Analysing and Improving the Security of Contactless Payment Cards

By

Ossama AL-Maliki

School of Computing

University of Buckingham

United Kingdom

A Thesis for the Degree of Doctor of Philosophy in Computer Science to the School of Computing at the University of Buckingham

June 2020

Abstract

Europay, MasterCard, and Visa (EMV) is the most used payment protocol around the world with 85.9% of the payment cards in the EU and the UK being EMV based cards in 2019. The EMV payment protocol has made contactless transactions faster and more convenient for cardholders as they only need to place the card next to the Point of Sale (POS) to make a payment. According to the latest report of the UK Finance, the total value of contactless card transactions in 2019 was higher than the cash ones for the first time ever.

On the other hand, the introduction of the wireless interface in the EMV contactless transactions opens the door for several attacks to be launched on contactless cards such as skimming, eavesdropping, replay, and relay attacks. Since April 2020, the limit of contactless transactions has increased to £45 as a response to the Covid-19 crisis. This might create an extra motivation for launching more attackers on contactless cards.

This thesis is primarily concerned with investigating and analysing the security of contactless card's payments and uncovering the impact of key vulnerabilities in the EMV contactless card specifications. The two main vulnerable are the one-way authentication methods and the lack of cardholder verification in such transactions.

The thesis also proposes the following four practical protocols to improve the security and the privacy of the EMV contactless cards.

1- A new tokenization protocol to replace the actual Primary Account Number (PAN) with a token to prevent the EMV contactless cards from revealing the actual PAN.
2- A mutual authentication protocol to address the vulnerabilities related to the EMV one-way card authentication methods in the EMV payment protocol.
3- A novel gyroscope sensor into EMV contactless cards to be used for activating the cards by perfuming a simple move by the cardholder.
4- A protocol to use cardholders' NFC enabled smartphones to activate contactless cards.

The two main aims of these four proposed protocols are to prevent such cards from being read by unauthorised NFC enabled readers/smartphones and to give cardholders more control of their contactless cards in order to prevent several attacks. Moreover, the thesis also describes a Java framework to mimic a genuine EMV contactless card and validate the four proposed solutions.

The thesis argues that the first two proposed solutions require minimal changes to the existing EMV infrastructures and do not have any impact on the user's experience while the last two proposed solutions require some changes the users' experience when making contactless card transactions.

Acknowledgment

I would like to express my deepest gratitude to God of universes, **Allah** the almighty, the most gracious and the most merciful for all the blessings in my life and for giving me the hope and strength to finish this thesis.

I wish to express my gratitude to my supervisor **Dr. Hisham Al-Assam**. His valuable time, guidance and encouragement advice are the real key to make this thesis sees the light.

I would like also to thank all the staff in the school of computing for all their support and guidance during the seminaries to shape up my research and ideas.

Special thanks to all my research colleagues especially **Ali Al-Ameri** and **Alnoman Abdulkhudhur** for their supports and helps in the bad times and the good times.

I should not forget to thank the Iraqi Ministry of Higher Education and Scientific Research (**MOHESR**) for sponsoring my financially during the whole period of my study.

Ossama AL-Maliki

© 2020

Dedication

This work is humbly dedicated to:

- My father **Prof. Lazim AL-Maliki** and my beloved mother (**Fatma**) , as I am absolutely certain that finishing my Ph.D. is the most desirable gift they wish for.
- My wife and soulmate **Hawraa,** I cannot imagine my life without you by my side. Thank you for your love, understanding, support and mostly for rising up our beautiful babies.
- My two babies and my treasures in this life, **Zaid** my son and the one and only who could make me smile at any time, **Ishtar**, my beautiful little girl which I cannot resist her eyes.
- My brothers (**Ahmed, Ali and Mohammed**) and beautiful sister (**Eman**).

Ossama AL-Maliki

© 2020

Declaration

I (Ossama AL-Maliki), hereby declare that all the work in this research thesis is my own work except where due reference is made within the text of the thesis.

I also declare that to the best of my knowledge, none of the material has ever previously been submitted for a degree in this or any other university.

Ossama AL-Maliki

Abbreviations

AB	Acquirer Bank
AC	Application Cryptogram
AES	Advance Encryption Standard
AFL	Application File Locater
AID	Application Identification
AIP	Application Interchange Profile
APDU	Application Protocol Data Unit
ATM	Automated Teller Machine
CA	Certificate Authority
CA_{pk}	Certificate Authority Public Key
CA_{sk}	Certificate Authority Private Key
CDA	Combine Data Authentication
CLS	Class
CNP	Card Not Present
C_{pk}	Card Public Key
C_{sk}	Card Private Key
CVC	Card Verification Code
CVM	Cardholder Verification Method
DDA	Dynamic Data Authentication
DTW	Dynamic Time Warping
EER	Equal Error Rate

EMV	Europay, Mastercard and Visa
GPO	Get Processing Options
IB	Issuer Bank
IB_{pk}	Issuer Bank Public Key
IB_{sk}	Issuer Bank Private Key
INS	Instruction
JCIDE	Java Card Integrated Development Environment
LC	Length Command
LE	Length Expected
NFC	Near Field Communication
P1	Parameter 1
P2	Parameter 2
PAN	Primary Account Number
PAR	Payment Account Reference
PDOL	Processing Data Option List
PIN	Personal Identification Number
POS	Point of Sale
PPSE	Proximity Payment System Environment
RFID	Radio Frequency Identification
RN	Random Number
SDA	Static Data Authentication
SI	Static Information
SW	Status Word

TD	Transaction Data
TR	Token Requester
TSP	Token Service Provider
UN	Unpredictable Number

Table of Contents

List of Figures

List of Tables

1. Chapter 1: Introduction

Nowadays, the Near Field Communication (NFC) and Radio Frequency Identification (RFID) technologies are being increasingly used in smartphones and smart cards. One of the most significant deployments of these technologies is the contactless smart cards for identification, accessing a building, contactless card payments and mobile payments [1]. This thesis focuses on the Europay MasterCard and Visa (EMV) payment protocol, which is an international standard for chip-based payment protocol owned by the EMVCO and introduced to replace the magnetic stripe payment protocol. EMV payment protocol supports five different payment methods namely chip & PIN, chip & signature, magnetic stripes, contactless card and mobile and payments [2].

The EMV contactless card transactions were introduced in the UK in 2007 to allow users to make a quick and easy payment for goods or services with an amount less than £45 in the UK without the need to enter the Personal Identification Number (PIN) [3][4]. The contactless payments are based on the ISO 14443 in which the cardholders should place their contactless card within 10cm of the Point of Sale (POS) [5]. The advantage of using a contactless card to end-users is the ability to do faster transactions within almost half a second while merchants spend less time on each contactless card transactions compared to chip & PIN or cash transactions [6]. The Smart Payment Association and the UK Card Association reported that customers who are using contactless cards transaction are likely to spend 30% more than using chip &PIN or cash [7].

1.1. Problem Statement

Introducing the EMV contactless card transactions has opened the door widely to launch new attacks that were not possible to launch on the chip & PIN transactions such as skimming, eavesdropping, and relay attacks [8][9][10]. This due to the wireless connectivity between the POS and EMV contactless cards. In fact, the cards are vulnerable to leaking sensitive information not only to POSs but also to any NFC enabled readers/smartphones [11][12]. The sensitive information that can be easily sniffed includes cardholder name, Primary Account Number (PAN), the expiry date of the card, Issuer Bank (IB) public key certificate and other kinds of information. Besides the wireless connectivity, the existing EMV specifications of contactless card transaction have a keys vulnerability related to the one-way authentication nature of the protocol. The specifications state the EMV card must always authenticate itself to the POS while the opposite is not required in the specifications [2].

The thesis argues the one-way authentication could potentially undermine the security contactless transactions as it allows unauthorised NFC readers/Smartphones to launch skimming, replay and relay attacks. Furthermore to maximise user's convenience, the EMV contactless specifications do not have any cardholder verification [13][14]. The no cardholder verification in the EMV contactless card transactions ends the relationship between cardholders and their cards. Figure 1-1 shows some of the attacks on the EMV contactless card transactions where more details about all the vulnerabilities and attacks are presented in Chapter 2.

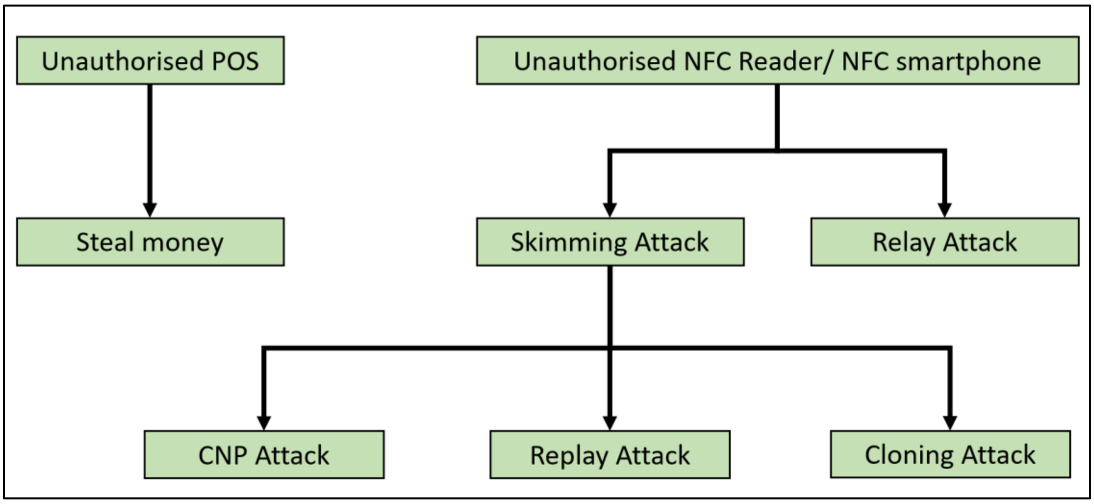


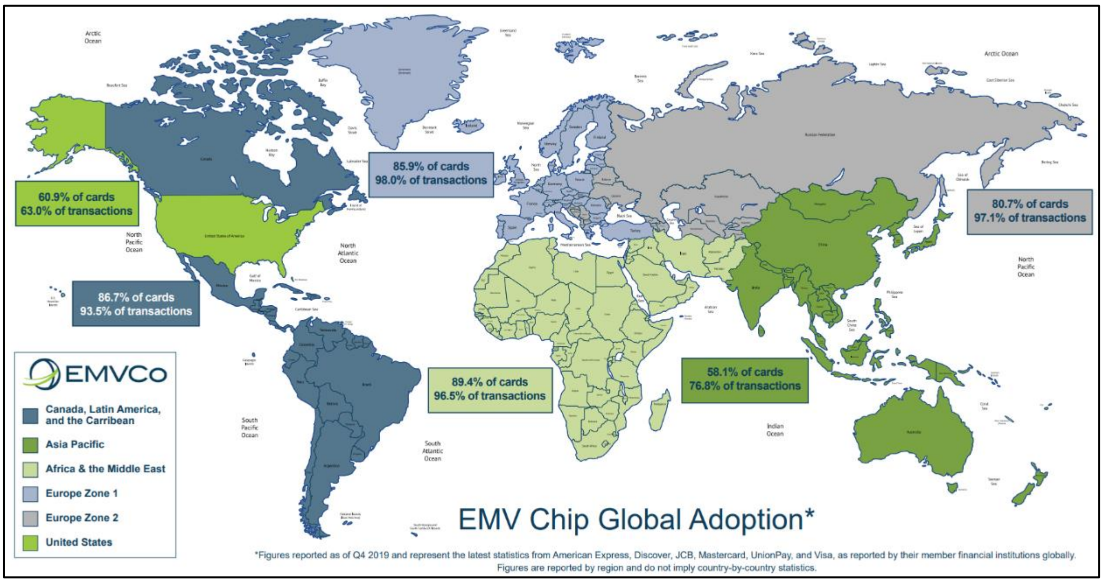
Figure 1-1: Contactless Card Transactions Attacks Tree

1.2. Research Motivations

This section explains the key motivations behind the research carried out in this thesis to improve the security of EMV contactless cards. We need to address that all the figures in this section are not part of my work, as all the figures have been used directly from different sources as clearly shown in the figure's title.

1.2.1. The Popularity of the EMV Payment Protocol Worldwide

The EMVCO reported in the last quarter of 2019 that 85.9% of the credit and debit cards in the EU and the UK were EMV based cards and 98% of payment transactions were EMV transactions processed by EMV supported POS/ATM [15]. Figure 1-2 shows the global adoption map of the EMV payment protocol around the world. It obviously that this payment protocol controls most of Europe zone 1 and 2, Canada, Latin America, Caribbean. However, the numbers of the EMV protocol adoption are still quite low in the United States, Australia and some parts of Asia where the magnetic stripe payment protocol still very popular.



*Figures reported as of Q4 2019 and represent the latest statistics from American Express, Discover, JCB, Mastercard, UnionPay, and Visa, as reported by their member financial institutions globally. Figures are reported by region and do not imply country-by-country statistics.

Figure 1-2: Global Adoption of the EMV Protocol, adapted from [15]

1.2.2. The Scale of Contactless Card in the UK

According to the latest report of the UK Finance, the number of contactless cards in the UK is increasing each year as shown in Figure 1-3 [16]. The figure shows that, by the end of 2019, the number of contactless cards is nearly 47 million credit cards and 84.4 million debit cards.

Moreover, the same report shows that the number of contactless card transactions in the UK had reached 760 million contactless card transactions in October 2019 as shown in Figure 1-4. The figure shows a dramatic increase over the last four years where only 81 million contactless transactions were made in June 2015.

In fact, the total value of contactless card transactions has dramatically increased over the last four years as shown in Figure 1-5. Due to the increasing popularity of contactless cards in the UK, the total value of these transactions was more than 7 billion GBP in October 2019 whereas the total value was around 0.5 billion GBP in June 2015. The UK Finance report states that the total value of contactless card transactions has passed the total value of cash transactions in 2019 for the first time ever since introducing the contactless cards more than a decade ago [17]. All the figures clearly indicate that the EMV contactless transactions are a popular way to pay by the consumers in the UK.

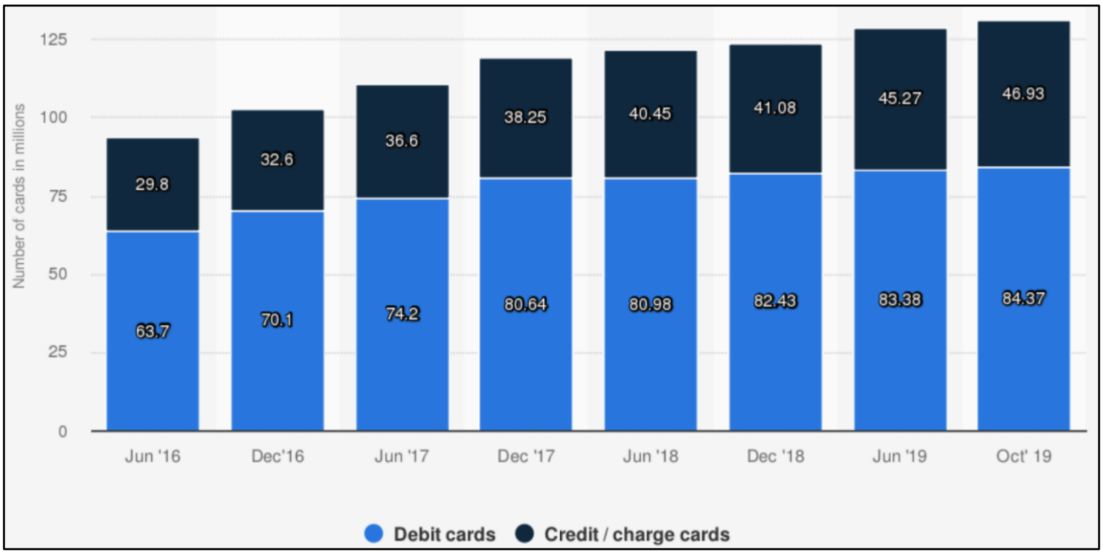


Figure 1-3: Numbers of Contactless Cards in the UK, adapted from [16]

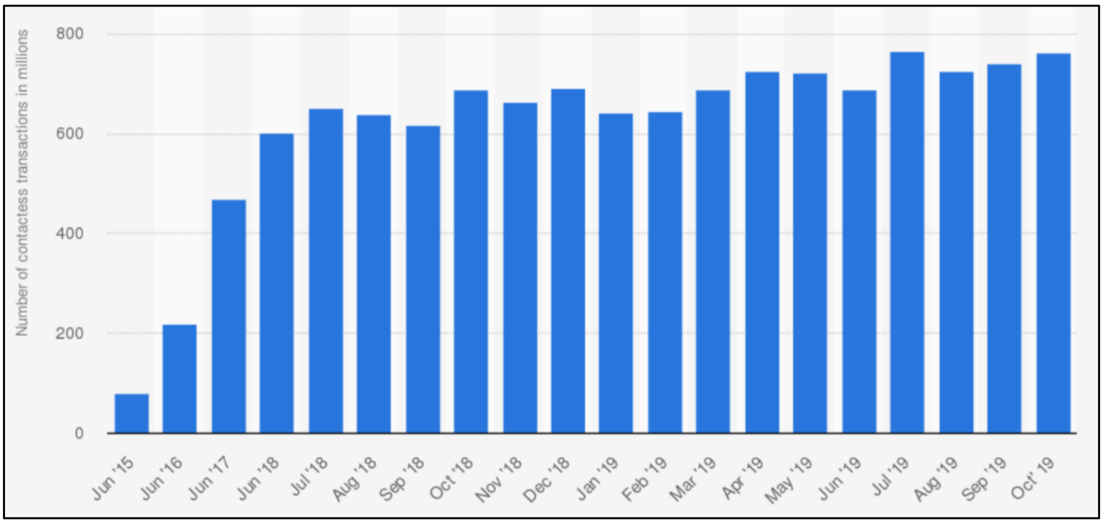


Figure 1-4: Numbers of Contactless Cards Transactions in the UK, adapted from [16]

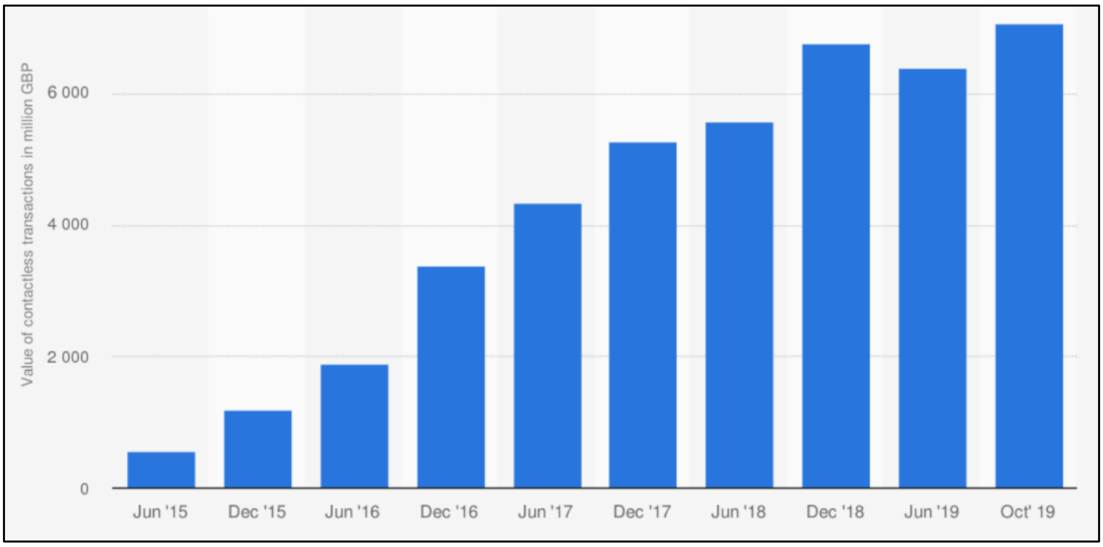


Figure 1-5: Value of Contactless Cards Transactions in the UK, adapted from [16]

1.2.3. The Scale of Frauds/Attacks on Contactless Cards

The UK Finance reported in its newest report of 2020 that 76% of the losses on the card payment was due to remote purchases or CNP attacks [18][19]. The fraudsters can arguably gain an advantage from their ability to sniff sensitive information from EMV contactless cards to launch such kind of attack (CNP)[20]. Figure 1-6 shows the card payment fraud by types in 2019, where remote purchases fraud losses were 76%, 6% for ID theft and 2% for counterfeit cards, and 15% for lost and stolen cards fraud.

The same source reported that the total amount of remote purchase or what is called Card Not Present (CNP) in the UK 2018 increased by 24% compared with the previous year with a total remote purchase fraud losses of £506.4 million as shown in Figure 1-7.

