can be supported.
- No entry barrier: Enabling NFC transactions in their system is the same for a world-wide tech giant and a local boutique service provider.
- No remote issuance: For a Service Provider who already has its mobile application in use by its customer base, enabling NFC transactions through RONFC doesn't require issuing anything into the handset chipset of the user. It can be enabled by just updating the mobile application which implements the RONFC protocol.

6.1.2. NFC Feature Box

- NFC-Enabler independent: NFC Feature Box allows all service providers to offer remote feature instance issuance into the mobile handset of the user without requiring the NFC enabler to take a role in this flow.
- Efficient memory usage: Secure Elements have limited space to hold chip applications although they are multi-application platforms. Therefore, a Secure Element is limited to carry a few active applications at a time. This is mainly because each application has implementation for lifecycle management, secure authentication and data management. With NFC Feature Box design, Secure Element holds one application that acts as a container for features and lifecycle management, authentication handling and communication management are provided by FIM which is common for all. Therefore, when a new feature added to NFC Feature Box application's storage footprint in Secure Element

only increases by the size of files and keys allocated for that feature. Therefore, NFC Feature Box allows an NFC phone to hold more applications than legacy NFC phone applications.

- Personalized Application Set: NFC Handsets are manufactured by worldwide suppliers and used by every person in the world. However, NFC transaction needs at each geography differ from each other considering local services. To illustrate, a user in city A would prefer having a public transportation payment application inside the NFC handset, whereas another user in city B may prefer having a vending machine payment application in the NFC handset. NFC Feature Box allows users to create their own feature set in their NFC handset by opting in the services provided in their region.
- Dynamic Memory Allocation: NFC Feature Box design allows extending and shrinking the memory footprint of the application on Secure Element while the user opts-in and out to the services. Whereas, legacy NFC handsets cannot offer this flexibility. This allows NFC Feature Box users to freely change their feature set available on their phones over time.

To summarize, this thesis study contributes to the mobile transactions field by introducing new capabilities that were not possible with legacy NFC transaction systems.

6.2. Results

Besides contributing to the Mobile Transactions domain by improving the way we pay and by using NFC handset to perform mobile transactions in an efficient way without being dependent on any third party, this thesis study had aimed to achieve several objectives that are provided in Chapter 1. The results of our objectives are explained in Table 6.1.

Table 6.1. *Achievements*

Objective	*Achievement*
Objective 1: Identify major problems in the NFC ecosystem that is blocking mass adaption of the technology and limiting the usage to a few payment schemes only.	This objective is met at the initial investigation phase of this thesis. We have identified the enabler dependency problem as the major blocker in front of the technology adoption. Formal definition and a detailed explanation about the effect of NFC Enabler dependency in the NFC ecosystem is provided in this thesis.
Objective 2: Develop a User-Centric solution for the NFC ecosystem that users can register and start using NFC services at their will, without requiring third-party involvement.	Two major contributions of this thesis – RONFC and NFC Feature Box protocols – both allow NFC handset owners to manage NFC services provided to his phone without third-party involvement. At user's own consideration NFC handset can be started to be used for the desired mobile transaction by registering to the corresponding service directly.
Objective 3: Design and develop open NFC feature issuance and activation protocols that Service Providers can implement and join the NFC ecosystem on their own.	This objective is met by creating open protocols for both RONFC and NFC Feature Box that can be implemented by any Service Provider around the world to start accepting NFC devices as a replacement or in addition to physical cards that they are using to offer their services to their customer base.
Objective 4: Besides solving identified problems, contribute to enhancements of the NFC ecosystem by introducing features and flexibilities that are improving NFC transaction user experience.	This objective is met by introducing several advantages and flexibilities that cannot be offered by legacy NFC transaction systems such as enabling multi-factor authentication, flexible memory to manage active features on NFC device, etc. Additional contributions of RONFC and NFC Feature Box are provided in the Contributions section.

Objective 5: Discuss and evaluate the security and performance aspects of proposed solutions to ensure goals are achieved with a secure and efficient protocol.	This objective is met by challenging both RONFC and NFC Feature Box from security and performance metrics that are common in the financial transactions industry. RONFC and NFC Feature Box allow enabler independent transaction processing with optimized and robust protocols.