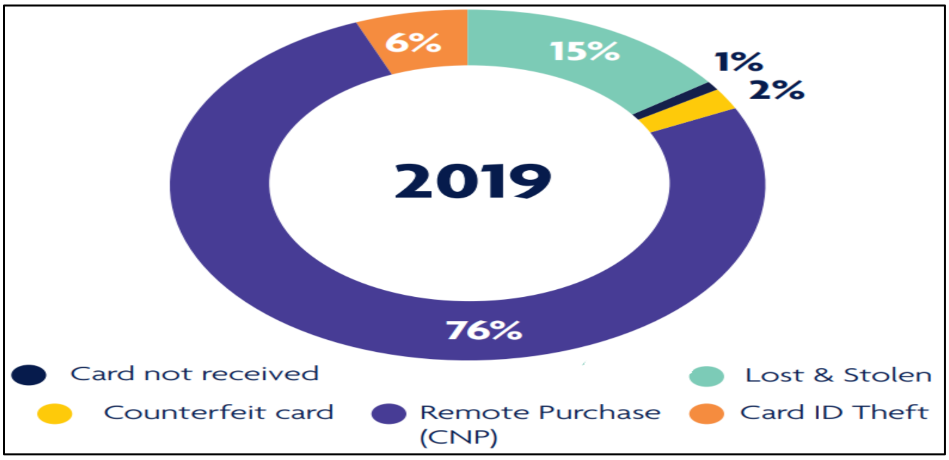


Figure 1-6: Card Frauds by types in the UK, adapted from [18]

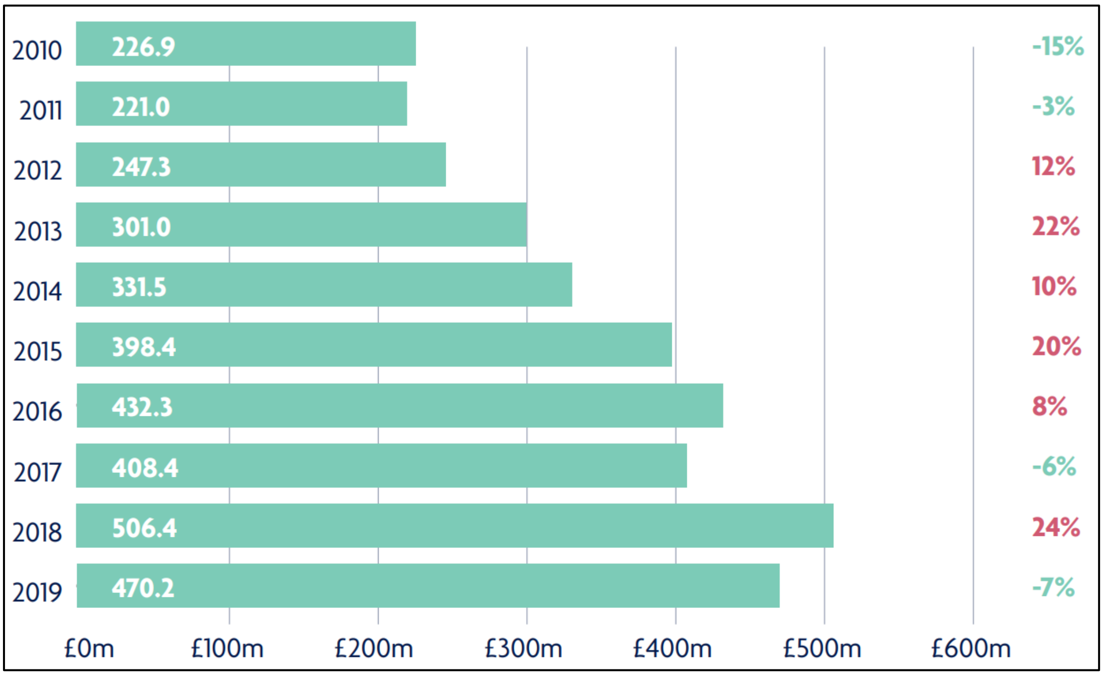


Figure 1-7: Remote Purchases Fraud in the UK, adapted from [18]

1.2.4. Increasing the Contactless Card Transactions Limit

When the contactless cards transactions first introduced in 2007 in the UK, the limit of such transactions was £10 [21]. After just three years, the limit increased by £5 to be £15 in 2010. Then in 2012, the limit increased again to £20 followed another £10 increase to a £30 limit in 2014 [22].

From 1st of April 2020, the limit of contactless card transactions has increased by £15 to be £45 according to both of British Retail Consortium and the UK Finance [23] [24]. This limit increase is being introduced as the payment industry's response to the Covid-19 epidemic to reduce the need for physical contact with the POSs.

While the limit increase encourages cardholders to use contactless card more often, fraudsters, on the other hand, could take an advantage of the situation to launch more attacks such as relay attack as the profit of such attacks is now bigger.

1.3. Aim and Objectives

The ultimate aim of this thesis is to enhance the security of the EMV contactless transactions to prevent several possible attacks on such cards. To achieve that, we proposed four protocols at two different levels in order to improve the security of such transactions. The first level requires minimum changes to the original EMV infrastructures whether on the card's side, POS's side or even the cardholders' experiences while the second level, requires more changes than the first level and it requires changes in the cardholders' experiences.

The following points summed up the thesis objectives:

1- Stopping the EMV contactless cards from revealing the sensitive information such as the PAN to unauthorised NFC enabled readers/smartphones.
2- Preventing the EMV contactless cards from being engaging in wireless connectivity with unauthorised NFC enabled POS/readers/smartphones without the knowledge of the card.
3- Bringing back the missing association between the cardholders and their EMV contactless cards in order to stop various attacks on such cards.

To achieve these objectives, we need to follow these tasks:

1- Investigating and understanding the EMV payment protocol and its specifications, especially the contactless card specification.
2- Evaluating the literature review of the EMV contactless cards and all kinds of attacks on such cards.
3- Building and developing system to simulate the EMV contactless card transactions in order to test the thesis's proposals.

1.4. Contributions to Knowledge

The following two points are a summary of the contribution to the knowledge in this thesis.

1. Designing and developing a Java contactless card framework to duplicate genuine EMV contactless cards and POS communications according to the EMV payment protocol specifications. The Java framework is used to simulate the contactless transactions and illustrate the feasibility of the security-improved protocols proposed in this thesis.
2. Presenting a comprehensive and rational analysis of the EMV contactless card transactions that focusses on the EMV card authentication methods to demonstrate their strength and weakens. The Java framework is used to implement different card authentication methods to test how each method reacts to various attacks.

1.5. Thesis Novelties

The following points are a summary of this research thesis novelties to enhance the security of the EMV contactless cards payment.

1. Proposing a new tokenisation technique to replace the actual PAN with a token to prevent the EMV contactless cards from revealing the PAN, which is one of the most sensitive information that could be used to launch a CNP attacks.
2. Proposing a mutual authentication protocol to address the vulnerabilities related to the EMV one-way card authentication methods in contactless transactions to prevent reading the EMV cards by NFC enabled readers/smartphones.
3. Proposing the integration of a gyroscope sensor into EMV contactless cards to be used for activating the cards by performing a simple move by the cardholder to prevent skimming and relay attacks i.e. the cards do not engage with any RFID communication before activating. The thesis shows that the proposal can be extended for the use of authenticating cardholders.
4. Proposing the use of cardholders' NFC enabled smartphones to send an activation command to the EMV contactless cards to allow them to engage with RFID communications to prevent different attacks.

1.6. List of Publications

The following shows our research publications during the Ph.D. study:

1. Ossama AL-Maliki, Hisham AL Assam, "On the Security of the EMV Authentication Methods of Contactless Cards", Accepted in the European Conference on Cyber Warfare and Security, 2020.
2. Ossama AL-Maliki, Hisham AL Assam, "A Tokenisation Technique for the Security of EMV Contactless Cards", Submitted to Information Security Journal: A Global Perspective, Taylor & Francis, 2020.
3. Ossama AL-Maliki, Hisham AL Assam, "Challenge-Response Mutual Authentication Protocol for EMV Contactless Card and POS", Submitted to Computer & Security Journal, Elsevier, 2020.

1.7. Thesis structure

The rest of the thesis is structured as follows.

Chapter 2 presents background information related to the EMV payment protocol and all the different payment methods supported by the EMV payment protocol. The chapter details all the main phases of this protocol, namely card authentication methods, CVMs, and transaction authorisation methods. It also presents the EMV contactless card specifications. Also, it presents a security analysis for the EMV contactless card where a number of relevant vulnerabilities are explained.

Chapter 3 presents our first contribution where a Java contactless card framework was designed and developed. The chapter reviews the Application Protocol Data Unit (APDU) and all the main EMV contactless card APDU commands and responses. It also shows all the hardware and software that were used to build the Java contactless card framework. Finally, the chapter shows the implementation results and discussion points.

Chapter 4 details the second phase of the EMV payment protocol, namely the card authentication methods, and dives into each of the three authentication methods by showing their strength and weaknesses. Also, Chapter 4 shows the implementation of all the three different authentication methods using the Java contactless card framework.

Chapter 5 proposes a tokenisation solution to improve the security of the EMV contactless payment cards. It also presents the motivations that led to our tokenisation proposal. Besides, the EMV mobile payment specification is detailed in this chapter. Finally, the chapter presents the implementation results of the tokenisation proposed approached using the Java contactless card framework.

Chapter 6 presents another contribution related to the mutual authentication between the POS and the EMV contactless card. The chapter provides details on the implementation results and some discussion points.

Chapter 7 presents the proposed solution to prevent both skimming and relay attacks on the EMV contactless cards by incorporating a built-in gyroscope sensor.

Chapter 8 proposes the use of the cardholders' NFC enabled smartphones to activate the EMV contactless cards to prevent unauthorised reading by any NFC enabled POSs/readers/smartphones.

Chapter 9 concludes the thesis with a summary of all the contributions and findings of this research and highlights potential future research directions.

2. Chapter 2: Europay MasterCard and Visa Payment Protocol

This chapter consists of two main parts, where the first part aims to give a detailed background of the EMV payment protocol and its specifications. Also, it explains the three main phases of the EMV payment protocol namely card authentication methods, cardholder verification methods and transaction authorisation. The first part focuses on the contactless card transactions as such transactions are the main concern of the thesis. While the second part explains and evaluates all known vulnerabilities, attacks, and countermeasures related to contactless card transactions.

This chapter is organized as follow. Section 2.1 shows the EMV payment protocol specifications and it presents all the main players in such transactions. Then, the same section details the three main phases of the EMV payment protocol and then explains the EMV contactless card transactions. While Section 2.2.12 shows all the vulnerabilities that let to various attacks and details most of the countermeasures to prevent several attacks on the EMV contactless card transactions. Finally, Section **Error! Reference source not found.** concludes the chapter.

Please note that expert readers can skip this chapter and move to Chapter 3.

2.1. EMV Payment Protocol Specifications

The EMVCO owned the EMV payment protocol specifications and it offers publicly available different specifications for each payment method which this protocol supported [25]. The EMV payment protocol offers a large number of functionalities and options for both EMV cards, mobile and POSs/ATMs. This makes the EMV protocol a very complex protocol with more than 2400 pages of specifications divide into contact, contactless and mobile payment specifications.

The EMV specifications offer four main books for the chip & PIN transactions (contact transactions). Book 1 details the Application independent of the EMV cards and the POS/ATM interface requirements [26]. Book 2 identifies the card authentication methods and details all the key management rules [27]. Furthermore, book 3 summarizes the application specifications and all the CVMs that could be used in such transactions [28]. Finally, book 4 defines the mandatory, recommended and optional POS/ATM requirements necessary to support EMV chip & PIN transactions [29].

On the other hand, the EMV payment protocol provides three main books (A, B and D) and seven different kernels (C1:7) for the EMV contactless card specifications. Book A defines the main architecture and the general requirements of the POS functionality in order to support the EMV contactless card transactions [30]. While book B the entry point specification describes how the protocol should work by listing the selection and activation requirements for such transactions [31]. However, book D explains the EMV contactless communication protocol specifications between the POS and the EMV card [32].

Additionally, the EMV contactless card specifications have seven different kernels namely kernel 1 for Visa [33], kernel 2 for MasterCard [34], kernel 3 for Visa card that support Fast Dynamic Data Authentication (FDDA) in the offline transactions [35], kernel 4 for American Express [36], kernel 5 for Japan Credit Bureau (JCB) [37], kernel 6 for Discover [38] and kernel 7 for Union Pay [39].

In addition to all these specifications books and kernels, the EMVCO provides specifications for the EMV mobile payment and magnetic stripe payment, secure remote commerce specifications and several technical white papers as all of the EMV specifications are publicly available on their website [25]. Table 2-1 shows the main specifications for both the chip & PIN and contactless card transactions.

Table 2-1: EMV Specifications Overview

Transaction Type	**Book**	**Book's Title**	**#Pages**
Chip & PIN (Contact)	Book 1	Application Independent ICC to Terminal Interface Requirements	189
	Book 2	Security and Key Management	174
	Book 3	Application Specification	230

	Book 4	Cardholder, Attendant, and Acquirer Interface Requirements	154
Contactless Card	Book A	Architecture and General Requirements	119
	Book B	Entry Point Specification	60
	Book C-1	Kernel 1 Specification (Visa)	34
	Book C-2	Kernel 2 Specification (MasterCard)	593
	Book C-3	Kernel 3 Specification (Visa FDDA)	153
	Book C-4	Kernel 4 Specification (American Express)	177
	Book C-5	Kernel 5 Specification (Japan Credit Bureau)	134
	Book C-6	Kernel 6 Specification (Discover)	122
	Book C-7	Kernel 7 Specification (Union Pay)	75
	Book D	EMV Contactless Communication Protocol Specification	249

2.1.1. EMV Transaction Parties

In this section, we define the main players involved in EMV transactions according to the EMV specifications [40]. Figure 2-1 shows all these players.

EMV Card: A device that includes both a secure microcontroller and a secure internal memory. The EMV card could be connected directly to a reader in case of contact transaction or remotely through a radio frequency interface in case of contactless transactions. The EMV card could store data and processed several activities such as encryption/ decryption and it could interact with a POS and ATM according to the EMV specifications.

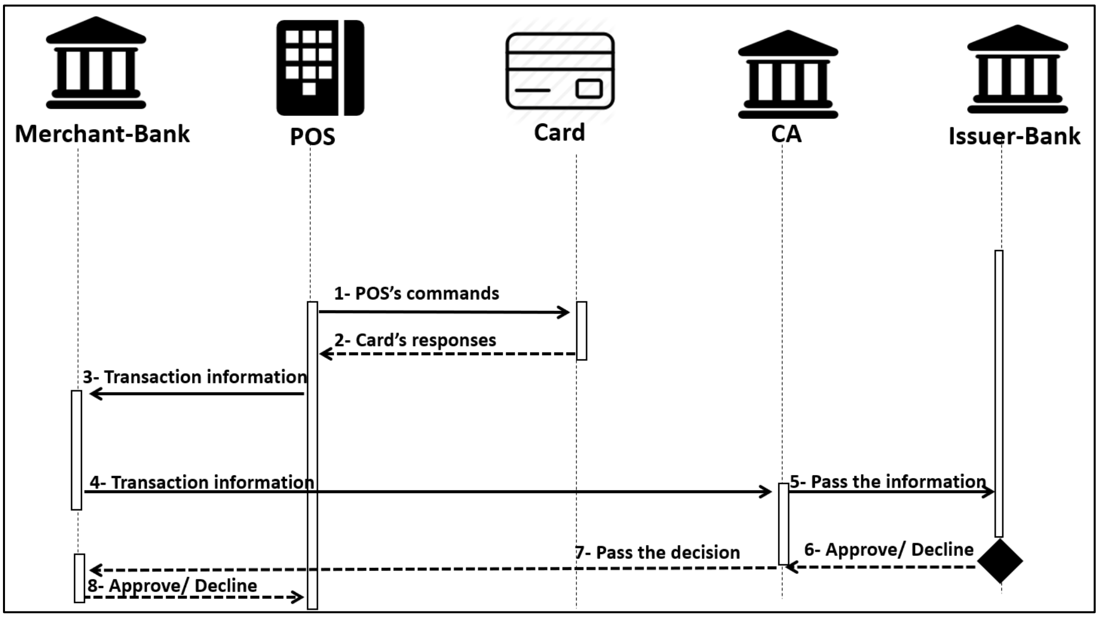


Figure 2-1: EMV Transaction Main Players

Issuer Bank: It refers to the bank that issued the EMV card to the cardholder. We refer to it here in this thesis as IB for simplicity. There are several activities that the IB could ensure such as the card data preparation, configuration set up, key management, define card profile and risk parameters and several other activities that the IB could provide.

Cardholder: This term is referring to the person that the EMV card is issued for by the Issue Bank. This person is the owner who is performing a transaction.

Acquirer Bank (AB): it refers to the bank that represents the merchants in the transaction. We refer to it here in this thesis as AB for simplicity. The AB connects the merchant transactions to the payment network, and it provides the merchants with the POS devices.

POS/ATM: POS or the Automated Teller Machine (ATM) are both able to process the EMV card transaction whether this transaction is contact or contactless transaction. In this thesis, we will focus on the POS as the thesis attention is about the contactless card transactions.

Certificate Authority: A trusted central administration that connects all the IBs with the ABs during the processing of transactions. We refer to it in this thesis as CA for simplicity. The CA could be MasterCard, Visa, American Express, etc. There are several activities that the CA could manage such as issuing certificate keys.

2.1.2. The Three Main Phases of EMV Payment Protocol

The EMV payment protocol consists of three main phases namely, card authentication methods, cardholder verification methods and the transaction authorisation methods. Each one of these stages has several different options that depend on both POS/ ATM and EMV card capabilities. To process a successful EMV transaction, all or at least two of these stages must be processed depends on the transaction type. The next subsections detail all these three stages.

2.1.2.1. EMV Card Authentication Methods

The EMV card authentication methods are used to authenticate the EMV cards to the POS/ATM by requesting the EMV cards to produce a digital signature in order to prove the authenticity and genuinely of such cards [27][41]. The EMV specifications support three types of card authentication methods as shown in Figure 2-2. These are the three methods:

1. Static Data Authentication (SDA)
2. Dynamic Data Authentication (DDA)
3. Combine Data Authentication (CDA)

Each EMV card must support at least one of the above authentication methods. This is typically decided by the IB who decides the capability of the EMV card [42]. All these EMV card authentication methods are detailed and studied in Chapter 4 where we show all the strengths and weak points of each method by analyzing and evaluating each method individually.

One of the key vulnerabilities in the EMV payment protocol is the one-way authentication that forces the EMV cards to authenticate themselves to POS/ATM while the reverse is not happening. This vulnerable point in the protocol led to several kinds of attacks such as skimming and relay attacks. Therefore, we designed a two-way (mutual) authentication protocol to authenticate the POS/ATM as well as the EMV cards. This mutual authentication protocol is detailed in Chapter 6.

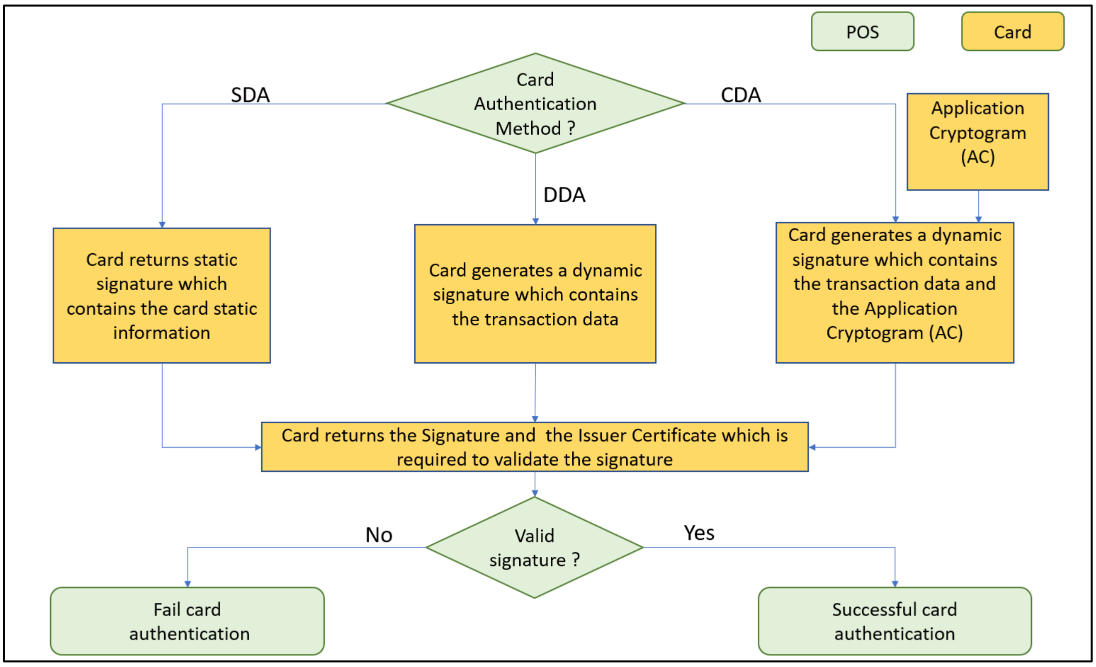

POS
Card
Card Authentication Method ?
SDA
CDA
DDA
Application Cryptogram (AC)
Card returns static signature which contains the card static information
Card generates a dynamic signature which contains the transaction data
Card generates a dynamic signature which contains the transaction data and the Application Cryptogram (AC)
Card returns the Signature and the Issuer Certificate which is required to validate the signature
No
Valid signature ?
Yes
Fail card authentication
Successful card authentication

Figure 2-2: EMV Card Authentication Methods

2.1.2.2.Cardholder Verification Methods (CVM)

The main aim of the second stage of the EMV payment protocol is to verify the genuine cardholder during the EMV transactions. The CVM is defined by the IB and each EMV card could support several different types of CVMs depends on the capability of each EMV card [43]. Figure 2.3 shows the three main types of EMV CVMs.

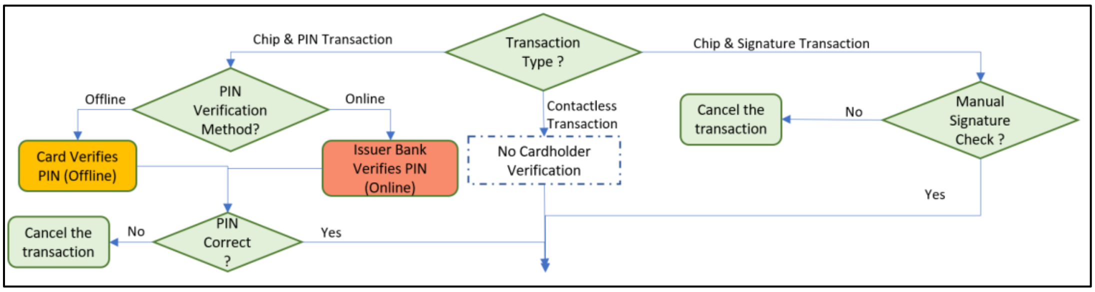


Figure 2-3: EMV Cardholder Verification Methods (CVM)

Cardholder Verification by Personal Identification Number (PIN): In this kind of CVM, the cardholder needs to enter his own PIN into the PIN pad on the POS/ATM to verify himself as genuine cardholders. This verification method is used in case of the EMV chip & PIN transactions where the PIN could be validated by either the IB in case of the EMV online transactions or by the EMV card itself in case of The EMV offline transactions [44].

In the case of the EMV online transactions, the POS/ATM encrypts the PIN entered by the cardholder using the IB public key (IB_{pk}) stored on the EMV card. The encrypted PIN is then sent to the IB in order to validate it. While, in case of the EMV offline transactions, the POS/ATM encrypts the PIN entered by the cardholder by the card's public key (C_{pk}) stored on the card. Then, the encrypted PIN is sent to the EMV card in order to validate it [43].

Cardholder Verification by Signature: The second CVM that is supported by the EMV payment protocol is cardholder verification by signature. To process this kind of CVM, the POS must print the receipt and the cardholder should sign it. Then, the person who is working at the POS must compare the signature of the cardholder which is already signed on the back of the EMV card with the fresh signature that the cardholder is asked to do it at the transaction time. If both signatures are matched, then the POS's attendant should press a button on the POS to indicate that the signature checked and approved. Otherwise, the POS's attendant might reject the transaction [29].

No Cardholder Verification: The EMV payment protocol does not require any CVMs in case of processing EMV contactless card transactions [34]. That's because the payment protocol intends to make such payments easier and faster to process in comparison with other types of payment methods such as chip & PIN and chip & signature. As well, another reason is that in this kind of payment, the transaction value is restricted to a low value such as for instance the EMV contactless payment is restricted to £30 and recently to $45.

The missing cardholder verification in the EMV contactless card transactions opened the door quit wildly to several attacks that chip & PIN transactions are not vulnerable to. Examples for these attacks are skimming and relay attacks. Therefore, we aim in this thesis to bring back association to link between the cardholders and the EMV cards when EMV contactless card transactions are processing.

2.1.2.3.Transaction Authorisation

The main aim of the EMV payment protocol third stage is to approve or decline the transactions. This decision could be decided by either the IB or the EMV card itself by generating the Application Cryptogram (AC) [34]. The EMV payment protocol supports four different types of ACs as following:

Transaction Certificate (TC): This type of AC is generated by the EMV card in order to approve the transaction. The TC includes all the dynamic Transaction Data (TD) such as the transaction amount, transaction currency, transaction date and time, Unpredictable Number (NU) which is generated by the EMV card at each transaction and other information depending on the transaction types. Along with all this dynamic transaction information, the TC included the EMV card static information (SI) such as the PAN and the card expiry date. All these dynamic and SI are signed by the EMV card RSA private key (C_{sk}). Then the signed TC is sent to the IB as proof that the transaction was processed and approved by the genuine EMV card itself. Later, the IB will validate the transaction by comparing the POS/ATM transaction information against the singed version of the TC which generated by the EMV card.

Application Authentication Cryptogram (AAC): This type of AC is generated by the EMV card to decline the transactions.

Authorisation Request Cryptogram (ARQC): This type of AC is generated by the EMV card in the online transaction to request the IB to approve or decline the transaction.

Authorisation Response Cryptogram (APQC): This type of AC is generated by the IB in the response of ARQC. When the IB receives the ARQC, the IB starts to verify the received ARQC by using the EMV card RSA public key (C_{pk}). Then, if the ARQC is verified, the IB checks if there is enough money to cover the transaction and checks whether the EMV card reported as stolen/ lost. If all the checks are passed positively, the IB approves the transaction. Otherwise, the transaction is declined.

The POS/ATM sends the generated AC command during the EMV transaction to the EMV card to request the AC, where the requested AC could be either TC, ARPC or ACC. The EMV card responds with the same requested AC or a lower AC. Table 2-2 shows the EMV payment protocol specification for AC requesting and generating.

Table 2-2: Application Cryptogram Requesting and Responding

#	Requested AC by POS/ATM	Response AC by the EMV card
1	TC	TC
		ARQC
		AAC
2	ARQC	ARQC
		AAC
3	AAC	AAC

2.1.3. EMV Contactless Card Transaction

The EMV contactless cards are based on the NFC technology [45][46]. The NFC is a short-range wireless technology based on RFID and the standard ISO 14443 [47]. The power is transferred from an active reader/POS that has its own power supply such as a battery to a passive NFC tag by using Faraday's principle of magnetic induction [48]. The reader generates a magnetic field in its proximity by passing a large current through its coil then a tag will develop a voltage across the tag's coil when this tag enters in the proximity of the reader and then this voltage is used to power up the NFC tag [49]. Figure 2-4 shows the concept of RFID technology.

As detailed in Section 2.1, the EMV contactless card payment specifications have seven different kernels. Both kernels 3 and 7 support the FDDA for the EMV offline contactless card transactions only. This means both kernels allow the EMV contactless cards to approve or decline the contactless transactions by the card itself without the need to go back to the IB for transaction authorisation. However, in case of the POS requests the EMV contactless card for an online authorisation or the number of offline contactless card transactions is exceeded, then the EMV contactless card processes chip & PIN EMV transactions instead of the EMV contactless card transactions. On the other hand, all the rest kernels (1,2,4,5 and 6) support the three types of the EMV card authentication methods (SDA, DDA, and CDA) and all these five kernels support both the EMV online and offline contactless card transactions. As the EMV contactless card could approve or decline the transactions by itself in case of the offline transactions or it could request the IB to make an online transaction authorisation in case of the online transactions.

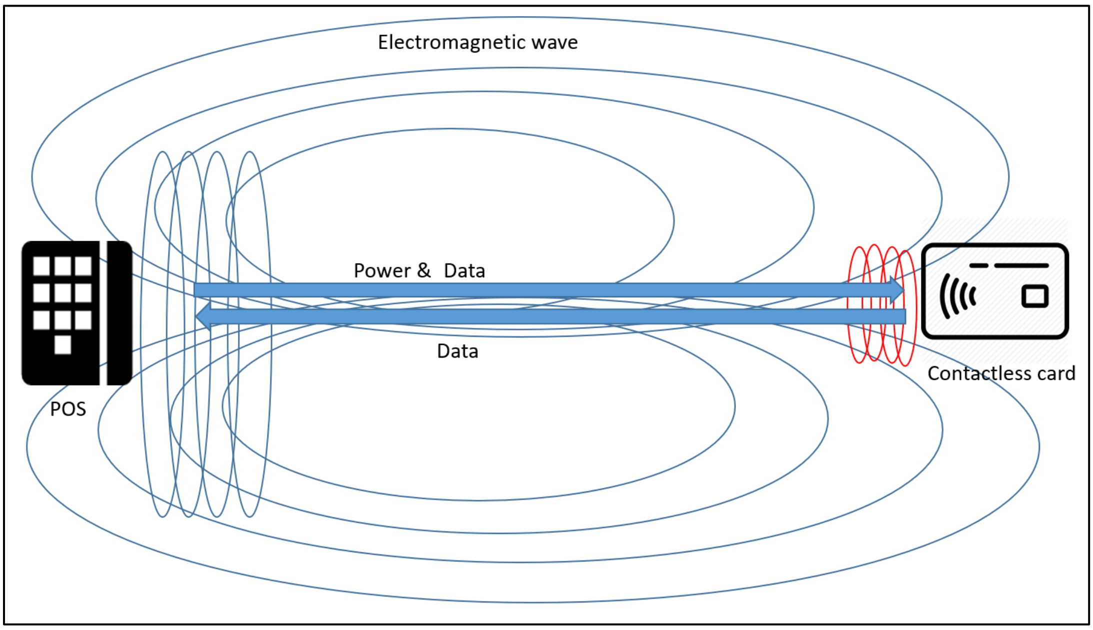


Figure 2-4: RFID Technology Concept

2.1.3.1.Online Contactless Transactions

Figure 2-5 shows the EMV online contactless card payment transaction sequences. Starting with the initiating of the transaction by entering the TD such as the transaction amount, transaction time, transaction date and transaction currency. Then, the POS waits for a contactless card to be presented in its NFC field (steps 1 & 2). When the cardholder places his EMV contactless card into the POS, the POS starts to send its APDU commands and receives the card's APDU responses (step 3).

The first APDU command is to request all the NFC Application Identification (AID) that are supported by the card. Then, the EMV contactless card responses with all the supported NFC AID as shown in Figure 2-5 steps 4 &5. Then, the second pair of APDU command and response are to select the requested AID (steps 8 & 9). Next, after successfully selecting the AID, the POS sends the Get Processing Options (GPO) APDU command to the EMV contactless card in order to request all the information that the card required to complete the transaction. Then the card responses with the

Processing Data Options List (PDOL) and Application File Locator (AFL) as shown in steps 10 and 11 in the same Figure. The next APDU command is the "Read Record" where the POS reads all the records that are showed in the AFL as shown in steps 12 and 13.

Then, the POS prepares the requested PDOL information such as transaction amount, currency, date, time and UN as shown in steps 14. The next APDU command-response pair is the "Generate AC". The EMV contactless card generates and signs both the DDA/CDA and the ARQC and sends them back to the POS. The card also updates the transaction counter and NFC counter as shown in step 16. Finally, the card can be removed from the POS's NFC filed. Then, the POS requests the transaction authorisation from the IB to approve or decline the online contactless card transaction by validating the ARQC and other information.

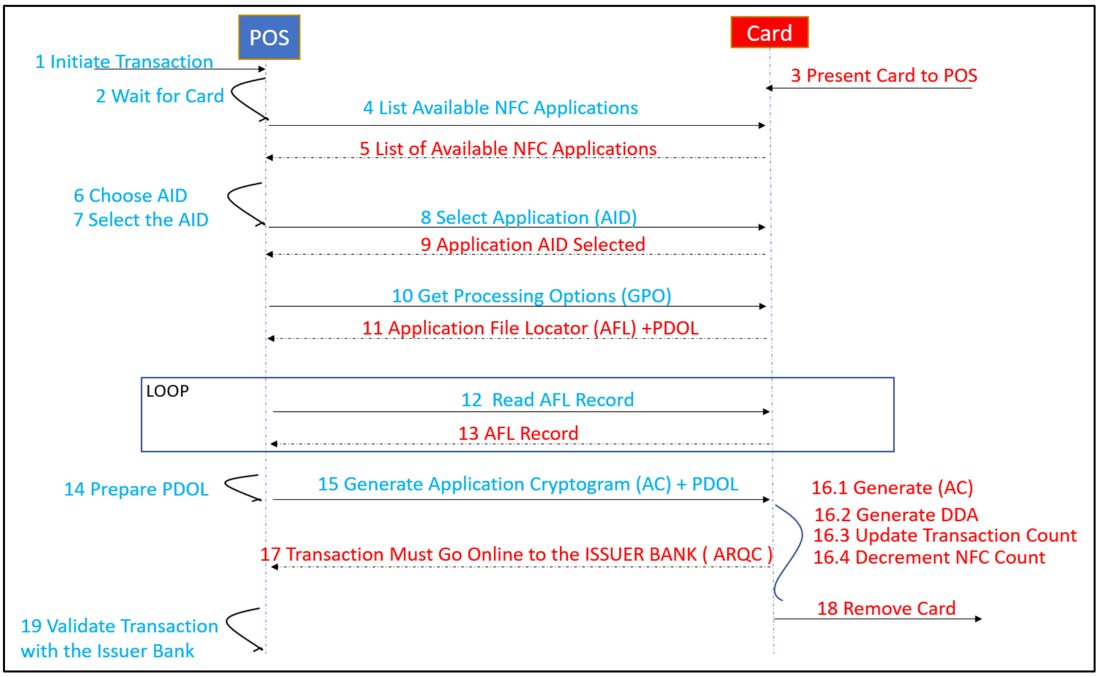

POS
Card
1 Initiate Transaction
2 Wait for Card
3 Present Card to POS
4 List Available NFC Applications
5 List of Available NFC Applications
6 Choose AID
7 Select the AID
8 Select Application (AID)
9 Application AID Selected
10 Get Processing Options (GPO)
11 Application File Locator (AFL) +PDOL
LOOP
12 Read AFL Record
13 AFL Record
14 Prepare PDOL
15 Generate Application Cryptogram (AC) + PDOL
16.1 Generate (AC)
16.2 Generate DDA
16.3 Update Transaction Count
16.4 Decrement NFC Count
17 Transaction Must Go Online to the ISSUER BANK (ARQC)
18 Remove Card
19 Validate Transaction
with the Issuer Bank

Figure 2-5: EMV Contactless Card Online Transaction Sequences

2.1.3.2.Offline Contactless Transactions

Figure 2-6 shows the EMV offline contactless card payment transaction sequences. All the APDU commands and responses are exactly the same as shown in the online ones. However, the order of the APDU commands differs as shown in Figure 2-6. This kind of contactless card transaction is supported by both kernels 3 and 7. However, the rest kernels support both online and offline contactless card translations. The number of offline contactless card transactions is limited to the contactless floor limit transaction as explains in the next subsection.

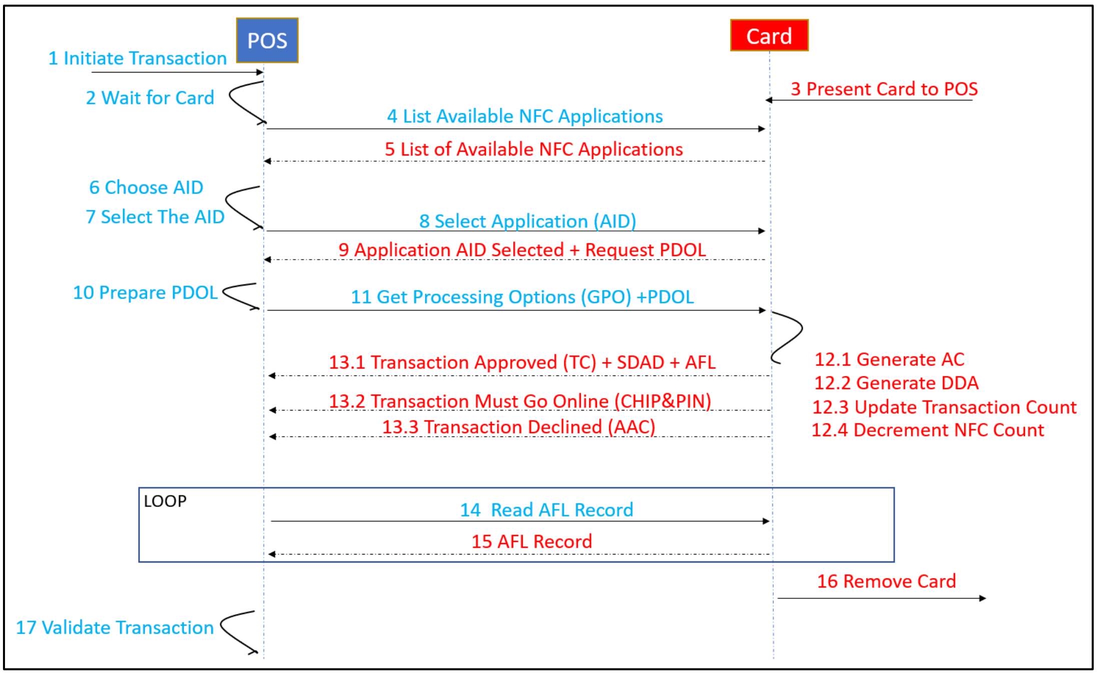

POS
Card
1 Initiate Transaction
2 Wait for Card
3 Present Card to POS
4 List Available NFC Applications
5 List of Available NFC Applications
6 Choose AID
7 Select The AID
8 Select Application (AID)
9 Application AID Selected + Request PDOL
10 Prepare PDOL
11 Get Processing Options (GPO) +PDOL
12.1 Generate AC
12.2 Generate DDA
12.3 Update Transaction Count
12.4 Decrement NFC Count
13.1 Transaction Approved (TC) + SDAD + AFL
13.2 Transaction Must Go Online (CHIP&PIN)
13.3 Transaction Declined (AAC)
LOOP
14 Read AFL Record
15 AFL Record
16 Remove Card
17 Validate Transaction

Figure 2-6: EMV Contactless Card Offline Transaction Sequences

2.1.3.3.Contactless Floor Limit Transaction

Both the POSs and EMV contactless cards have their own specific contactless floor limit transaction that can be determined by either the AB in case of the POSs or the IB in case of the EMV cards. The POS contactless floor limit transaction value of the transaction that is determined by the AB where any contactless transactions above this value force the POS to go to an online contactless transaction. While the POS process the transaction in the offline mode if the transaction value is below the POS contactless floor limit transaction [7].

The card contactless floor limit transaction could be a value of contactless transactions or a number of contactless transactions where in both cases the floor limit is determined by the IB. The contactless floor limit value for the EMV cards is the same concept of the POS contactless floor limit transactions where if the transaction amount value above the determined value, then the transaction must go online to be authorised by the IB. Else, the transaction process in the offline transaction.

On the other hand, in case the contactless floor limit is a number, the IB set two numbers where the first number represents the maximum number of contactless card transactions in the offline case process respectively without entering the PIN by the cardholder. While the second number represents the maximum number of contactless card transactions in the offline case process respectively without entering the PIN by the cardholder. Most of the IBs sets the online floor limit number higher than the offline number because of the extra checks that are done by the IB during the online contactless transactions. In case any of these numbers are exceeded, then the transaction must be processed in chip & PIN instead of contactless transaction and the counters of these numbers are set to zero again.

2.2. Literature Review on the Security of Payment Contactless Cards

This section consists of two parts where the first part details the vulnerabilities and attacks on the EMV contactless cards while the second part details the countermeasures in the literature review to prevent the documented attacks.

2.2.1. Vulnerability Analysis of EMV Contactless Cards

The EMV contactless card transactions can be subjected to several different attacks due to a number of vulnerabilities resulting from flaws in the EMV payment protocol itself or due to the wireless connectivity between the POSs and contactless cards. The next subsections detail these vulnerabilities and categorise them.

2.2.1.1.Connectivity Vulnerabilities

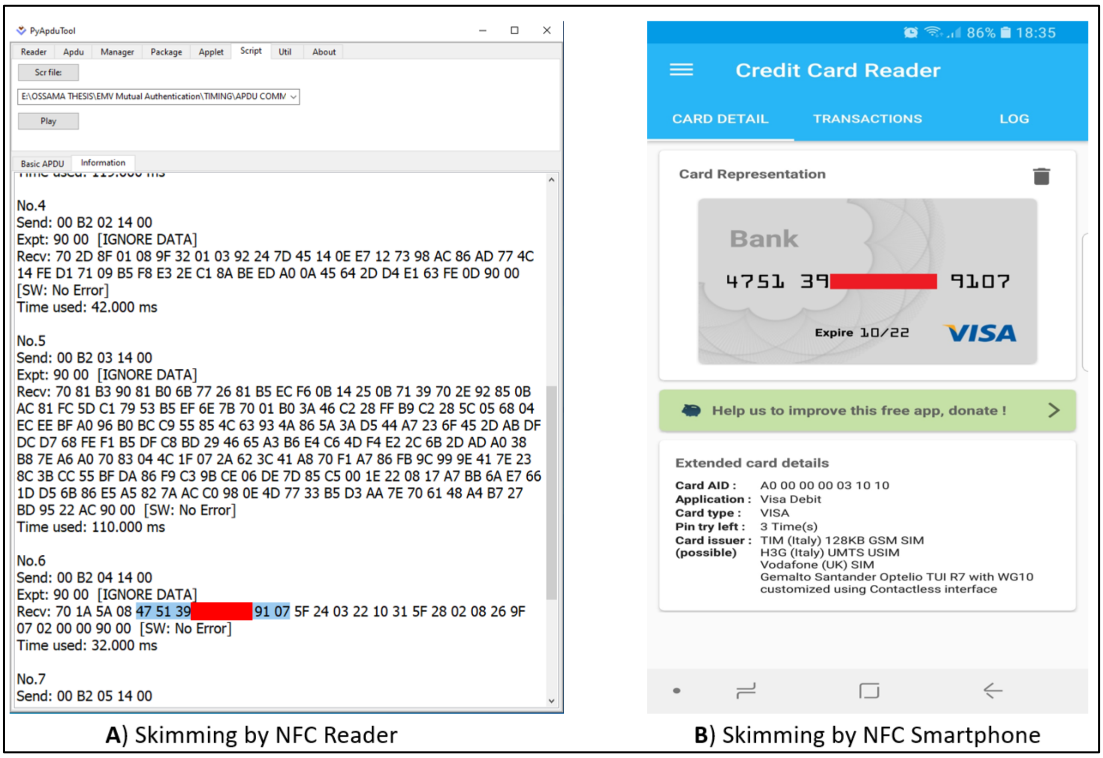
The following attacks are mainly possible to be launched against the EMV contactless cards due to the wireless connectivity between POSs and the EMV contactless cards

Skimming Attack

In case of skimming attack, the fraudsters could use on-the-shelf hardware and software to obtain the EMV card's information such as the PAN, cardholder name and expiry date of the card. The fraudsters could carry out the skimming attack on any EMV contactless cards without the knowledge of the genuine cardholder while the card is still in the cardholder's wallet [50][51][52][53].