6.3. Future Work

This thesis achieves its objectives by offering "An Open, NFC Enabler Independent Mobile Payment and Identification Method". Furthermore, many other related topics will be created and become a candidate for research after having NFC Feature Box available in all NFC Phones. Some of these topics are listed below.

6.3.1. PKI Extension

NFC Feature Box supports only symmetric keys for transaction execution. Symmetric Key Cryptography is the main choice in Close-Circuit Offline Payment and Identification systems. Thus, NFC Feature Box covers the majority of the existing systems. However, technology improves rapidly and the crypto technologies that are used today become weaker after new attack factors are released. Public Key Infrastructure (PKI) based cryptography is the main choice in Online Payment systems such as credit card transactions. As a result, we expect enhancing NFC Feature Box with PKI based authentication mechanism will be a future research topic as it has challenges on transaction timing requirements of offline payment systems. Symmetric crypto calculations are much faster compared to Asymmetric crypto calculation which makes this choice difficult for offline payment and identification systems.

6.3.2. Feature Discovery Privacy

NFC Feature Box is a container application that keeps Feature Instances underneath. Therefore, a transaction flow starts with Feature Discovery to determine if the Feature is available in the tapped NFC Phone or not. This is a mandatory step in NFC Feature Box; however, it might be considered as a privacy issue as the availability of a feature in NFC Feature Box can be queried without any authentication. To demonstrate the privacy issue; let's assume a company wants to accept NFC Phones to access their buildings. They implement NFC Feature Box and employees register to the service and Feature Instance is installed on their phones. Let's assume that the FeatureUID of this company is 0x01020304. Sending Feature Discover command to all NFC Phones with 0x01020304 will reveal if the phone user is working for that company or not. In order to avoid such privacy issues, a new methodology of Feature Discovery can be investigated and developed by still maintaining the enabler independency. Adding additional authentication mechanism for Feature Discovery brings the necessity to distribute the relevant key to the authorized Service Providers which is again creating the central management requirement and so compromise enabler independence offered by NFC Feature Box. Offering a better solution to ensure user privacy is a challenging topic that needs to be studied.

6.3.3. Feature Recommendation System

NFC Feature Box offers a User-Centric model that allows NFC Phone user to register to any service that under user's exclusive control. However, NFC Feature Box assumes that a user is aware of the system that will be registered to. Considering that NFC Feature Box is implemented by thousands of Service Providers, searching for available Features would be a compelling task. In order to solve this problem, Smart Feature Recommendation Systems can be developed based on user profile or demographic data. To illustrate, a user living in a city might be aware of the services available within that city such as public transportation, parking system, et cetera.

When a user travels to a different city, Feature Recommendation System may recommend available features within that city when the user arrived at the airport. Filtering and matching accurate Services for a user based on user profile or demographic data would not be possible without advanced machine learning or artificial intelligence techniques.

REFERENCES

Akram, R.N., Markantonakis, K., & Mayes, K.E. (2010) 'A Paradigm Shift in Smart Card Ownership Model', in Proceedings of the 2010 International Conference on Computational Science and Its Applications (ICCSA), B. O. Apduhan, O. Gervasi, A. Iglesias, D. Taniar, and M. Gavrilova, Eds. Fukuoka, Japan: IEEE Computer Society. pp. 191–200.

Akram, R.N., Markantonakis, K., & Sauveron, D. (2015). A novel consumer-centric card management architecture and potential security issues. Information Sciences, 321, pp. 150-161.

Aldershof, P. (2012) 'Kick Start the NFC Ecosystem', in IT-Trans, IT Solutions for Public Transport (No. Session 1).

Al-Haj, A. and Al-Tameemi, M.A. (2018) 'Providing security for NFC-based payment systems using a management authentication server,' 4th International Conference on Information Management (ICIM), Oxford, pp. 184-187. doi: 10.1109/INFOMAN.2018.8392832

Alliance, S. C. (2011) 'The Mobile Payments and NFC Landscape: A US Perspective', Smart Card Alliance.

Alpar, G., Batina, L., & Verdult, R. (2012) 'Using NFC phones for proving credentials', In Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance, Springer. pp. 317-330.

Andersson, P., Markendahl, J., & Mattsson, L. G. (2011). Global Policy Networks' Involvement in Service Innovation. Turning the Mobile Phone into a Wallet by Applying NFC Technology. The IMP Journal, 5(3), pp. 193-211.

Barker, E., Barker, W., Burr, W., Polk, W., & Smid, M. (2007). NIST Special Publication 800-57 Recommendation for key management—part 1: general(revised).

Benyo, B. (2007) ‘Near Field Communication Technology: Contactless Applications in Mobile Environment’, In proceedings of the 8th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics.

Biswas, D. K., Sinclair, M., Hyde, J., & Mahbub, I. (2018). “An NFC (Near-field Communication) based Wireless Power Transfer System Design with Miniaturized Receiver Coil for Optogenetic Implants,” Texas Symposium on Wireless and Microwave Circuits and Systems, Waco, TX, USA, Apr. 5-6, pp. 1-5.

Bojjagani, S., and Sastry, V. N. (2019). "A secure end-to-end proximity NFC-based mobile payment protocol." Computer Standards & Interfaces, p.103348. DOI: 10.1016/j.csi.2019.04.007

Bos, Joppe W., et al. (2014) "Elliptic curve cryptography in practice." International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg.

Ceipidor, U. B., Medaglia, C. M., Marino, A., Sposato, S., & Moroni, A. (2012). A protocol for mutual authentication between NFC phones and POS terminals for secure payment transactions. ISC Conference on Information Security and Cryptology, pp. 115-120.

Chaum, D. (1985) ‘Security without identification: transaction systems to make big brother obsolete’, Communication of the ACM, 28(10), pp. 1030-1044.

Choudhary, B. & Risikko, J. (2006) ‘Mobile Financial Services Business Ecosystem Scenarios & Consequences’, Mobey Forum, c/o Nordea Bank, Satamaradankatu 3 B, 3rd floor, 00020 Nordea, Helsinki/Finland.

Coskun, V., Ok, K., & Ozdenizci, B. (2012) Near Field Communication (NFC): From Theory to Practice. London: Wiley. ISBN: 978-1-1199-7109-2, February, 2012.

Curran, K., Millar, A., & Garvey, C.M (2012) 'Near Field Communication', in International Journal of Electrical and Computer Engineering, Institute of Advanced Engineering and Science Press. 2(3). pp. 371-382.

Dias, J., Matos, J. N., & Oliveira, A. S. R. (2014). The Charge Collector System: A New NFC and Smartphone-based Toll Collection System. Procedia Technology, 17, pp. 130-137.

Dworkin, M. (2005) Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST Special Publication 800-38B

Dzafic, L., Ebner, M., & Youngkyun, B. (2017). "Game Based Learning Through Near Field Communication," In Game-Based Learning Theory, Strategies and Performance Outcomes, Youngkyun B. (ed.), Nova Publisher, pp. 295-322.

Eberspaecher, J., Voegel, H. J., & Bettstetter, C. (2008) 'GSM - Architecture, Protocols and Services', 3rd Ed., Wiley, New York.

EMV 4.2(2008) Book 1 Application Independent ICC to Terminal Interface Requirements, Book 2 - Security and Key Management, Book 3 - Application Specification, Book 4 - Cardholder, Attendant, and Acquirer Interface Requirements, EMVCo Std. 4.2.

EMVCo (2007) EMV mobile contactless payment technical issues and position paper EMVCo, Tech. Rep.

ETSI Specification of the Subscriber Identity Module (1996) 'Mobile Equipment (SIM - ME) interface (GSM 11.11)', European Telecommunications Standards Institute (ETSI Std. Version 5.3.0), Available at: http://www.etsi.org/deliver/etsi_gts/11/1111/05.03.00_60/gsmts_1111v050300p.pdf. [Accessed May 20, 2019].

Fernandez, M. J. L., Fernandez, J. G., Aguilar, S. R., Selvi, B. S., & Crespo, R. G. (2013). Control of attendance applied in higher education through mobile NFC technologies. Expert Systems with Applications, 40(11), pp. 4478-4489.

Francis, L., Hancke, G., Mayes, K., & Markantonakis, K. (2010). On the security issues of NFC enabled mobile phones. International Journal of Internet Technology and Secured Transactions, 2(3/4), pp.336-356.