We tested the idea in two different scenarios. The first was based on using readily available hardware called ACR122 NFC reader while the second was simply by installing an app called “Credit Card Reader” freely available on both Google Play and Apple App Store as shown in Figure 2-7.

In both scenarios, we were able to obtain these data to do online shopping with some websites that do not require the Card Verification Code (CVC) which is the 3-digit or 4-digit number which written on the back of the card. Moreover, our investigations concluded that these sensitive data along with other information such as the IB public key certificate are enough to create a counterfeit card to trick the POS to process a transaction in case of offline contactless card transactions as will detail in Section 4.2.2.



E:\OSSAMA THESIS\EMV Mutual Authentication\TIMING\APDU COMM

```
No.4
Send: 00 B2 02 14 00
Expt: 90 00  [IGNORE DATA]
Recv: 70 2D 8F 01 08 9F 32 01 03 92 24 7D 45 14 0E E7 12 73 98 AC 86 AD 77 4C
14 FE D1 71 09 B5 F8 E3 2E C1 8A BE ED A0 0A 45 64 2D D4 E1 63 FE 0D 90 00
[SW: No Error]
Time used: 42.000 ms

No.5
Send: 00 B2 03 14 00
Expt: 90 00  [IGNORE DATA]
Recv: 70 81 B3 90 81 B0 6B 77 26 81 B5 EC F6 0B 14 25 0B 71 39 70 2E 92 85 0B
AC 81 FC 5D C1 79 53 B5 EF 6E 7B 70 01 B0 3A 46 C2 28 FF B9 C2 28 5C 05 68 04
EC EE BF A0 96 B0 BC C9 55 85 4C 63 93 4A 86 5A 3A D5 44 A7 23 6F 45 2D AB DF
DC D7 68 FE F1 B5 DF C8 BD 29 46 65 A3 B6 E4 C6 4D F4 E2 2C 6B 2D AD A0 38
B8 7E A6 A0 70 83 04 4C 1F 07 2A 62 3C 41 A8 70 F1 A7 86 FB 9C 99 9E 41 7E 23
8C 3B CC 55 BF DA 86 F9 C3 9B CE 06 DE 7D 85 C5 00 1E 22 08 17 A7 BB 6A E7 66
1D D5 6B 86 E5 A5 82 7A AC C0 98 0E 4D 77 33 B5 D3 AA 7E 70 61 48 A4 B7 27
BD 95 22 AC 90 00  [SW: No Error]
Time used: 110.000 ms

No.6
Send: 00 B2 04 14 00
Expt: 90 00  [IGNORE DATA]
Recv: 70 1A 5A 08 47 51 39[redacted]91 07 5F 24 03 22 10 31 5F 28 02 08 26 9F
07 02 00 00 90 00  [SW: No Error]
Time used: 32.000 ms

No.7
Send: 00 B2 05 14 00
```

A) Skimming by NFC Reader

4751 39[redacted]9107

Expire 10/22

Extended card details

Card AID : A0 00 00 00 03 10 10
Application : Visa Debit
Card type : VISA
Pin try left : 3 Time(s)
Card issuer : (possible) TIM (Italy) 128KB GSM SIM
H3G (Italy) UMTS USIM
Vodafone (UK) SIM
Gemalto Santander Optelio TUI R7 with WG10 customized using Contactless interface

B) Skimming by NFC Smartphone

Figure 2-7: Examples of Skimming Attack by both NFC Reader & Smartphone

Relay Attack

Ghost and Leech attack or the Mafia attack is another name for the relay attack [54]. To perform a relay attack on the EMV contactless card, it required two attackers with two NFC devices (any NFC enabled smartphone). The first NFC device is called the Proxy that fakes a card to a genuine POS while the second device is called the Mole that fakes a POS to a genuine card [55][56]. In such an attack, the attackers can pay for any goods or services at any genuine POS by using the genuine EMV contactless card that is still inside the wallet of the victim without the victim's knowledge, while the victim could be located far away from the POS. Figure 2-8 shows all the parties that involve performing the relay attack on the EMV contactless cards. As shown in Figure 2-8 that both the Proxy and the Mole could be connected by Bluetooth and Wi-Fi. When the relay attack is successfully performed by the fraudster on the EMV contactless card, the attack will defeat one of the rules of the contactless cards payment that says that an EMV contactless card should be within 10 cm from POS.

Banks and financial companies consider that the relay attack as an attack requires specialized hardware and very high skilled attackers to perform such an attack. Therefore, banks and financial companies report that relay attack is not possible to happen in real life and they said, ‘there has been no example of it happening in the real world and we find it highly unlikely that it will happen’ [56]. Surprisingly, a number of researchers such as J. van den Breekel and M. Roland approved that the opinion of banks and financial companies about relay attacks is wrong by implementing a practical relay attack on the EMV contactless card by suing Off-the-shelf hardware [57][58]. They implemented the relay attack on the EMV contactless card by using two NFC smartphones and straightforward android application to relay APDU commands from the genuine POS to the genuine EMV card through both the proxy and the mole and relay the APDU response from the EMV card back to the POS through both the proxy and the mole [59].

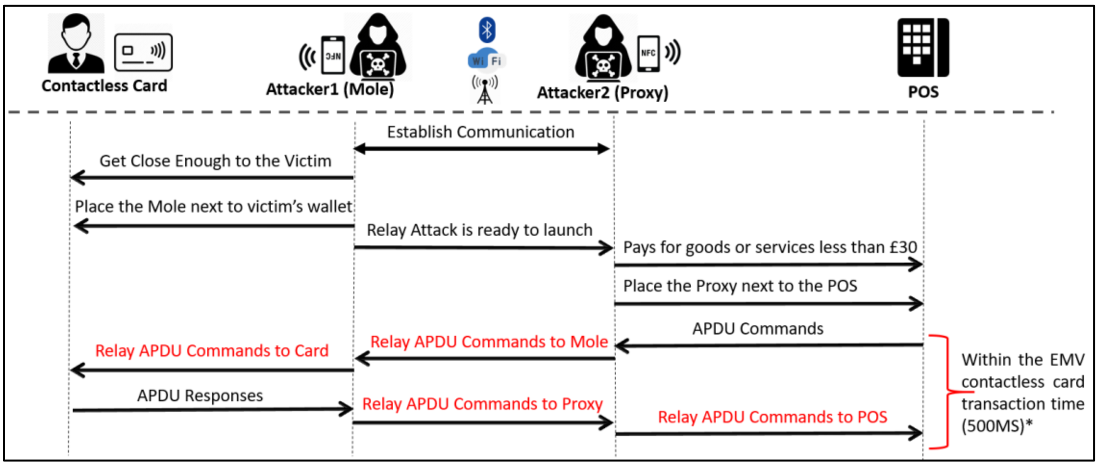

Contactless Card
Attacker1 (Mole)
Attacker2 (Proxy)
POS
Establish Communication
Get Close Enough to the Victim
Place the Mole next to victim's wallet
Relay Attack is ready to launch
Pays for goods or services less than £30
Place the Proxy next to the POS
APDU Commands
Relay APDU Commands to Mole
Relay APDU Commands to Card
APDU Responses
Relay APDU Commands to Proxy
Relay APDU Commands to POS
Within the EMV contactless card transaction time (500MS)*

Figure 2-8: Relay Attack Setup

Time and distance are the main limitations of the relay attacks. Successful relay attack on the EMV contactless cards should not take a long time to do a transaction comparing with normal EMV contactless card transactions [60]. While the second limitation is the distance between the EMV contactless card and the POS should be within 10 cm, in this case, the attacker might be raised suspicion by the genuine cardholder.

There are several proposed approaches to overtake these two limitations in relay attacks. J. van den Breekel was one of the first researchers to who tackled successfully the first limitation of the relay attacks which is the time as shown in Figure 2-9. The proposed method succeeds in reducing the relay attack time close enough to the normal EMV contactless transaction time by caching all the static commands that sent by the POS and all the static responses that sent by the EMV contactless card [57].

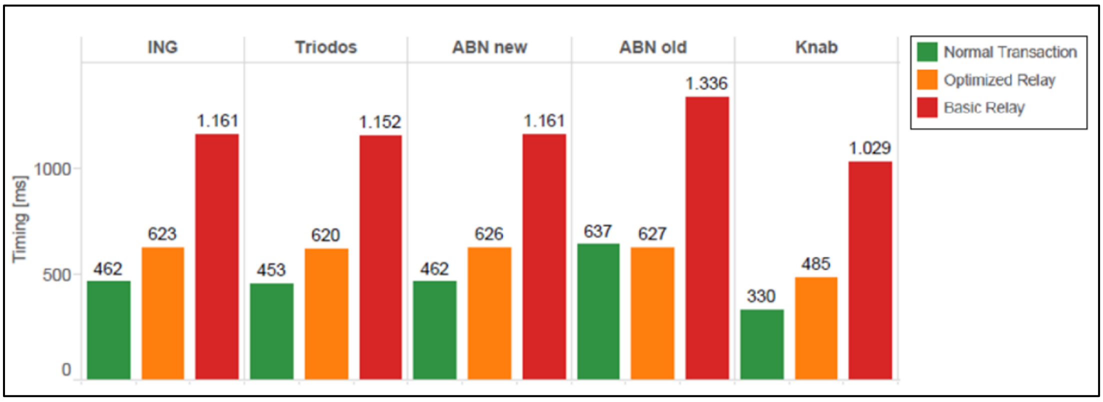


Figure 2-9: Relay Attack Timing, adapted from [57]

Both of Z. Kfir and A. Wool proposed another work successfully to increase the distance between the Mole and the victim's EMV contactless card by generating a stronger electromagnetic field by using a bigger antenna to maximize the distance between the fraudster and the victim in order to no raise any suspicion when preforming the relay attack [56].

Stolen/ Lost EMV contactless Card

If a cardholder's EMV chip & PIN card (contact) is stolen or lost, the cardholder must report it to his own IB to block the card and a new card should be issued to the cardholder with new and different PAN, expiry date and CVC. The fraudsters will not be able to use the stolen/lost EMV chip & PIN card as they do not know the PIN. However, they could use this card to make online purchases by using all the information printed on the stolen/lost EMV contact card. this could work until the cardholder reports the stolen/lost card to his own IB.

On the other hand, the story is different in case of the stolen/lost card is EMV contactless card. Several news reported that stolen/lost EMV contactless cards could be used by fraudster months after cancellation [61][62]. That's means, even if the cardholder reports to his IB that his EMV contactless card has been stolen/lost, and the IB cancels the card and issues a new one to the cardholder. It is still possible for the person who has stolen/found the EMV contactless card to make contactless transactions. As in such transactions, the fraudsters do no need the PIN in order to make such transactions. Besides, the transaction should be an offline transaction to make it possible to use the stolen/lost EMV contactless card even after cancellation. As in case of the offline transaction the EMV contactless card approves/declines the transaction without the knowledge of the IB. These two points made this kind of attack possible on the EMV contactless cards. Figure 2-10 shows the value of lost/stolen cards fraud from 2010 until 2019 according to the UK Finance [18]. As almost 95 million were lost in just 2019 due to this kind of attacks.

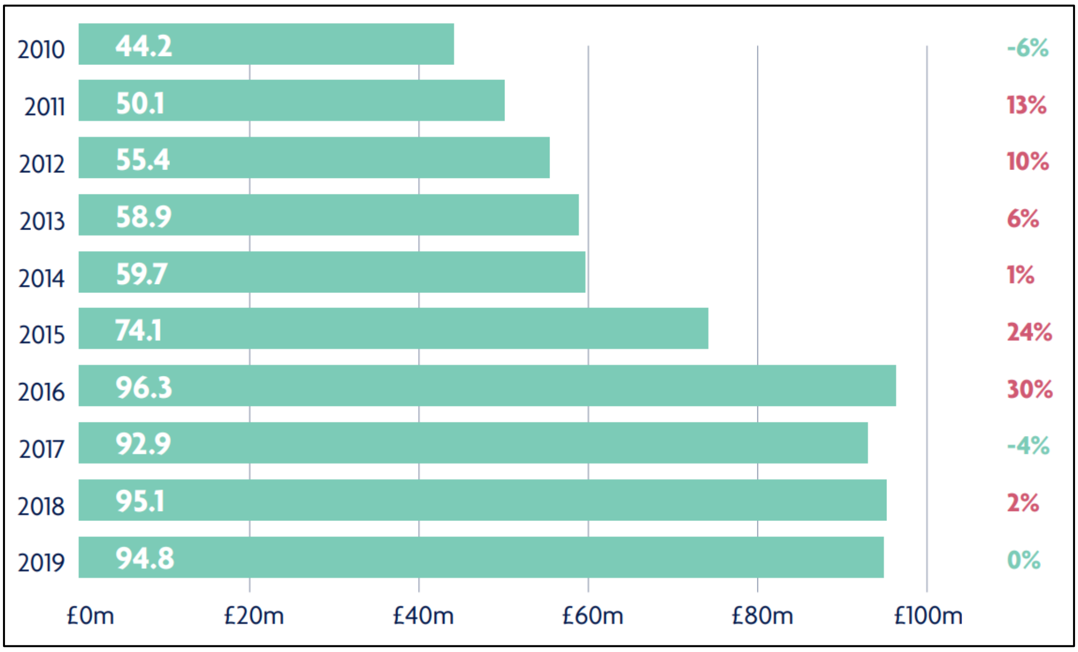


Figure 2-10: Lost/Stolen Cards Fraud Losses, adapted from [18]

2.2.1.2.EMV Payment Protocol Vulnerabilities

The EMV payment protocol designed to make sure that all the EMV cards are accepted by any POS/ATM around the world. This makes this payment protocol very complex protocol with various optional processing that could be chosen by both EMV cards and POS/ATM during the transaction time in order to process any kind of transaction.

For example, the EMV cards support online/offline transactions, three different card authentication methods (SDA, DDA, and CDA), several CVMs such as online PIN, offline PIN, signature and no CVM in case of contactless card transactions. All these different options in the EMV payment protocol open the door to downgrade attacks. In such attack, the POS or the EMV card could be fooled by an attacker who is able to change and modify both POS and EMV cards capabilities. This attack (downgrade) gained a massive advantage in the wireless connectivity in the EMV contactless card transactions.

EMV cards designed to become very hard to clone by attackers in comparison with magnetic stripe cards. This is due to the use of powerful encryption algorithms in such cards while in case of the magnetic stripe cards, such encryption algorithms are not used. However, both M. Roland, A. Usenix in [63] and M. Bond, O. Choudary in [64] demonstrated successfully that the EMV contactless cards are still suffering from cloning attacks due to the ability of altering the capabilities of such cards and fooling the POS by changing the AIP of the EMV cards to indicate that the EMV card supports only magnetic stripe contactless transaction instead of the EMV contactless card transactions.

The cloning attack gained advantage from the limited number of the Unpredictable Number (UN) which is provided by the POS in case of magnetic stripe contactless transaction. The UN used in such transactions in order to prevent replay attacks. However, the authors found out that the UN is limited to range from 0 to 999. This make it possible to the attacker to request all the possible digital signature in the Compute Cryptographic Checksum APDU command. The cloning attack starts by skimming the genuine EMV contactless card by the use of NFC enabled smartphone in order to collect and build a table to all the possible digital signatures by sending 0-999 of the UN from the NFC enabled smartphone. Then, the collected data (table) used to create a functional clone card. Later, the created card fools the genuine POS by sending AIP with value of “0000” to manipulate the POS to process magnetic stripe contactless transaction instead of EMV contactless card transaction. When the POS sends its UN, the created clone card checks the stored table and sends the right responses back to the POS.

This cloning attack consists of two main attacks which are the pre-play and downgrade attacks. The pre-play attack took advantage of skimming the genuine EMV contactless card without the cardholder’s knowledge and collect all the required data in order to create a functional clone card. On the other hand, the downgrade attack took advantage form the complexity of the EMV payment protocol and where the AIP can be change by the attackers in order to fool the POS to process less secure transaction than the EMV mode.

One of the most obvious functions of the EMV contactless card transactions is the no CVM is required due to the main aim of the EMV payment protocol to make such transactions easier and faster and more convenient to be processed by cardholders. However, M. Emms, B. Arief and A. Van Moorsel in [65] discovered that EMV contactless cards still allow some sort of offline PIN verification in such transactions, where the attacker could use any NFC enabled reader/smartphone to verify the PIN without the cardholder’s knowledge by manipulating the behaviour of the EMV contactless card. The attacker sends the Get Data APDU command to discover how many PIN attempts left. Then, the attacker sends his guessing of the PIN to the EMV contactless card by sending Get challenge APDU command as shown in Figure 2-11.

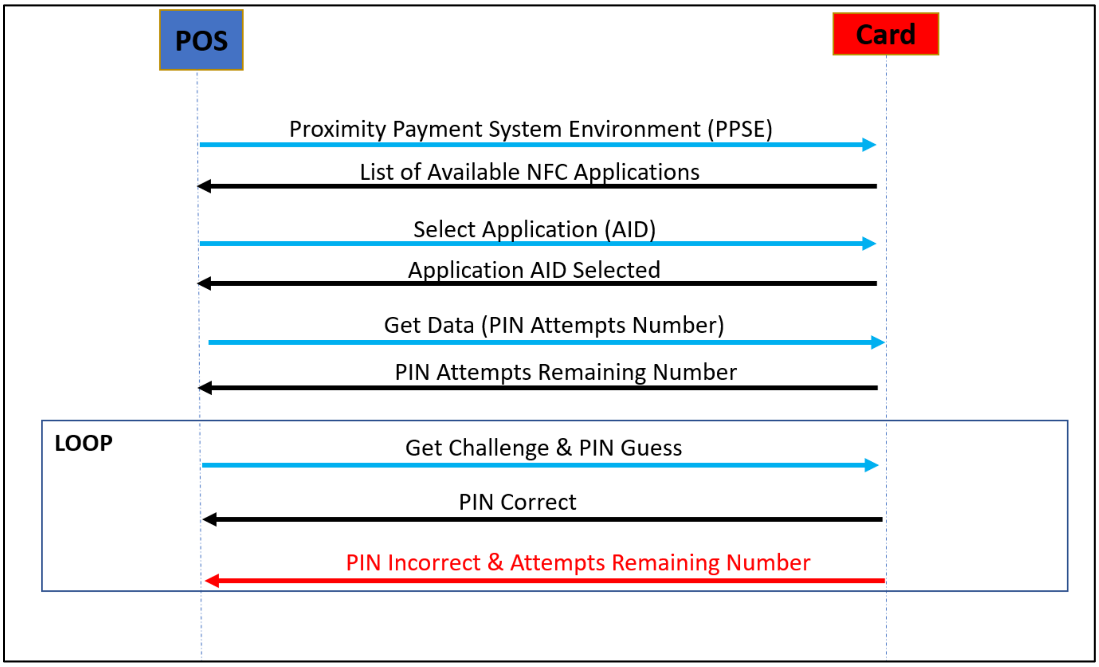

POS
Card
Proximity Payment System Environment (PPSE)
List of Available NFC Applications
Select Application (AID)
Application AID Selected
Get Data (PIN Attempts Number)
PIN Attempts Remaining Number
LOOP
Get Challenge & PIN Guess
PIN Correct
PIN Incorrect & Attempts Remaining Number

Figure 2-11: PIN Verification Attack on the EMV Contactless Cards

The card verifies the PIN provided by the attacker and if both PINs are matched. Then the card sends a confirmation APDU response that the PIN is verified. Otherwise, the

EMV contactless card sends back APDU response with the PIN incorrect and update the PIN attempts. This allows fraudsters to make a limited number of PIN guessing until the PIN attempts are finished. This kind of attack approved our thought of getting advantage of the complexity of the EMV protocol with all these options which are available to both POSs and EMV contactless cards to fool POS/ EMV cards or both to serve the attacker purposes.

As we mentioned previously that the EMV contactless card transactions are limited with a maximum of £45 GBP in the UK. However, M. Emms, B. Arief and A. Van Moorsel discovered in [66] that the EMV payment protocol allows doing far away more than the limited value in foreign currency by the EMV contactless card transaction without the cardholder’s PIN as stated in the next attack which is called the harvesting attack, where the EMV contactless cards vulnerable to this attack which allows the attackers to do unlimited value of contactless transactions in foreign currency transactions for any amount without the need of PIN. This attack overtakes the fixed limited value of EMV contactless card transactions. There are several vulnerabilities that helped to launch this specific attack such as the merchant details are not included in the TD that is signed and sent by the EMV card to the IB.