GlobalPlatform (2006): GlobalPlatform Card Specification, Version 2.2,, GlobalPlatform Std.

GlobalPlatform (2012). GlobalPlatform A New Model: The Consumer-Centric Model and How It Applies to the Mobile Ecosystem.

Gronli, T., Pourghomi, P., & Ghinea, G. (2015). Towards NFC payments using a lightweight architecture for the Web of Things. Computing, 97(10), pp. 985-999.

Hang, A., Broll, G., & Wiethoff, A. (2010) 'Visual design of physical user interfaces for NFC-based mobile interaction', in proceedings of the 8th ACM Conference on Designing Interactive Systems (DIS '10). New York, NY, USA. ACM. pp. 292-301.

Hassinen, M., Hypponen, K., & Haataja, K. (2006). An open, PKI-based mobile payment system. In, Emerging Trends in Information and Communication Security, Berlin: Springer-Verlag, pp. 86–100.

ISO/IEC 7816-4 (2013) Organization, security and commands for interchange, International Organization for Standardization (ISO) Std.

ISO/IEC 9797-1 (2011) Message Authentication Codes Part 1, International Organization of Standardization (ISO) Std.

ISO/IEC 14443-3 (2011) Contactless Integrated Circuit Cards Part 3, International Organization for Standardization (ISO) Std.

ISO/IEC 18092 (2004) Near Field Communication - Interface and Protocol (NFCIP-1), International Organization for Standardization (ISO) Std.

Java Card Platform Specification (2009): Classic Edition; Application Programming Interface, Runtime Environment Specification, Virtual Machine Specification, Connected Edition; Runtime Environment Specification, Java Servlet Specification, Application Programming Interface, Virtual Machine Specification, Sample Structure of Application Modules, Sun Microsystem Inc Std. Version 3.0.1.

Kanniainen, L. (2010) 'Alternatives for banks to offer secure mobile payments', International Journal of Bank Marketing, 28(5), pp. 433-444.

Kitsos, P. (Ed.). (2016). Security in RFID and sensor networks. CRC Press.

Konidala, D. M., Dwijaksara, M. H., Kim, K., Lee, D., Lee, B., Kim, D., & Kim, S. (2012). Resuscitating privacy-preserving mobile payment with customer in complete control. Personal and Ubiquitous Computing, 16(6), pp. 643-654.

Koster, A., Matt, C., & Hess, T. (2016). Carefully choose your (payment) partner: How payment provider reputation influences m-commerce transactions. Electronic Commerce Research and Applications, 15, pp. 26-37.

Lehdonvirta, V., Soma, H., Ito, H., Yamabe, T., Kimura, H. & Nakajima, T. Ubipay (2009) 'Minimizing transaction costs with smart mobile payments', in proceedings of the 6th International Conference on Mobile Technology, Application; Systems, Mobility. New York, NY, USA, ACM. pp. 1-7.

Li, X., Niu, J., Khan, M. K., & Liao, J. (2013). An enhanced smart card based remote user password authentication scheme, Journal of Network and Computer Applications, 36(5), pp. 1365–1371.

Liu, J., Kauffman, R. J., & Ma, D. (2015). Competition, cooperation, and regulation: Understanding the evolution of the mobile payments technology ecosystem. Electronic Commerce Research and Applications, 14(5), pp. 372-391.

Lu, Y., Li, L., Haipeng, P., & Yang, Y. (2016). "Robust ID based mutual authentication and key agreement scheme preserving user anonymity in mobile networks." KSII Transactions on Internet & Information Systems 10.3.

Lyne, M., Dekozan, D., & Paetzold, K., (2016). 'Personal point of sale', U.S. Patent No. 9,312,923. Assignee: Cubic Corporation, Inc. Washington, DC: U.S. Patent and Trademark Office.

Madlmayr, G., Langer, J., & Scharinger, J. (2008) 'Managing an NFC ecosystem', in Proceedings of the 7th International Conference on Mobile Business, Washington, DC, USA: IEEE Computer Society, pp. 95–101.

Mayes, K.E. & Markantonakis, K. (2008) Smart cards, tokens, security and applications, Springer-Verlag New York Inc.

Morse, E., A. & Raval, V. (2008) 'Payment card industry data security standards in context', Computer Law Security Review, 24(6), pp. 540-554.

NFC Forum (2013) What are the operating modes of NFC devices?, Available at: http://nfc-forum.org/resources/what-are-the-operating-modes-of-nfc-devices/. [Accessed May 20, 2019].

NFC Forum (2016) NFC Forum Protocol Technical Specification, http://www.nfc-forum.org. Accessed May 20, 2019.

Ok, K., Coskun, V., Yarman, S. B., Cevikbas, C., & Ozdenizci B. (2016). SIMSec: A Key Exchange Protocol Between SIM Card and Service Provider, Wireless Personal Communications 89(4), pp. 1371-1390.

Ozdenizci, B., Ok, K., & Coskun, V. (2013). NFC loyal for enhancing loyalty services through near field communication. Wireless Personal Communications, 68(4), 1923–1942.

Pourghomi, P., Saeed, M. Q., & Ghinea, G. (2014). A Secure Cloud-Based Nfc Mobile Payment Protocol. International Journal of Advanced Computer Science and Applications, 5(10), pp. 24-31.

Raina, V.K. (2017). 'NFC Payment Architecture' in NFC Payment and the New Era of Transaction Processing, Hersley, PA, USA: IGI Global, ch 2, pp. 43-73. doi: 10.4018/978-1-5225-2306-2.ch002

Renardi, M. B., Kuspriyanto, K., Basjaruddin, N. C., & Prafanto, A. (2017). 'Baggage Claim in Airports Using Near Field Communication,' Indonesian Journal of Electrical Engineering and Computer Science, vol. 7, no. 2

Reveilhac, M. & Pasquet, M. (2009) 'Promising Secure Element Alternatives for NFC Technology', in proceedings of the 1st International Workshop on Near Field Communication, 2009. NFC '09, pp. 75.

Ruiz-Martinez, A., Sanchez-Montesinos, J., & Sanchez-Martinez, D. (2011). A mobile network operator-independent mobile signature service. Journal of Network and Computer Applications, 34(1), pp. 294–311.

Staykova, S. K., & Damsgaard, J. (2015). The race to dominate the mobile payments platform: Entry and expansion strategies. Electronic Commerce Research and Applications, 14(5), pp. 319-330.

Subashini, S., & Kavitha, V. (2011) 'A survey on security issues in service delivery models of cloud computing', Journal of Network and Computer Applications, 34(1), pp. 1-11.

Suryotrisongko, H., Sugiharsono, & Setiawan, B. (2012). A Novel Mobile Payment Scheme based on Secure Quick Response Payment with Minimal Infrastructure for Cooperative Enterprise in Developing Countries. Procedia - Social and Behavioral Sciences, 65, pp. 906-912.

Svitok, Miroslav (2014). Implementation of payment protocol on NFC-enabled mobile phone. Bacholar Thesis. MASARYKOVA UNIVERZITA

Tan, G. W. H., Ooi, K. B., Chong, S. C., & Hew, T. S. (2014). NFC mobile credit card: The next frontier of mobile payment? Telematics and Informatics, 31(2), pp. 292-307.

Turk, I., & Cosar, A. (2015a). Having 4G, Enabling Cloud Based Execution for NFC based Financial Transactions. In, 11th International Conference on Innovations in Information Technology (IIT), pp. 63-67.

Turk, I., & Cosar, A. (2015b). Internet Connection Sharing Through NFC for Connection Loss Problem in Internet-of-Things Devices. In, Internet of Things, Smart Spaces, and Next Generation Networks and Systems, Switzerland:Springer, pp. 329-342.

Turk, I., Angin, P., & Cosar, A. (2019). RONFC: A Novel Enabler-Independent NFC Protocol for Mobile Transactions. IEEE Access, Vol 7. doi: 10.1109/ACCESS.2019.2929011

Van Damme, G., Wouters, K. M., Karahan, H., & Preneel, B. (2009) 'Offline NFC payments with electronic vouchers', in Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds, pp. 25-30. ACM.

Widmann, R., Grunberger, S., Stadlmann, B., & Langer, J. (2012) 'System Integration of NFC Ticketing into an Existing Public Transport Infrastructure', in proceedings of the 4th International Workshop on Near Field Communication (NFC),IEEE. pp. 13-18.