The second point is the wireless connectivity of such cards that allows the attacker to engage with the EMV contactless card while the card still at the cardholder wallet without his knowledge. The harvesting attack consists of two stages: collecting transaction information and converting the transaction information into money. In the first stage, the attacker uses his NFC enabled smartphone to communicate with the victim’s EMV contactless card by an application that developed to launch this specific attack. The used NFC application has been fixed to a set value and currency (value more than £30 in foreign currency). Then, the NFC application starts communicating with the victim’s EMV contactless card, the victim’s card responses with generating AC. At the end of the first stage the NFC application stores this information and sends it to a rogue merchant in order to start the second stage.

In the second stage, the criminal already opened a rogue merchant account with specific AB. Then, the criminal adds the merchant’s information (merchant ID, terminal ID, merchant’s bank account details) to the transaction information collected in the first stage. When the transaction information (card’s transaction information &

merchant’s information) is completed, the whole information sends by the merchant to the AB and then to the IB to authorise the transaction as offline contactless card transaction.

We do believe that this kind of attack is possible in theory due to the vulnerable point in the EMV contactless card specifications. However, making this attack in a real scenario is too risky for the attacker as it is too easy to find out the rogue merchant that is already registered with the AB. This attack could be prevented by either making the EMV contactless cards do online transactions instead of offline transactions in case of a foreign currency or by forcing these cards to do chip & PIN transaction in case of transaction foreign currency.

2.2.2. Literature Review on the Countermeasures for the EMV Contactless Cards Transactions

As discussed earlier, the most vulnerable point in the EMV contactless cards is the leakage of information to unauthorised NFC enabled readers/smartphones. As most of the discussed attacks in the earlier section gained a huge advantage from this vulnerable point. Here, we detailed most of the documented countermeasures.

Several countermeasures have been suggested to prevent reading the EMV contactless cards by unauthorised NFC enabled readers/smartphones to prevent skimming and relay attacks. One of the earliest ideas was proposed by L. Hong, H. C. Yong, and Q. H. Zhang in [67] which relying on an RFID blocking wallet that designed to stop any RFID and NFC signals to communicate with EMV contactless cards while it is inside the cardholder‘s RFID blocking wallet.

Furthermore, two solutions introduced the idea of turning the EMV contactless cards into an active card with a built-in battery instead of the original passive cards (battery less). The first method was proposed by both M. Emms and A. van Moorsel in [51] and J. W. Yum in [68] which was based on using cards with an activation button where the cardholder needs to push the activation button to activate the card and allow the contactless transaction. The cardholder then needs to deactivate the card after the transaction is performed. The second mothed suggested the use of a built-in light sensor on the EMV contactless cards. Once the card is exposed to the light (outside the cardholder’s wallet), the light’s sensor activates the card while if the card inside the cardholder’s wallet, the card is always deactivated. When a genuine cardholder needs to do a transaction, the sensor activates the card and allows it to communicate with the POS [69]. In both methods, reading the EMV contactless cards by unauthorised NFC readers/smartphones is not easily possible. However, the main limitation of the above two methods is the battery’s life span i.e. when the card’s battery is dead, the battery needs to be replaced or recharged.

Another approach was presented by MasterCard to use a fingerprint built-in sensor to prevent unauthorized NFC enabled readers/smartphones from obtaining the EMV contactless cards sensitive information. In this method, the cardholders should place their fingerprint on the built-in sensor and place the EMV card next to the POS. The POS then powers up the EMV card and a fresh fingerprint data is compared with the stored template (stored at the personalization phase). If both fingerprints are matched, the EMV contactless card is activated [70].

Moreover, there are two countermeasures to prevent relay attacks on the EMV contactless cards that do not require significant change to the EMV contactless card itself. The first proposed was proposed by C. H. Kim and G. Avoine in [71] and S. Drimer and S. J. Murdoch in [72] which is called distance bounding protocols that depends on one assumption which both POS and the EMV card share a secret between them, then the POS measures the time to send this secret between the POS and the EMV card during the contactless card transaction. And based on accurate time which measures the level of nanosecond and the knowledge of the speed of the light, the POS can estimate the distance EMV card. Distance bounding protocol assumes that the attackers can send relay messages close to the speed of light. This could be possible for attackers to do if and only if, they used specialized and very expensive hardware [73][74][75].

The second countermeasure is time bounding protocols which was proposed by J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji in [76] . The first of these protocols was attempted to set an overall time to the whole EMV contactless card transaction. EMV contactless specifications set a specific time to complete the whole transaction within 500 ms, However, some genuine EMV contactless card transactions take more than 500 ms. Furthermore, optimizing the relay attack time by caching the static commands and responses reduces the time to less than 500 ms as explained in [57]. Hence, the overall time protocol cannot be used to stop the relay attack on EMV contactless cards.

The Pay safe was suggested by T. Chothia, F. D. Garcia and J. De Ruiter in[77] to prevent the relay attacks on the EMV contactless cards. The main idea was to count the required time for exchanging nonce between the POS and the EMV contactless card during the “GPO” APDU command and response. However, this kind of counting the time of exchange nonce between the EMV contactless cards and the POS is depended on several factors such as the EMV contactless card chip’s capacities which are different between each IB. Also, the way that the cardholder places his EMV contactless card next to the POS affecting the time of transactions and exchanging the APDU commands and responses. Therefore, the time bounding countermeasures could end up by rejecting genuine transactions.

Some other documented countermeasures are detailed in the next chapters due to their relevance to our contributions.

2.3. Summary of the Chapter

The chapter detailed the EMV payment protocol and showed the main involved players in such a protocol. Furthermore, it gave an overview of the different specifications that this payment protocol is offered. Moreover, the chapter explained the three main phases of the EMV payment protocol and showed the EMV contactless card transactions. Furthermore, a literature review of the EMV contactless cards vulnerabilities, different kinds of attacks and countermeasures are detailed in this chapter.

In chapter two, we conclude that the EMV contactless cards suffer from several kinds of attacks due to either the wireless connectivity which is introduced in such transactions between the POS and EMV contactless cards. Or due to the missing of the CVMs in such transactions where there are no associations between the cardholder and his EMV contactless card. Therefore, more works need to be done to improve the security of the such transactions and prevent these detailed attacks.

Our attempts to improve the security of such transactions are categorized into two main levels. The first level is to improve the security of the EMV contactless cards with minimum changes in the original EMV infrastructures as will detail in both Chapters 45, and 6. As in both these chapters, all the required changes are done at the personalisation and transaction phases without changing the EMV payment protocol itself or requiring any extra hardware or even change the cardholder’s experience to process such transaction. In contrast, the second level requires more changes into the original EMV infrastructures and adding extra hardware and for sure changing the cardholder’s experience as will detail in both Chapters 78.

3. Chapter 3: Java Framework for EMV Contactless Cards

The previous chapter presented the background of the EMV payment protocol specifications and the related literature review of the EMV contactless card transactions. We established that such cards have several vulnerabilities due to the wireless connectivity between the POSs and the cards or gaps in the EMV specifications . The vulnerabilities were exploited by various kinds of attacks on contactless cards such as skimming, replay, cloning, relay and other attacks.

As the main aim of the thesis is to improve the security of the EMV contactless cards, we needed to have better understanding of such cards and how they are processing the contactless card transactions. within addition to using genuine EMV contactless cards, there was a need to build our own contactless cards in order to test our proposals.

In this chapter, a Java framework for simulating EMV contactless cards is developed and built in which personalisation, transaction and authorisation phases are implemented to simulate the feasibility of our proposed solutions in the next chapters.

Chapter 3 starts with a description of the APDU and its five commands and responses in the EMV contactless card transactions. Then, it details all the hardware and software required to develop and build the Java framework. Next, the chapter shows the implementations of the three main EMV payment protocol phases. Finally, results, validations, discussions and conclusions are presented.

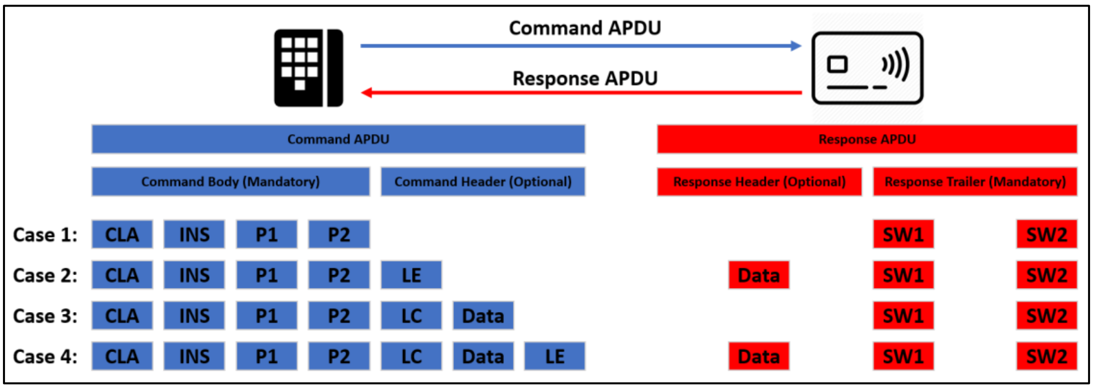
3.1. Application Protocol Data Unit (APDU)

The APDU is the communication protocol used to communicate between POSs/ATMs/NFC readers and all kinds of smart cards whether contact or contactless cards. This communication protocol consists of two different types namely APDU commands and APDU responses, where the APDU commands represent all the messages send by POSs/ATMs/NFC readers/smartphones while the APDU responses represent all the messages send by smart cards [32].

The APDU's structure is defined according to the ISO/IEC 7816-4 [5]. The APDU command consists of two parts where the first part is a mandatory part that is called the command body, which consists of four bytes, namely class of instruction (CLA), instruction code (INS), instruction parameter 1 (P1) and instruction parameter 2 (P2). While the second part is optional and is called the command header. This command header consists of three parts: one-byte representing the length of expected response (LE), one-byte representing the length of command data (LC) and the command data (Data) with size (0-255 bytes).

The APDU response consists of two parts which are the response header (optional) with size from 0 bytes to 255 bytes while the second part is called the response trailer (mandatory) with two bytes for Status Word 1 (SW1) and Status Word 2 (SW2).

There are four cases of APDU commands and responses as shown in Figure 3-1. In the first case, there is no command data and no response data are required whereas in case number 2, there are no command data and there is a need for response data. While in case number 3, there are command data and there is no need for response data. Moreover, in case number 4 both command and response data are required. In our applet development, we used three cases which are 2,3 and 4 to represent the EMV contactless card according to the EMV payment specifications.



	Command APDU							Response APDU		
	Command Body (Mandatory)				Command Header (Optional)			Response Header (Optional)	Response Trailer (Mandatory)	
Case 1:	CLA	INS	P1	P2					SW1	SW2
Case 2:	CLA	INS	P1	P2	LE			Data	SW1	SW2
Case 3:	CLA	INS	P1	P2	LC	Data			SW1	SW2
Case 4:	CLA	INS	P1	P2	LC	Data	LE	Data	SW1	SW2

Figure 3-1: APDU Command & Response Structure

3.2. EMV Contactless Card Transactions APDU Commands & Responses

As shown in both Figure 2-5 and Figure 2-6, there are five APDU commands and responses in order to process the EMV contactless card transactions for both online and offline contactless card transactions. We needed to understand each one of these five APDU commands and responses in order to develop and build our Java framework for the EMV contactless card to duplicate all the functionality of genuine EMV contactless cards.

The EMV contactless card transactions start by initiating the transaction by preparing to enter the transaction's details such as the transaction amount sent by the POS. Then, the POS activities its own RFID antenna and waits for EMV contactless card to be presented in the RFID field. When the cardholder places his EMV contactless card next to the POS, the POS starts the transaction by sending the APDU commands and receiving the card's APDU responses. The next five subsections detail the main five APDU commands and responses between the POS and EMV contactless cards in order to process the contactless card transactions according to the EMV payment protocol specifications [28].

3.2.1. Proximity Payment System Environment (PPSE)

The first APDU command which the POSs send to the EMV contactless cards, is the Proximity Payment System Environment (PPSE) command. The main goal behind this command is to request all NFC applications that are supported by the EMV contactless card to send back to the POS. When the EMV contactless card receives this APDU command, its answers back with all the supported NFC applications by sending the tag number 4F to represent the AID and the AID priority tag number 87 to represent the priority of the AID among the rest if there are any. The "PPSE" APDU command parameters according to the EMV specifications are as shown in Table 3-1.

3.2.2. Select Application (AID)

The second APDU command that the POS sends to the EMV contactless card during contactless card transaction is the "Select AID" command. After processing the "PPSE" APDU command, the POS chooses one of the NFC AIDs that the card supported and was part of the "PPSE" APDU response. Then, the POS sends a request to the EMV contactless card to confirm the selection of the chosen AID. The "Select AID" APDU command parameters are shown in Table 3-1, where the APDU command consists of the chosen AID by the POS and the card should confirm the selection process.

3.2.3. Get Processing Options (GPO)

After agreeing on the NFC AID selection process by the POS and the EMV contactless card in the previous two APDU commands and responses, the next step is to agree on transaction processing options and return the location of the AID specific data stored inside the EMV contactless card.

The "GPO" APDU command which is sent by the POS varies among all the seven different contactless card kernels that the EMV payment protocol is supporting. For instance, kernel 2 which represents "VISA", the "GPO" is coded according to the EMV specifications and as shown in Table 3-1. Here, the POS sends the transaction related information such as the amount, other amount, currency, date and time. All these data are dynamic which mean they could be different in each transaction. However, in case of kernel 3 "MasterCard", the coded of this command is different as the transaction related data are not being sent during this command.

Once the card receives the "GPO" APDU command from the POS, it responds with several information which could be vary depends on which kernel the EMV card is supported. Mainly, there are four data should be included in the respond which are the Application File Locator (AFL), Application Interchange Profile (AIP), Processing Data Option List (PDOL) and Application Transaction Counter (ATC).

The AFL shows the location of all the data required by the POS in order to process the EMV contactless card transactions. The AFL's tag is 94 and it should include the file name and how many records in each file. All files and records should read by the POS during the next APDU command. While the AIP is a two-byte value and it indicates the EMV contactless card functionality such as which card authentication method is supported, CVM supported and whether online or offline or both are supported.

On the other hand, the PDOL represents all the needed data to process the transaction from the POS such as transaction amount with tag 81, transaction date with tag 9A, transaction currency with tag 5F 2A and transaction time with tag 9F 21. Moreover, the ATC represents a counter which is increased each time the EMV card process a transaction and it is used to generate the transaction session key in the last APDU command and response.

3.2.4. Read Record

After agreeing the required data to process during the transaction in the previous command, the POS starts to read all the EMV card details required to process the transaction by reading all the files and records are indicated in the AFL in the previous APDU response.

The AFL consists of four bytes where the first byte represents the Short File Identifier (SFI) which represents the record name. The second byte represents the first record under the SFI while the third byte represents the last record under the SFI. Moreover, the fourth byte represents the record that involved in the offline data authentication if applicable.

The "Read Record" APDU command parameters are shown in Table 3-1, where the P1 represents the record number and P2 represents the record name according to the AFL that provided by the EMV card in the "GPO" APDU response. The POS sends several "Read Record" commands instead of only one command due to the limitation of the APDU protocol which is limited to the smart card to send a maximum of 255 bytes according to the ISO 14443-4 [5]. Therefore, the POS needs to send several record commands to read all the required data that are needed to complete the transaction. The records include All the EMV card SI such as the PAN, card expiry date, IB public key certificate, card public key certificate, the singed version of SI and many more data depend on which kernel the EMV card is supported.

3.2.5. Generate Application Cryptogram (AC)

After reading the card's information by the POS using the "Read Record" APDU command, the POS sends the last required APDU command which is the "Generate AC" APDU command. In this command, the POS sends all the PDOL information that requested by the EMV contactless card in the "GPO" APDU response sideways the kind of AC that the POS wishes to process. This PDOL information could include transaction amount, transaction currency, transaction time and transaction date, etc. The AC in the EMV specifications has four types as explained in Section 2.1.2.3 namely TC, AAC, ARQC and APQC. Section 2.1.2.3 and Table 2-2 explained what each one of cryptogram is meaning. The AC type determines by the P1 in the

"Generate AC" APDU command as shown in Table 3-1, where tag number 40 represents the TC, 80 represents the ARQC and 00 represents the AAC.

Table 3-1: EMV APDU Commands Parameters

Command	CLA	INS	P1	P2	LC	Data	LE
"PPSE"	00	A4	04	00	0E	32 50 41 59 2E 53 59 53 2E 44 44 46 30 31	00
"Select AID"	00	A4	04	00	00	AID	00
"GPO"	80	A8	00	00	Var,	Var,	00
"Read Record"	00	B2	Var.	Var.	00	None	00
"Generate AC"	80	AE	Var.	00	Var.	Var.	00

Most of the APDU commands' parameters are static values as these parameters follow the EMV payment protocol specifications. However, few of these parameters are dynamic values that depend on the transaction type and the card's functionality as shown in Table 3-1.

The "GPO" APDU command could have a command data or not depending on which kernel the card follow. Therefore, the LC byte in this command does also vary for the same reason. While in case of the "Read Record" APDU command, both of the P1 and P2 are dynamic values depend on the record's name and the record number according to the AFL where some EMV cards could have two records and other card could have three or more records (depends on both the IB and the kernel).

Furthermore, in case of the "Generate AC" APDU command, there are three dynamic values in the APDU command parameters, where the P1 represents the AC type, LC represents the length of the PDOL that varies according to the transactions types and the data represents the AC which is unique for each transaction.

The first three APDU commands and responses between the POS and the EMV contactless card considered like a handshake between them. The main aim behind this handshake is to agree on all the next steps in order to process the transaction. Also, these three APDU commands and responses are static values in each transaction which mean both of the commands and responses are the same in each transaction. This helped the authors of [57] to reduce the time of relay attacks by cashing all the static commands and responses in such attack.

Now, after understanding the APDU protocol and most of the APDU commands and responses involved in the EMV contactless card transactions, we needed to test our knowledge by making several trails and tests on genuine EMV contactless cards in order to develop and build our own Java framework to duplicate genuine EMV contactless cards.

3.3. The Required Hardware & Software to Build the Java Contactless Card Framework

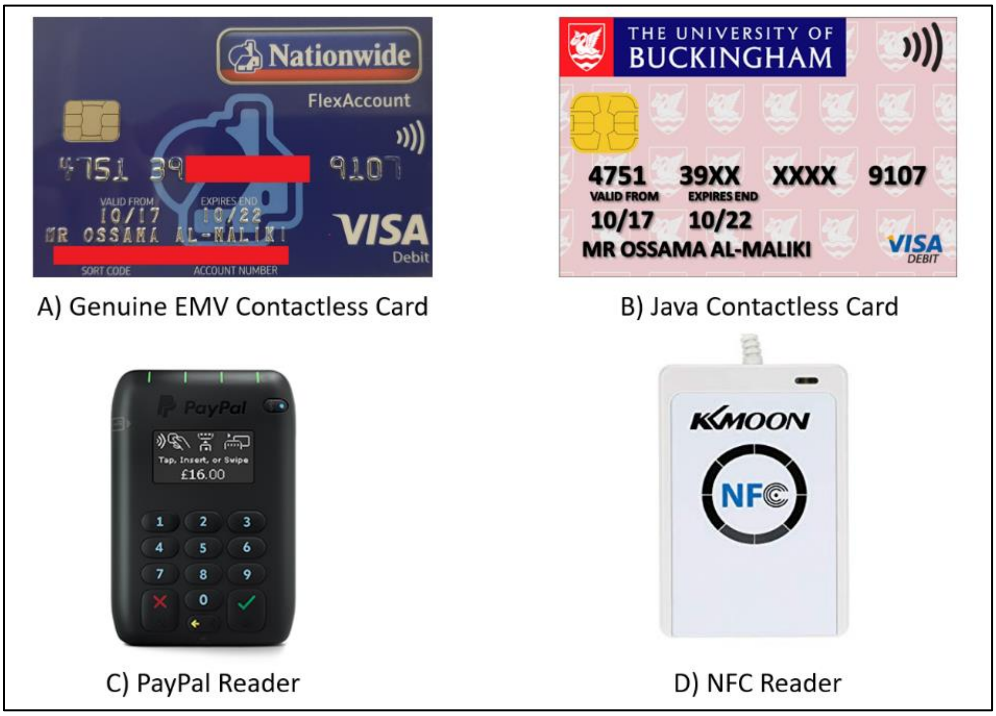
This section describes all the hardware and software required in order to achieve our trials and implementations to understand the EMV contactless card specifications. It also shows the required hardware and software to develop and build the Java contactless card framework.

3.3.1. The Required Hardware

We used several hardware devices in our trials and implementations as shown in Figure 3-2.

Genuine EMV contactless Cards

We used various genuine EMV contactless cards from different IBs in the UK such as Barclays, NatWest, Halifax, Nationwide and Santander banks as shown in Figure 3-3. The reason behind all of these cards is to test each one of them and figure out how each reacts to each one of the five main APDU commands. Also, another reason is to check the floor limit of the contactless card transactions. As the EMV contactless card transactions do not require any CVMs, therefore, these kinds of transactions are typically limited to a set maximum amount per transaction that known as the floor limit where this value is varying between each IB. All the testing is shown in the trail section.



A) Genuine EMV Contactless Card

B) Java Contactless Card

C) PayPal Reader

D) NFC Reader

Figure 3-2: Required Hardware to build Java Contactless Card Frameworks

Java Contactless Card

We used The JC30M48CR Java Card to simulate and duplicate a genuine EMV contactless card in our implementations [78]. This JC30M48CR Java contactless card comes with 48 EEPROM and 1.66KB as a RAM. It does have a dual interface with both contact and contactless and it supports all of these encryption algorithms MD5, SHA1/SHA256/SHA512, DES, AES, RSA. In addition, this kind of card support communication protocol type A for contactless and both of T=1 and T=0 for contact interface.