Zhang, X. & Zhao, S. (2018). Design of NFC Mobile Payment System Based On Fingerprint Identification. In 2017 3rd International Forum on Energy, Environment Science and Materials (IFEESM 2017). Atlantis Press. doi: doi.org/10.2991/ifeesm-17.2018.240

Zhao, H., Anong, S. and Zhang, L. (2019). "Understanding the impact of financial incentives on NFC mobile payment adoption", International Journal of Bank Marketing, Vol. 37 No. 5, pp. 1296-1312. doi: /10.1108/IJBM-08-2018-0229

APPENDICES

A. Example Commands

In order to clarify command data construction for NFC Feature Box developers, some examples are provided below;

Select applet

Command: 00 A4 04 00 0B 46 65 61 74 75 72 65 42 6f 78 31 00

Response: 90 00

Data Segment of Create Instance Command

01 04 F1 F2 F3 F4 02 04 A1 A2 A3 A4 03 02 02 01 04 17 4f 66 66 6c 69 6e 65 20 50 61 79 6d 65 6e 74 20 53 65 72 76 69 63 65 05 15 41 6e 6b 61 72 61 20 54 72 61 6e 73 70 6f 72 74 61 74 69 6f 6e 06 01 01 07 01 01 08 01 02

Coded Data:

Feature UID: F1F2F3F4

Instance UID: A1A2A3A4

Version: 2.1

Feature Info: "Offline Payment Service"

Public Info: "Ankara Transportation"

One Static Data File, One Value File, Two Authentication Keys

0x06 0x01 <#of Data Files>

0x07 0x01 <#of Value Files>

0x08 0x01 <#of Authentication Keys>

Put Data Command for Static Data File

80 DA 01 01 10 <encrypted data> 00

Data before encryption

01 02 00 64 02 01 FF 03 01 01

Coded Data:

File Size: 100 bytes

Read Access: Public

Write Access: With Key-1

Put Data Command for Value File

80 DA 02 02 20 <encrypted data> 00

Data before encryption

01 04 00 00 00 05 02 04 00 00 00 00 03 04 00 00 03 E8 04 01 01 05 01 00 06 01 02

Coded Data:

Initial Value: 5

Allowed Min: 0

Allowed Max: 1000

Read Access: Key-1

Increment Access: Forbidden

Decrement Access: Key-2

Put Data Command for Authentication Key

80 DA 03 01 20 <encrypted data> 00

Data Before Encryption

01 01 01 02 10 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F

Coded Data:

Key Type: 2KTDES

Key Data: 4142434445464748494A4B4C4D4E4F

Feature Discovery

Command: 80 FD 00 00 04 F1 F2 F3 F4 00

Response: A1 A2 A3 A4

Read 10 bytes from Static Data File

80 91 01 00 0A 00 0A <MAC> 00

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name	: Türk, İsmail
Nationality	: Turkish (TC)
Date and Place of Birth	: 5 April 1983, Konya
Phone	: +90 532 562 07 42
E-mail	: ismail.turk@metu.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	METU Computer Engineering	2010
BS	Bilkent Computer Tech & Inf Systems	2006
High School	SD Science High School, Isparta	2001

WORK EXPERIENCE

Year	Place	Enrollment
2018-Present	Symphony Communication Services	Principal Security Engineer
2012 May	NXP Semiconductors	Field Application Engineer

FOREIGN LANGUAGES

Advanced English, Basic German

PUBLICATIONS

1. Turk I. and Cosar A. "Having 4G, Enabling Cloud Based Execution for NFC based Financial Transactions", Innovations in Information Technology (IIT), 2015 11th International Conference, 63-67 (2015)

2. Turk I. and Cosar A. "Internet Connection Sharing Through NFC for Connection Loss Problem in Internet-of-Things Devices", Internet of Things, Smart Spaces, and Next Generation Networks and Systems Conference, 337-350 (2015)

3. Turk I. and Cosar A. "Extended Abstract, An Open, NFC Enabler independent Mobile Payment and identification method: NFC Feature Box", IEEE 17th

International Symposium on A World of Wireless, Mobile and Multimedia Networks, 1-3 (2016)

HOBBIES

Puzzles, Entrepreneurship, Reading, Trading Metric Analysis