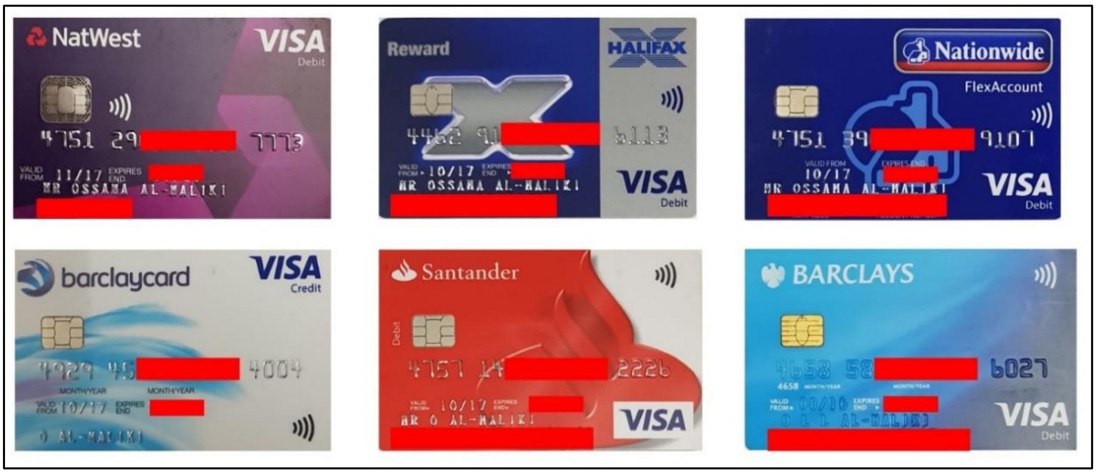


Figure 3-3: Six Genuine EMV Contactless Cards

Java Contactless Card

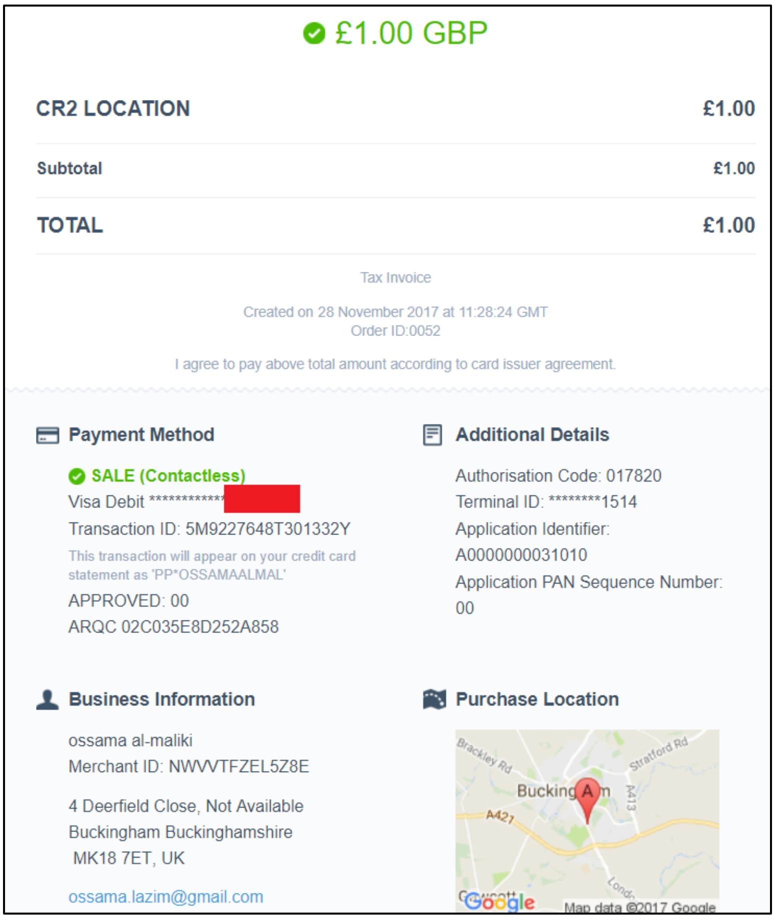
We used The JC30M48CR Java Card to simulate and duplicate a genuine EMV contactless card in our implementations [78]. This JC30M48CR Java contactless card comes with 48 EEPROM and 1.66KB as a RAM. It does have a dual interface with both contact and contactless and it supports all of these encryption algorithms MD5, SHA1/SHA256/SHA512, DES, AES, RSA. In addition, this kind of card support communication protocol type A for contactless and both of T=1 and T=0 for contact interface.

PayPal Here Card Reader

For more understanding of the EMV contactless card specifications, we needed to make real transactions on our EMV contactless cards. Therefore, we needed a genuine POS to process these real contactless transactions. We had two options for the genuine POS, the first option is the high street bank POS and the second option is the PayPal Here card reader. In order to apply for a high street bank POS, we were asked by the high street banks to present our business plan in order to check it. Then, credit checks must be performed on the account. Then, if everything goes right, the high street bank rents us the POS with monthly rent, and we need to guarantee minimum spending each month. All these points made a high street bank’s POS is a costly and complicated process for us. Therefore, we preferred to go with the second option which is the PayPal reader POS. All what we needed to have the PayPal Here card reader is to create a PayPal account with a personal bank account. It costs just £45 and no needs for a business plan or monthly rental fees as the PayPal reader transaction fees depend on the transaction amount with 0.03% fees [79]. Figure 3-4 shows a receipt of a contactless card transaction done by the PayPal POS.

ACR122U NFC Reader

For our implementations and trials, we used the ACR122U NFC reader to represent the POS in the implementations and trials to develop an applet to represent an EMV contactless card. This NFC reader is based on the 13.56 MHz RFID technology and supports ISO 14443 parts 1-4 for the contactless communications [80]. The ACR122U NFC reader sends and receives all the required APDU commands and responses between the POS and the genuine EMV contactless cards or our Java contactless cards.



✓ £1.00 GBP

CR2 LOCATION	£1.00
Subtotal	£1.00
TOTAL	£1.00

Tax Invoice

Created on 28 November 2017 at 11:28:24 GMT
Order ID:0052

I agree to pay above total amount according to card issuer agreement.

Payment Method

✓ SALE (Contactless)
Visa Debit ************[redacted]
Transaction ID: 5M9227648T301332Y
This transaction will appear on your credit card statement as 'PP*OSSAMAALMAL'
APPROVED: 00
ARQC 02C035E8D252A858

Additional Details

Authorisation Code: 017820
Terminal ID: ********1514
Application Identifier: A0000000031010
Application PAN Sequence Number: 00

Business Information

ossama al-maliki
Merchant ID: NWVVTFZEL5Z8E

4 Deerfield Close, Not Available
Buckingham Buckinghamshire
MK18 7ET, UK

ossama.lazim@gmail.com

Purchase Location

Figure 3-4: Transaction Receipt Done by PayPal Here

3.3.2. The Required Software

We used two software in our trials and implementations as follow:

Java Card Integrated Development Environment (JCIDE)

The Java Card Integrated Development Environment (JCIDE) is a software for developing applets by the use of Java card programming language [81]. We used the JCIDE to develop and build our applet to represent an EMV contactless card. This software also used to update the applet in order to cope with all of our contributions. As shown in Chapters 45, 68. This software supports several libraries that make developing an applet easier. For instance, APDU, security and ISO7816 libraries.

PyApdu Tool

The other software is called PyApduTool was used to connect both of our NFC readers and either the EMV genuine contactless cards or our Java contactless cards [78]. This software was also used to download and install our Java applet on the Java contactless card. Moreover, the PyApdu Tool used to send all the APDU commands and receive back all the APDU responses from either a genuine EMV contactless card or our Java contactless card.

3.4. EMV Contactless Card Transactions Trails and Testing

To understand more about how the EMV contactless cards work during the transactions, we used the above hardware to achieve several tests and trails as described in the next subsections.

3.4.1. Trails to Understand the APDU Responses

To test how each genuine EMV contactless card reacts to the APUD commands, we send the same APDU commands to three of our genuine EMV contactless cards namely Barclaycard credit, Barclays debit and Nationwide debit card which are shown in Figure 3-3.

As shown in Table 3-2, each one of the three EMV contactless cards reacted differently to the three APDU commands (“PPSE”, “Select AID” and “GPO”) regardless of that the three cards are based on kernel 2 “Visa”. The reason behind that is because each one of the tested cards is issued by different IB. Therefore, their APDU responses are differed according to the risk parameters and the SI of each one of them.

Moreover, for the rest two APDU commands (“Read Record” and “Generate AC”), all the three cards respond differently to each of these two commands for very clear reason. One of the reasons is the different keys that each card has. Therefore, the IB public key certificate, SDA, DDA, CDA and the card public key certificate are varying between all the cards. Also, the static data of these cards are different for example the PAN and the expiry date are different. Therefore, the responses of the “Read Record” and “Generate AC” differ among all the tested cards.

The complete analysis of traces of both APDU commands and responses which belong to the Nationwide EMV contactless card is shown in Appendix A.

Table 3-2:Testing Three EMV Contactless Cards APDU Responses

Command Name	Card Name	APDU Command	APDU Response
PPSE	Barclaycard (Credit)	00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 00	6F 47 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 35 BF 0C 32 61 30 4F 07 A0 00 00 00 03 10 10 87 01 01 50 10 42 41 52 43 4C 41 59 43 41 52 44 20 56 49 53 41 9F 0A 08 00 01 05 02 00 00 00 00 BF 63 04 DF 20 01 80 90 00
	Barclays (Debit)		6F 36 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 24 BF 0C 21 61 1F 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 02 BF 63 04 DF 20 01 80 90 00
	Nationwide (Debit)		6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90 00
Select AID	Barclaycard (Credit)	00 A4 04 00 07 A0 00 00 00 03 10 10 00	6F 56 84 07 A0 00 00 00 03 10 10 A5 4B 50 10 42 41 52 43 4C 41 59 43 41 52 44 20 56 49 53 41 87 01 01 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 5F 2D 02 65 6E BF 0C 13 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 02 00 00 00 00 90 00
	Barclays (Debit)		6F 42 84 07 A0 00 00 00 03 10 10 A5 37 50 0A 56 69 73 61 20 44 65 62 69 74 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 5F 2D 02 65 6E BF 0C 08 9F 5A 05 31 08 26 08 26 90 00
	Nationwide (Debit)		6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 90 00
GPO	Barclaycard (Credit)	80 A8 00 00 23 83 21 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	77 81 D1 82 02 20 00 94 04 10 02 05 00 57 13 49 29 45 76 63 24 40 04 D2 01 12 01 00 93 60 00 00 00 1F 5F 34 01 00 9F 10 07 06 0D 0A 03 90 00 00 9F 26 08 E5 4B 95 F8 82 33 23 3B 9F 27 01 40 9F 36 02 00 C8 9F 4B 81 80 C3 CB 43 59 F5 93 20 25 AF 13 92 3E F5 D3 97 6E D1 1C 61 0E EE 46 D7 D6 41 07 32 05 AF 9A C7 2A B1 3E 04 D7 AE 3F E1 CE 7F F6 B2 B9 82 1B D2 2E DD D3 DF 98 29 DC DA D6 C9 BF 7B 1B 66 4B 4B 5A B3 62 28 7D 03 40 19 7D 70 84 4D 4D C0 08 45 48 43 47 AA 49 BC 73 3E 2E AD 1B 16 96 CD 5F 0D 5A 63 91 AE E8 FB AB 47 97 C8 7F 81 A2 6E A3 3E 2D 3D BA B3 4E EA 67 C4 15 5F 32 C3 13 EB AB C8 07 9F 6C 02 10 00 9F 6E 04 20 70 00 00 90 00
	Barclays (Debit)		77 47 82 02 20 00 57 13 46 58 58 55 79 61 60 27 D1 90 82 21 83 00 00 00 00 00 1F 5F 34 01 00 9F 10 07 06 0B 0A 03 A0 00 00 9F 26 08 D9 51 21 5F C4 F9 EB F3 9F 27 01 80 9F 36 02 04 C9 9F 6C 02 10 00 9F 6E 04 20 70 00 00 90 00
	Nationwide (Debit)		77 4D 82 02 20 00 94 04 10 02 05 00 57 13 47 51 39 05 86 29 91 07 D2 21 02 21 99 20 00 00 00 00 0F 5F 34 01 00 9F 10 07 06 01 0A 03 A0 00 00 9F 26 08 F4 E1 96 C3 0D 93 CD 81 9F 27 01 80 9F 36 02 00 5E 9F 6C 02 2E 00 9F 6E 04 20 70 00 00 90 00

3.4.2. Trails to Understand the Floor Limit

As described in Section 2.1.3.3, the contactless card floor limit is determined by the IB and it could either a number of contactless card transactions which are processed correspondingly or a maximum amount to spend in such transactions before requesting chip and PIN transactions by the EMV cards. We needed to understand how this parameter is work with our different genuine EMV contactless cards in order to develop and built out Java framework closer to the real life EMV contactless cards.

To do that, we needed to restart the NFC counter to zero again, we did that by making a chip & PIN transaction one time for each genuine EMV contactless card. The reason behind this is to test the actual contactless card floor limit where the NFC counter is zero. Then, we did £1 contactless card transactions by the use of PayPal Here POS until the EMV contactless card reject the transaction and request a chip & PIN transaction instead. The tested cards react to this differently where one cards approved 12 contactless card transactions in the row while another card approved 15 transactions until rejecting and requesting chip & PIN.

3.5. Developing and Building the Java Framework

The main aim behind developing and building the Java framework is to simulate the genuine EMV contactless cards. Therefore, we followed the EMV payment protocol specifications and the EMV contactless card specifications practically as possible. Figure 3-5 shows the development process and all the required specifications in order to develop the framework. The next subsections show the development steps.

3.5.1. Implementation of the Personalisation Phase

In the original EMV personalisation phase, the IB or a third party who has contracted with the IB is in charge of this phase where all the data preparation is done and the card profile is created [82]. The risk parameters are set in this phase where these parameters determine how the EMV card preform and functions during the transactions. For example, whether the EMV card supports online or offline transactions or both. The risk parameters determine both of the CVMs and card authentication methods that the EMV card could support. The keys management is done at the personalisation phase where all the required keys and certificates are uploaded in this phase [28]. Moreover, all the static data are stored into the EMV card such as assigning a unique PAN for each EMV card with the expiry date. When all these data are uploaded and all the different functionalities are declared to the EMV card, the card is ready to be hand over to its owner (cardholder).

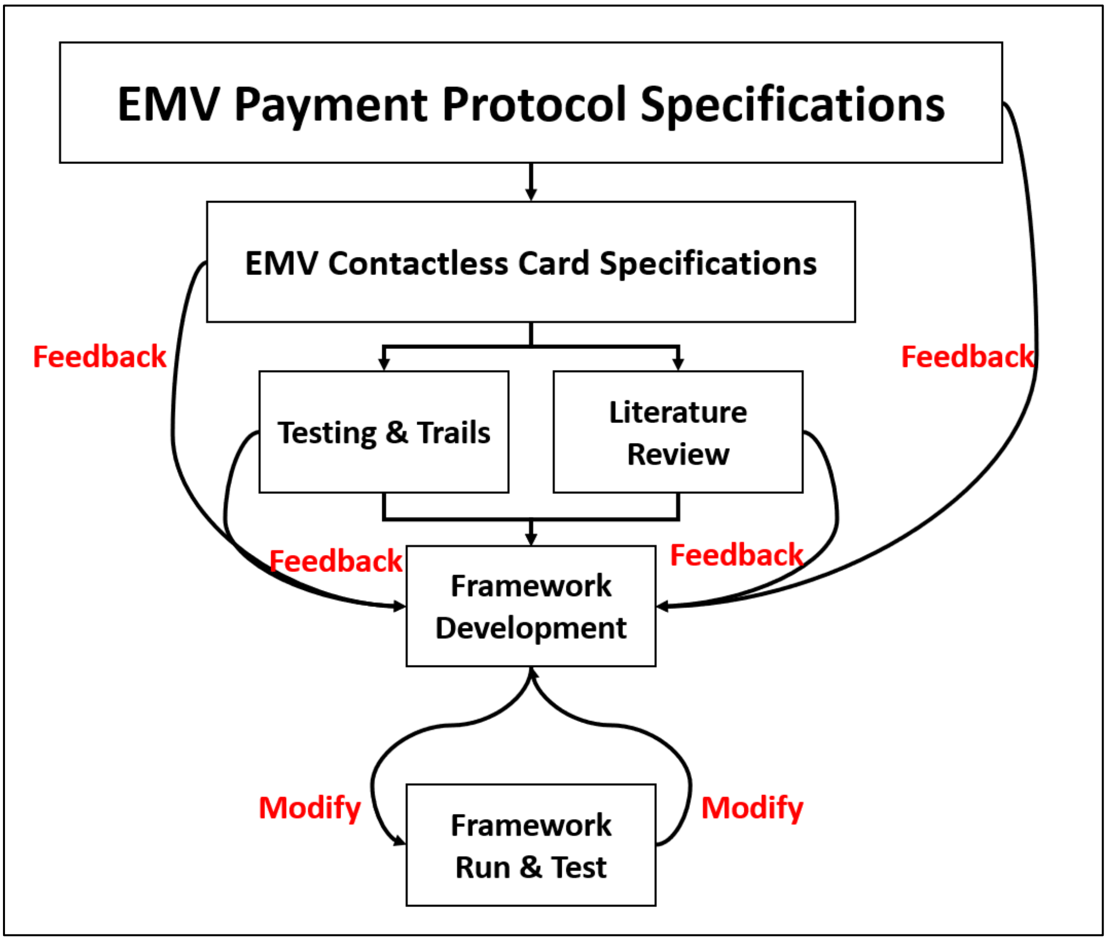


Figure 3-5:Java Framework Required Specifications

We developed our Java framework according to the EMV payment protocol specifications in order to duplicate a genuine EMV contactless card. We started by setting the AIP where we decided that our Java card supports the SDA authentication method and the CVMs are not required in our Java card as we focus on implementing the EMV contactless card only in this framework. Therefore, we ended up setting the AIP into two bytes value 20 00 (see Table 37, page 160 [28]). While for the AFL, we decided to make our Java contactless card consists of three records where we set the record name into B2 and the three files starting from record number 02 until 04.

The PDOL as explained in Section 3.2.3 is determined at this stage (personalisation) where the PDOL is represented by tag number 9F 38. We set the PDOL in our Java framework into 16 bytes to represent all the data that are required by the EMV card (our Java framework) in order to process the contactless card transaction. These 16 bytes represent the transaction amount tag and the length, other transaction amount tag and the length, transaction date tag and length, transaction time tag and length, transaction currency tag and length and finally the UN tag and length.

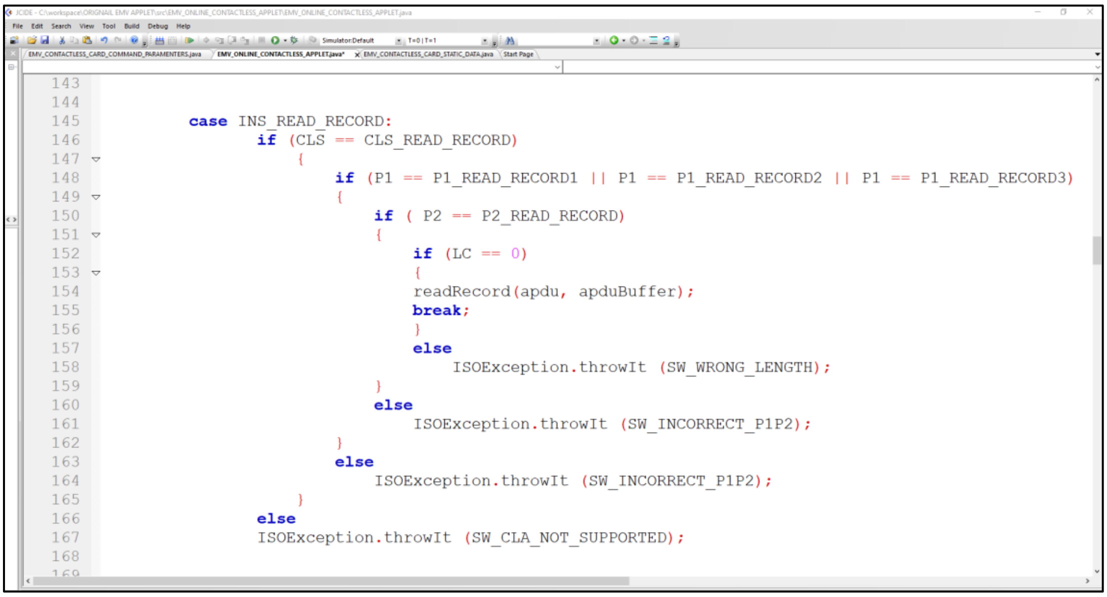
Furthermore, the three records that we chose to have in our Java contactless framework are uploaded with different kinds of SI. The first record is uploaded by all the EMV IS such as PAN and expiry date. The second record is uploaded with IB public key certificate while the third record is uploaded with the singed version of all the SI. Also, at this phase, the AID for our Java framework is assigned.

We intended to make our Java framework as flexible as possible in order to modify it to serve any kind of testing. That means giving the developers the ability to change the framework's code according to the developers' requirements as they might want to design their framework with different kinds of risk parameters, PDOL, SI and for sure different kinds of keys and IB public key certificates.

3.5.2. Implementation of the APDU Communication protocol

In the second part of the Java framework, we made sure that the developed Java framework is designed to respond to all the five main APDU commands that are detailed in Section 3.2 in order to process the EMV contactless card transactions. All these APDU parameters are declared in our framework according to the EMV specifications.

For instance, the APDU command parameters for the "GPO" APDU command are declared in our framework according to the EMV book 3, Section 6.5.8.2 on page 59 [28]. We declared four APDU commands parameters for "PPSE", "GPO", "Read Record" and "Generate AC" APDU commands. While the "Select AID" APDU command is already been installed by the JCIDE as a default. Therefore, we did not declare the "Select AID" command parameters.


JCIDE - C:\workspace\ORIGNAIL EMV APPLET\src\EMV_ONLINE_CONTACTLESS_APPLET\EMV_ONLINE_CONTACTLESS_APPLET.java
File Edit Search View Tool Build Debug Help
Simulator:Default
T=0 | T=1
EMV_CONTACTLESS_CARD_COMMAND_PARAMENTERS.java
EMV_ONLINE_CONTACTLESS_APPLET.java*
EMV_CONTACTLESS_CARD_STATIC_DATA.java
Start Page

```
case INS_READ_RECORD:
       if (CLS == CLS_READ_RECORD)
           {
               if (P1 == P1_READ_RECORD1 || P1 == P1_READ_RECORD2 || P1 == P1_READ_RECORD3)
               {
                   if ( P2 == P2_READ_RECORD)
                   {
                       if (LC == 0)
                       {
                       readRecord(apdu, apduBuffer);
                       break;
                       }
                       else
                           ISOException.throwIt (SW_WRONG_LENGTH);
                   }
                   else
                       ISOException.throwIt (SW_INCORRECT_P1P2);
               }
               else
                   ISOException.throwIt (SW_INCORRECT_P1P2);
           }
       else
       ISOException.throwIt (SW_CLA_NOT_SUPPORTED);
```

Figure 3-6: Checking the APDU Command parameters in the Java Framework

In our Java contactless card framework, we made sure that our code is following the EMV specifications precisely to duplicate the genuine EMV contactless cards. Therefore, one of the essential points that we were focusing on is to make sure that the framework is responding to only the correct APDU commands. However, in case of a wrong or missing APDU command, the framework responds with the suitable SWs according to the EMV specifications. For example, as shown in Figure 3-6, the framework checks each part of the APDU command is a genuine one before executing any command. In the same figure, all of the ISN CLS, P1, P2, and LC bytes are checked and if all these bytes are matched with the original EMV command according to the specifications, then our Java framework excuses the APDU command and sends a correct response. Else the Java framework responds with specific SWs according to which parameter byte is wrong as shown in Table 3-3 [34][28].

Table 3-3 Status Word 1 & 2 Meaning

SW1 & SW2	Meaning
90 00	Successfully Executed (OK)
6E 00	CLA Value not supported
6D 00	INS value not supported
6A 89	Incorrect Parameters (P1: P2)

6A 82 *	*File Not Found (in case of Read Record command)
6A 89 6A 82 *	Incorrect Parameters (P1: P2) *File Not Found (in case of Read Record command)
67 00	Wrong Length

3.5.3. Implementation of the Transaction & Authorisation Phases

After all the checking of each APDU command's parameters as discussed in the previous section are correctly matched with the original EMV APDU commands' parameters. Then, the Java framework should respond to each command with specific actions as explained in Section3.2.

The implementation of the authorisation phase is harder than the previous phases due to the dynamicity of this phase. As in this phase, the "Generate AC" APDU command is handled according to the EMV specifications [27]. Therefore, a session key is generated from the master key that shared between the IB and the EMV cards. This session key is used to create the AC whether it is ARQC, TC or AAC as explained in 2.1.2.3. As the data are used to generate the AC are the Application Transaction Counter (ATC), the TD from the POS and the TD from the card. However, before generating the AC, our Java framework several checks to decide whether to generate the requested AC or a lower priority AC. One of these checks is the contactless floor limit transaction is exceeded in both of the online and offline transactions. In the Java framework, we set the contactless floor limit transaction of the online transaction to 10 times while the offline to 5 times as these are basically decided by the IB at the personalisation phase.

3.6. Methodology of Validation for the Java Framework

The Java contactless card applet supported the main five APDU commands and their responses as discussed in Section 3.2. The applet was downloaded and installed into the Java contactless card by using the PyApdu Tool. Next, we tested our applet by using the NFC ACR 122U. The NFC reader represents the POS in the reality where a genuine POS sends EMV commands to the EMV card. The NFC reader sends the identical EMV commands to our Java contactless card and the applet responses with the proper EMV responses.

Both “PPSE” and “Select AID” APDU commands are showing how our applet response with the AID and how the POS (NFC reader) will select it. While at the third command “GPO”, the applet responses to the correct APDU command with AFL, AIP and PDOL.

Moreover, in the next APDU command, which is the “Read Record” command, our Java applet responses to each of the three “Read Record” commands (according to the AFL) with different data as explained in more details in the appendix. As its response to the first “Read Record” by the card’s static data such as the PAN and card expiry date which we had to make sure they both follow our genuine EMV contactless card that mentioned in the hardware section. While the second record contents of the IB public key certificate where the IB public key is signed by the CA private key at the personalisation phase. Furthermore, the third record in our applet contents of the signed version of the EMV card’s details where the IB signed these details by its private key.

Finally, at the fifth APDU command the “Generate AC”, our applet signs both the static data and the PDOL which was sent by the POS (NFC reader in our implementation) by a session key known by both IB and the card itself.

We can conclude that our Java Framework can successfully mimic works as a genuine EMV contactless cards by responding to the APDU commands as the same way a genuine EMV contactless card does. Also, the first three APDU responses are being copied form one of our genuine EMV contactless cards (Nationwide debit). Therefore, all the responses are matched with the Nationwide card’s responses. Furthermore, the last response which belongs to the “AC” APDU command is vary due to the TD are being dynamic and the different keys between our Java framework and the keys in the Nationwide card as shown in Table 3-4. As a result, the Java framework works perfectly in order to duplicate any genuine EMV contactless card.

Table 3-4 Application Cryptogram APDU Command and Response for both the Java Framework and Genuine EMV contactless Card

APDU Command Name	**APDU command**	**Genuine EMV Contactless Card**	**Java Contactless Card**
Generate Application Cryptogram (AC)	80 AE 40 00 1D 81 00 00 00 05 9F 04 00 00 00 00 9A 20 10 17 9F 21 04 00 00 5F 2A 82 60 9F 37 12 34 56 00	98 65 41 D2 7F 79 95 6F B5 55 34 86 74 F4 B1 ED 6F CA F4 78 BB 72 AD 09 3C 2F 8B 20 26 79 86 11 4C 19 85 CD 0D F7 3F C5 5A CC 44 40 6B 33 68 9D 79 11 A9 9C 4F 18 E7 75 11 A3 20 48 BF DF 6B 71 75 0B 90 00	98 65 41 59 BB D8 42 7F 4C D4 90 42 36 F9 DB 87 B7 76 FB 4E 77 9C 86 CA 91 74 7D CA 35 51 49 B5 37 33 6C 38 3C 0E DC 3F FE DB 2A 39 48 07 B3 4C C5 69 34 87 3B CD AC 46 7F 3E 8E 8C A5 48 5C 22 9E 0F 90 00

3.7. Summary of the Chapter

Developing and building a Java framework for the EMV contactless cards helped us to understand how such cards are working at bits and bytes level. The aim of the developed Java framework is to duplicate the genuine EMV contactless card by following the EMV payment protocol specifications. Therefore, we started our development process from the first phase of the EMV cards lifespan which is the personification phase, followed by all of the communication, transaction and authorisation phases.

We made the Java framework easy to update in order to implement different risk parameters and functionalities. For example, we updated our Java framework in Chapter 4 to implement the other two kinds of card authentication methods the DDA and CDA. While in Chapter 5, the Java framework was updated to implement the tokenisation proposal. The mutual authentication proposal in Chapter 6 also requires updating the Java framework. In Chapter 8, the Java framework was also modified in order to implement the proposal presented in this chapter.

4. Chapter 4: Security Analysis of the EMV Card Authentication Methods

This chapter is mainly concerned with the security analysis and evaluation of the EMV card authentication methods of the contactless cards as well as investigating their vulnerabilities to sniffing and replay attacks. It provides a detailed evaluation of the three EMV card authentication methods (SDA, DDA and CDA). To demonstrate that, we updated the Java framework described in Chapter 3 to simulate all the stages of The EMV contactless card transactions under the three authentication methods. The chapter also simulates the two attacks using Java contactless cards including a counterfeit card to show the feasibility of the replay attack.

Our security analysis reveals that the authentication methods have a different level of robustness against the attacks. The chapter concludes that the SDA has the lowest level of security with vulnerabilities to both attacks. It also demonstrates that although DDA and CDA are perceived to be secure, they are both vulnerable to sniffing attacks that can be easily launched to steal credit/debit card details. We argue that the card authentication methods share a fundamental flaw related to the trust model between the POS and the EMV contactless card. While POS enforces strict rules to ensure the authenticity of the card, EMV contactless cards release sensitive information to anyone with NFC enabled readers/smartphones without any checks whatsoever.

The chapter starts with Section 4.1 by breaking down the EMV card authentication methods into five different stages namely keys distribution, IB public key certificate, card personalisation, transaction, and authentication. Then, it presents the implementation and evaluation of the three types of the EMV card authentication methods and how each method reacts to both of sniffing and relay attacks in Section **Error! Reference source not found.**. Next, Section 4.3 details the discussion and recommendation. Finally, Section 4.4 concludes the chapter findings.

4.1. The Authentication Methods of EMV Cards

Each EMV card must support at least one of the three authentication methods as discussed in 2.1.2.1. This is typically decided by the IB who decides the capability of the EMV card. Then, the supported card authentication method is declared by the AIP sends by the EMV card to the POS during the transaction [28]. After requesting the AIP using the “GPO” APDU command, the POS chooses the authentication method with highest priority where CDA is the highest followed by DDA and then SDA [27]. The rationale behind these different priorities is explained in the next subsections.

Each of the three EMV authentication methods consists of five stages that are the keys distribution, IB public key certificate, card personalisation, transaction, and authentication. The first three stages are done before issuing the EMV card to its cardholder as shown in Figure 4-1 while the last two stages are done during the transaction as shown in both Figure 4-2 and Figure 4-3 .

4.1.1. EMV Card Authentication at the Keys Distribution Stage

In this stage, all the required keys are shared between all the involved parties as shown in the next three steps. This stage applies to all three types of EMV card authentication methods namely SDA, DDA, and CDA as shown in Figure 4-1.

1- The CA sends its public key (CA_{pk}) in a secure environment to the IB.

$$\mathbf{CA \rightarrow IB: CA_{pk}}$$

2- The CA sends its CA_{pk} in a secure environment to the AB.

$$\mathbf{CA \rightarrow AB: CA_{pk}}$$

3- The AB upload the CA_{pk} into the POS which belongs to AB.

$$\mathbf{AB \rightarrow POS: CA_{pk}}$$

4.1.2. EMV Card Authentication at the IB Public Key Certificate Stage

In the second stage of the EMV card authentication, the IB certificate is generated as shown in the next two steps. This stage applies to all three types of EMV card authentication methods as shown in Figure 4-1.

4- The IB sends its IB_{pk} to the CA.

$$\mathbf{IB \rightarrow CA: IB_{pk}}$$

5- The CA uses its Private key (CA_{sk}) to sign the IB_{pk} and then send it back to the IB.

CA→ IB: CA_{sk} {IB_{pk}}

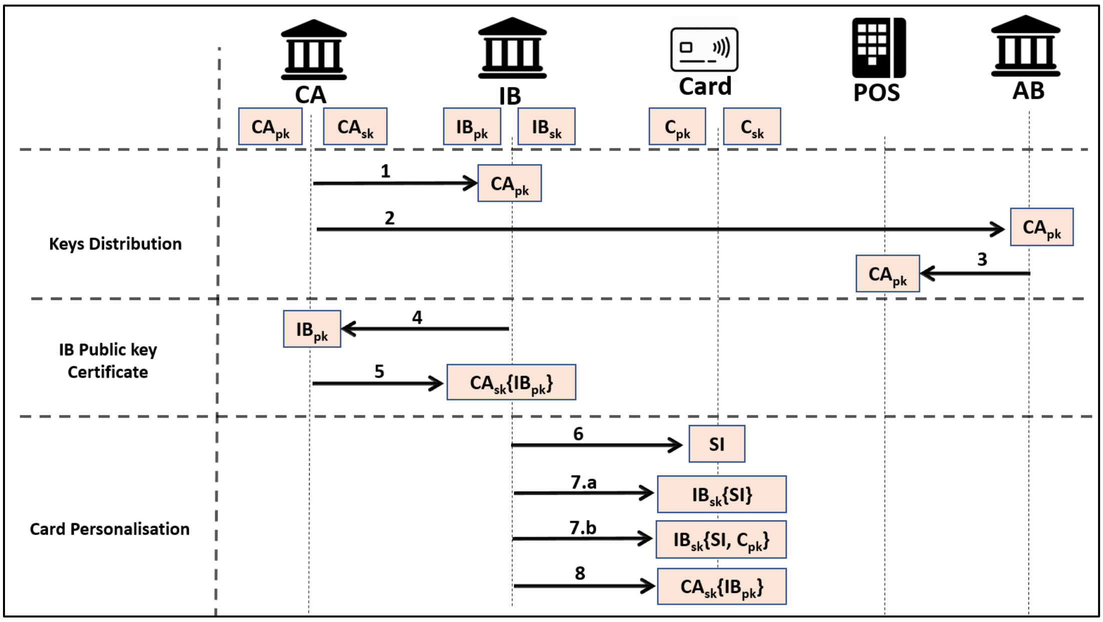


Figure 4-1: Keys Distribution, IB public key certificate and Personalisation processing in the EMV Card Authentication Methods

4.1.3. EMV Card Authentication at the Card Personalisation Stage

In the third stage of the EMV authentication methods, the EMV card is uploaded with all its SI and the IB public key certificate. This stage applies to all three types of EMV card authentication methods as shown in Figure 4-1.

6- The IB stores the SI for each EMV card. These SI could include the cardholder name, the PAN and the expiry date of the card.

IB→ Card: SI

7- The IB signs the SI by its IB_{sk} and stores into the card in case of the SDA as shown in Figure 4-1 step (7.a). While in the case of the DDA and CDA, the IB signs both SI and the card public key (C_{pk}) by the IB_{sk} as shown in the same figure in step (7.b). Step 7.a represents the process in case of the SDA while step 7.b represents the process in case of both the DDA and CDA.

(7.a) IB→ Card: IB_{sk} {SI'}

(7.b) IB→ Card: IB_{sk} {SI', C_{pk}}

8- The IB stores the IB_{pk} certificate which was generated at step 5 into the card.

IB→ Card: CA_{sk} {IB_{pk}}

4.1.4. EMV Card Authentication at the Transaction Stage

In the fourth stage of the EMV authentication methods, both POS and EMV cards start to communicate with each other to process a transaction. Steps 9, 10 and 11 apply for all the three types of EMV card authentication methods while steps 12 and 13 apply just for both DDA and CDA. However, step 14 applies just in case of the CDA as shown in Figure 4-2.

9- The card sends its own SI in plaintext (that was stored by the IB at step 6) to the POS in response to "Read Record" APDU command.

Card→ POS: SI

10- The card sends the signed SI (that was generated and stored at step 7.a in case of the SDA) to the POS in response to the "Read Record" command as shown at the step 10.a while in case of both DDA and CDA, the card sends both of the signed SI and the C_{pk} (which were generated and stored at step 7.b in case of both DDA and CDA) to the POS in response to the "Read Record" command as shown at the step 10.b.

(10.a) Card→ POS: IB_{sk} {SI'}

(10.b) Card→ POS: IB_{sk} {SI', C_{pk}}

11- The card sends the IB_{pk} certificate (which was generated at step 5 and stored at step 8) to the POS in response to the "Read Record" command.

Card→ POS: CA_{sk} {IB_{pk}}

12- POS sends the Transaction Data (TD) to the card during the "Generate AC" command. This TD includes the transaction amount, the transaction date, transaction time, transaction currency and the UN.

POS→ Card: TD

13- In case of the DDA, the card signs the TD which was sent by the POS at step (12) by its Private key (C_{sk}) and sends it back to the POS during the "Generate AC" response as shown in the step 13.a while in case of the CDA, the card signs the TD together with the AC by its C_{sk} and send it back to the POS during the "Generate AC" response as shown in the step 13.b in Figure 4-2.

(13.a) Card→ POS: C_{sk} {TD'}

(13.a) Card→ POS: C_{sk} {TD', AC'}

14- The card sends its own AC in plaintext to the POS. This step is valid just in case of the CDA.

Card→ POS: AC

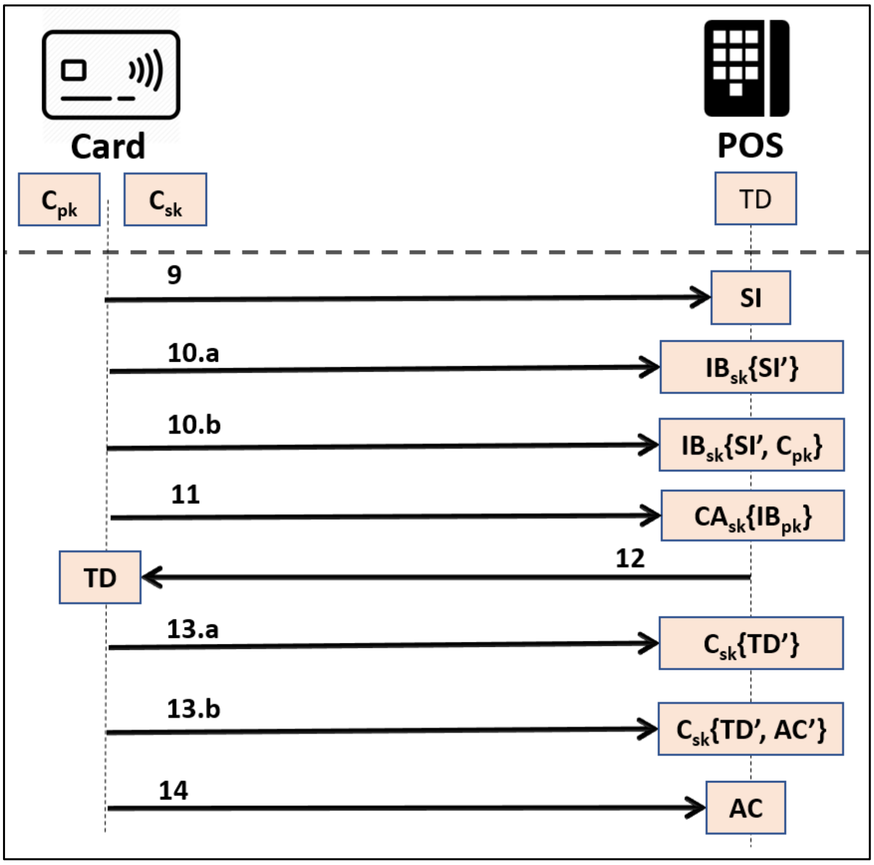


Figure 4-2: Transaction stage of the EMV Card Authentication Methods

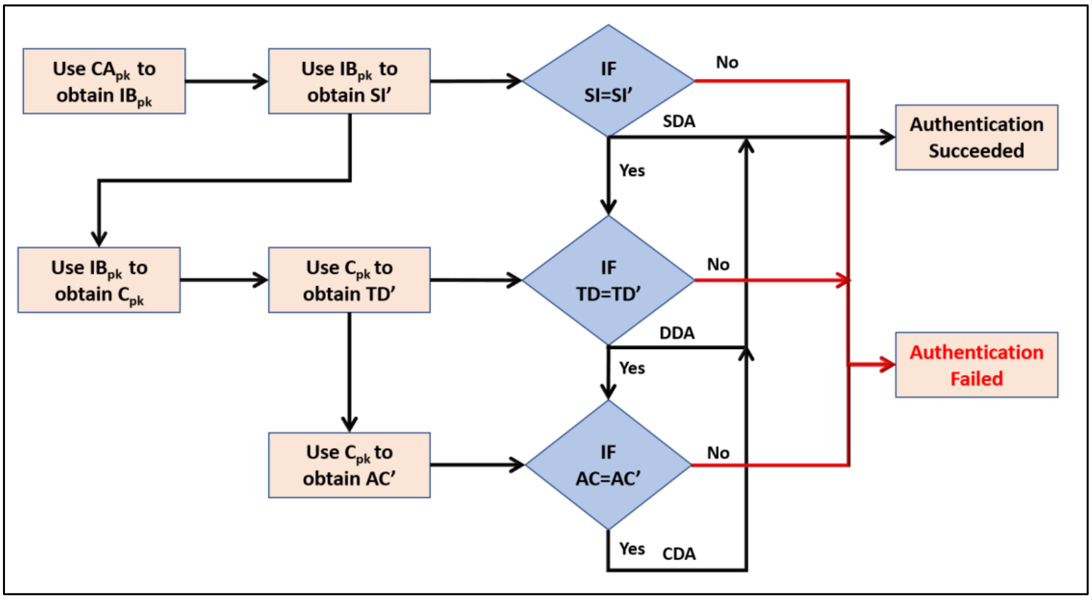
4.1.5. EMV Card Authentication at the Authentication Stage

In the fifth and last stage of the EMV authentication methods, the POS retrieves the IB_{pk} in case of SDA, DDA and CDA and the C_{pk} in case of both DDA and CDA. These retrieved keys help the POS to obtain all the required information and decide whether the EMV card is a genuine card or not. Steps 15 and 16 apply for all the three types of

EMV authentication methods. While steps 17, 18 and 19 apply for both DDA and CDA. Figure 4-3 shows the EMV card authentication methods at the authentication phase.

15- POS uses the CA_{pk} (which was uploaded by the AB at step 3) to obtain the IB_{pk} from the message number (11). This step applies to the three types of EMV card authentication methods.
16- In the case of the SDA, the POS uses the IB_{pk} (which was obtained at step 15) to verify the message number (10.a) in Figure 4-2 and obtain the SI' which belongs to the EMV card. Then, the POS compares both SI data and all the card SI which was sent to the POS in plaintext during the "Read Record" command at step 9 in the same figure. If both are matched. Then, the POS successfully authenticates the EMV card as a genuine one and it should set the SDA failed bit to 0 in the first byte of the Terminal Verification Results (TVR) [31]. Else, the authentication process is failed, and the POS should set the SDA failed bit to 1 in the first byte of the TVR. While in the case of both DDA and CDA, the POS uses the IB_{pk} (which was obtained at step 15) to decrypt message number (10.b) to obtain the SI' which belongs to the EMV card along with the C_{pk}.
17- In the case of the DDA, the POS uses the C_{pk} (which was obtained at step 16) to verify the message number (13.a) in Figure 4-2 and obtain the TD. While in the case of the CDA, the POS uses the C_{pk} to verify message number (13.b) and obtain both TD' and the AC'.
18- In the case of the DDA, the POS compares two components which are the SI and TD to make its decision whether the card is a genuine EMV card or not. Firstly, the POS matches between the signed version of the SI' (which was obtained at step 16) and all the card SI which was sent to the POS in plaintext during the "Read Record" command at step 9. Secondly, the POS compares between its own TD (which was sent to the card at step (12) and the TD' which was signed by the card' C_{sk} at step (13) and obtained at step (17). If both two comparisons are true. Then, the POS successfully authenticates the EMV card as a genuine one and it should set the DDA failed bit to 0 in the first byte of the TVR. Else, the authentication process is failed, and the POS should set the DDA failed bit to 1 in the first byte of the TVR.

19- In the case of the CDA, the POS compares three components which are the SI, TD and the AC to make its decision whether the card is a genuine EMV card or not. Firstly, the POS matches between the signed SI' (which was obtained at step 16) and all the card SI which was sent to the POS in plaintext during the "Read Record" command at step 9. Secondly, the POS compares between its own TD (which was sent to the card at step 12) and the TD' which was signed by the card' C_{sk} at step (13) and obtained at step (17). Thirdly, the POS matches between the signed AC' (which was obtained at step 17) and the AC which was sent by the card as a response of the "Generate AC" command at step (14). If all the three comparisons are true. Then, the POS successfully authenticates the EMV card as a genuine one and it should set the CDA failed bit to 0 in the first byte of the TVR. Else, the authentication process is failed, and the POS should set the CDA failed bit to 1 in the first byte of the TVR.



Use CA_{pk} to obtain IB_{pk}
Use IB_{pk} to obtain SI'
IF SI=SI'
No
SDA
Yes
Authentication Succeeded
Use IB_{pk} to obtain C_{pk}
Use C_{pk} to obtain TD'
IF TD=TD'
No
DDA
Yes
Authentication Failed
Use C_{pk} to obtain AC'
IF AC=AC'
No
Yes
CDA

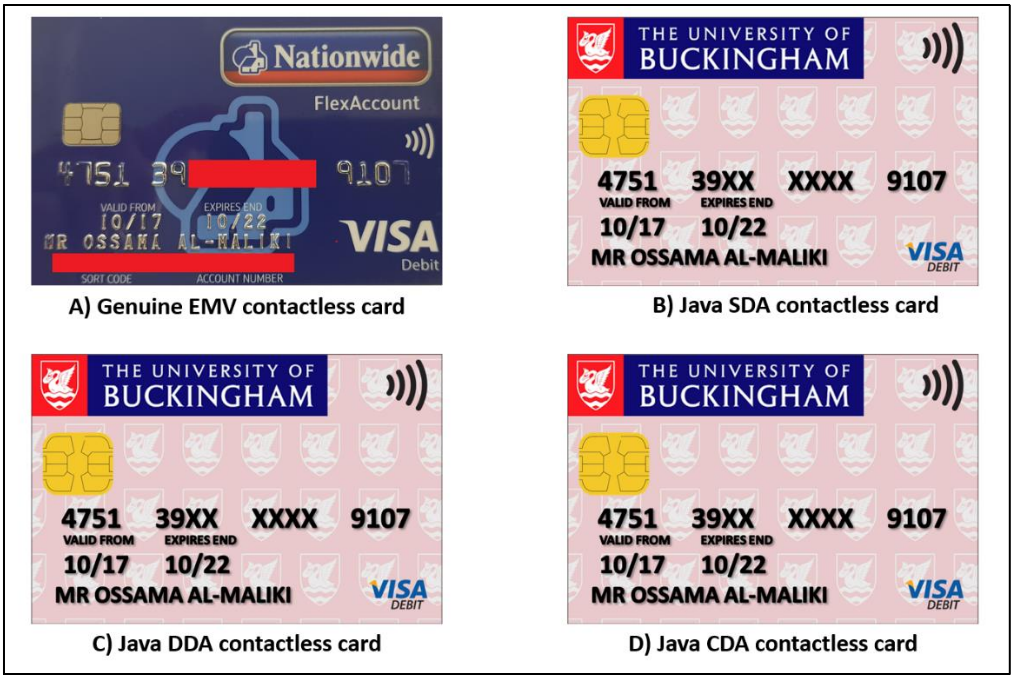
Figure 4-3: Authentication stage of the SDA, DDA, and CDA

4.2. Study the EMV Card Authentication Methods Against Both Sniffing and Re-play Attacks

This section details our implementations for the EMV contactless cards payment protocol and the three authentication methods. Our main aim behind the implementation is to implement the three EMV card authentication methods and how these methods react to both sniffing and replay attacks. We did our implementation by using both hardware and software as explained in Chapter 3.

4.2.1. EMV Contactless Card Applets

We developed three EMV applets using the JCIDE software to represent the three types of EMV authentication methods. The applets act as an EMV contactless card by responding to the POS commands according to the EMV specifications. However, all these three applets developed with the same keys (CA_{sk}, CA_{pk}, IB_{sk}, IB_{pk}, $C_{sk,}$ and C_{pk}) and the same card static information (cardholder name, PAN and expiry date) to serve the purpose of this chapter and show how the SDA, DDA, and CDA react to each of sniffing and replay attacks. Figure 4-4 shows our four testing contactless cards which represent genuine EMV contactless card (A), Java SDA Contactless card (B), Java DDA contactless card (C), and Java CDA contactless card (D). We built our Java applets to duplicate the genuine EMV contactless card (A) while all of Java cards (B, C and D) represent the SDA, DDA and CDA respectively.



A) Genuine EMV contactless card

B) Java SDA contactless card

C) Java DDA contactless card

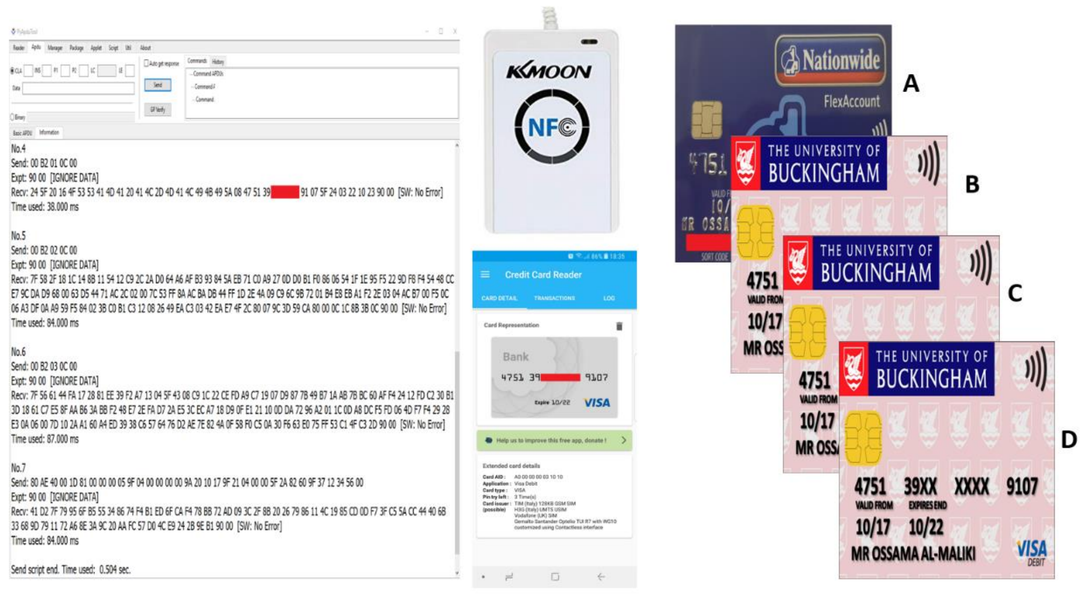
D) Java CDA contactless card

Figure 4-4: The Used Contactless Cards for Testing the EMV Card Authentication Methods

4.2.2. Sniffing Attack

The EMV contactless cards can be read by any POS or even by any NFC enabled smartphones due to the wireless connectivity of the cards, which means that fraudsters can easily obtain most of the card's sensitive information such as the PAN, expiry date, and even the cardholder name using on-the-shelf hardware and software. We successfully obtained all these data using the NFC ACR 122U reader as POS and the PyApduTool to send a set of commands to a genuine EMV contactless card and our Java contactless cards. Moreover, we used a free off-the-shelf Android application called "Credit Card Reader" to read most of the contactless card information even though the card still inside the cardholder wallet, which can be easily done without the cardholder's knowledge as shown in Figure 4-5.

All three EMV authentication methods (SDA, DDA, and CDA) fall to the sniffing attack. That is because the EMV card sends its own SI in plaintext to the POS to enable the POS to verify the signed version of the SI either IBsk{SI} (step 10.a Figure 4-2) or IBsk{SI', Cpk} (step 10.b in Figure 4-2) with the plaintext version of the SI (step 9 in Figure 4-2). The stolen data could be eventually used by the attacker to achieve CNP attacks such as online shopping as some of the online websites do not require the CVC.



```
No.4
Send: 00 B2 01 0C 00
Expt: 90 00  [IGNORE DATA]
Recv: 24 5F 20 16 4F 53 53 41 4D 41 20 41 4C 2D 4D 41 4C 49 4B 49 5A 08 47 51 39[redacted]91 07 5F 24 03 22 10 23 90 00  [SW: No Error]
Time used: 38.000 ms

No.5
Send: 00 B2 02 0C 00
Expt: 90 00  [IGNORE DATA]
Recv: 7F 58 2F 18 1C 14 8B 11 54 12 C9 2C 2A D0 64 A6 AF B3 93 84 5A EB 71 C0 A9 27 0D D0 B1 F0 86 06 54 1F 1E 95 F5 22 9D F8 F4 54 48 CC
E7 9C DA D9 68 00 63 D5 44 71 AC 2C 02 00 7C 53 FF 8A AC BA DB 44 FF 1D 2E 4A 09 C9 6C 9B 72 01 B4 E8 EB A1 F2 2E 03 04 AC B7 00 F5 0C
06 A3 DF 0A A9 59 F5 84 02 3B C0 B1 C3 12 08 26 49 EA C3 03 42 EA E7 4F 2C 80 07 9C 3D 59 CA 80 00 0C 1C 8B 3B 0C 90 00  [SW: No Error]
Time used: 84.000 ms

No.6
Send: 00 B2 03 0C 00
Expt: 90 00  [IGNORE DATA]
Recv: 7F 56 61 44 FA 17 28 81 EE 39 F2 A7 13 04 5F 43 08 C9 1C 22 CE FD A9 C7 19 07 D9 87 7B 49 B7 1A AB 7B BC 60 AF F4 24 12 FD C2 30 B1
3D 18 61 C7 E5 8F AA B6 3A BB F2 48 E7 2E FA D7 2A E5 3C EC A7 18 D9 0F E1 21 10 0D DA 72 96 A2 01 1C 0D A8 DC F5 FD 06 4D F7 F4 29 28
E3 0A 06 00 7D 10 2A A1 60 A4 ED 39 38 C6 57 64 76 D2 AE 7E 82 4A 0F 58 F0 C5 0A 30 F6 63 E0 75 FF 53 C1 4F C3 2D 90 00  [SW: No Error]
Time used: 87.000 ms

No.7
Send: 80 AE 40 00 1D 81 00 00 00 05 9F 04 00 00 00 00 9A 20 10 17 9F 21 04 00 00 5F 2A 82 60 9F 37 12 34 56 00
Expt: 90 00  [IGNORE DATA]
Recv: 41 D2 7F 79 95 6F B5 55 34 86 74 F4 B1 ED 6F CA F4 78 BB 72 AD 09 3C 2F 8B 20 26 79 86 11 4C 19 85 CD 0D F7 3F C5 5A CC 44 40 6B
33 68 9D 79 11 72 A6 8E 3A 9C 20 AA FC 57 D0 4C E9 24 2B 9E B1 90 00  [SW: No Error]
Time used: 84.000 ms

Send script end. Time used:  0.504 sec.
```

4751 39[redacted]9107

Expire 10/22

Extended card details

Card AID : A0 00 00 00 03 10 10
Application : Visa Debit
Card type : VISA
Pin try left : 3 Time(s)
Card issuer (possible) : TIM (Italy) 128KB GSM SIM
H3G (Italy) UMTS USIM
Vodafone (UK) SIM
Gemalto Santander Optelio TUI R7 with WG10 customized using Contactless interface

4751 39XX XXXX 9107
VALID FROM 10/17 EXPIRES END 10/22
MR OSSAMA AL-MALIKI

Figure 4-5: Sniffing Attacks on the testing cards

4.2.3. Replay Attack

After demonstrating the ease of launching a sniffing attack, this section highlights the vulnerability of EMV contactless cards to the replay attack. We show that an attacker can use the sniffed data to create their own counterfeit card to launch the replay attack. To do so, we loaded a Java contactless cards with the following sniffed information:

SI, signed SI', and the IB public certificate (in case of the SDA) so that the Java card can act as a counterfeit card. As illustrated in Figure 4-6, the attack can be successful if the attacker tries to pay for goods or services using the counterfeit card when the transaction is processed offline where the POS and the card approve or decline the transaction without the IB. In fact, some retail prefers to use an offline contactless transaction instead of the online for several reasons such as the cost of the connectivity and even the availability of the connectivity. For example, most of the POS at the London Underground Stations are using an offline contactless transaction [7]. In this case, the attacker's card can fool the POS and act as a genuine EMV contactless card in case of the SDA. This is mainly because POS gets a positive outcome when upon verifying the signed version of SI using the plaintext version of the SI and the IB public certificate, which leads to approving the offline transaction.

It is essential to highlight that after the attacker uses the services or collects the goods, the POS sends later the transaction information to the IB through the AB to process the transaction. At that point, the IB declines the transaction because the AC is not signed by the C_{sk} as shown in Figure 4-6. However, this might be too late.

On the other hand, the details of DDA and CDA in Section 4.1 shows that the above replay attack has no chance of success even in offline transactions due to the negative output that the POS gets when attempting to verify the signed version of the TD in case of the DDA and the signed version of TD and AC in case of the CDA, as the attacker does not have the C_{sk} to sign the TD and AC. Therefore, the POS rejects the transaction initiated by the attacker using the counterfeit card as shown in Figure 4-6.

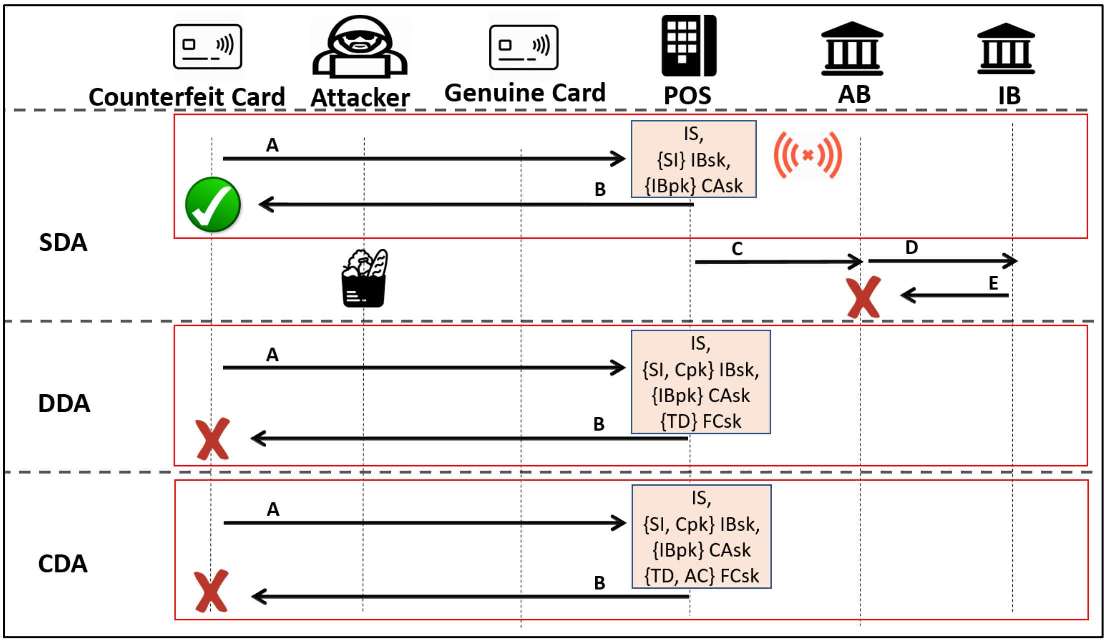


Figure 4-6: Replay Attacks on the three EMV Card Authentication Methods

4.3. Discussion & Recommendations

This section is an attempt to unwrap the EMV authentication methods and evaluate the security of the three EMV authentication methods. To underpin the exact source of vulnerabilities, this chapter explains the five stages of the authentication methods, namely the keys distribution, Generation of IB public key certificate, card personalisation, transaction and authentication stages. Moreover, we presented a Java framework and made it publicly accessible to other researchers to simulate all the stages of an EMV contactless transaction under the three authentication methods as well as simulating the sniffing and replay attacks using Java contactless cards.

Section 4.1 showed that the SDA, which usually has the lowest priority amount of the three EMV authentication methods, is vulnerable to the two attacks. However, the SDA still provides strong evidence to the POS that the card's static information is genuine and guarantees its integrity after the personalisation phase.

The DDA, on the other hand, has a higher level of security than the SDA as it provides a countermeasure against the replay attack via the unique signature for each transaction. However, the DDA is still vulnerable to the sniffing attack for the same reasons the SDA does. In addition to providing strong evidence about the authenticity and integrity of the EMV card's static data to the POS, it provides the EMV card approval of the TD by signing the TD using C_{sk}.

The CDA provides another layer of security and thus it often given the highest priority among the three EMV authentication methods. In addition to guaranteeing the integrity and authenticity of the static data, and signing the TD, the CDA signs its own transaction decision by signing the AC together with the TD using C_{sk} to give another level of assurance to the POS that the card has indeed approved the transaction and prevent the possibility of non-repudiation attacks. Although the CDA is perceived to be highly secure, this chapter demonstrated its vulnerability to sniffing attacks that can be easily launched to steal credit/debit card details.

Our final recommendations after breaking down the three EMV card authentication methods are as follow:

1- The SDA card authentication method should not be used in the new issued EMV cards due to the low level of security that the SDA is provided comparing with both DDA and CDA.

2- A mutual Authentication protocol must be introduced to the EMV payment protocol, especially for the EMV contactless cards to prevent such cards from being read by any unauthorised NFC enabled readers/smartphones.

Table 4-1 summarises most of the different characteristics of the three EMV authentication methods. The table summarises our findings of the three EMV authentication methods. For instance, the SDA requires one certificate to be stored inside the card while both DDA and CDA require two certificates in order to be processed by the POS. Also, the EMV cards which support just the SDA are using symmetric cryptography while the EMV cards which support DDA or CDA are using asymmetric cryptography where each card has its own RSA pair of keys to be use in order to process the supported card authentication method.

Table 4-1: Overview of the EMV Authentication Methods

Characteristics	**EMV Card Authentication Methods**

	SDA	DDA	CDA
Can the method prove the genuinely of the EMV card to the POS/ATM?	Yes, in case of online transactions No, in case of offline transactions	Yes, in both of online/offline transactions	Yes, in both of online/offline transactions
Number of Required Certificates	1- $CA_{sk}\{IB_{pk}\}$	1- $CA_{sk}\{IB_{pk}\}$ 2- $IB_{sk}\{C_{pk}\}$	1- $CA_{sk}\{IB_{pk}\}$ 2- $IB_{sk}\{C_{pk}\}$
Number of digital signatures	1- $IB_{sk}\{SI\}$	1- $IB_{sk}\{SI\}$ 2- $C_{sk}\{TD\}$	1- $IB_{sk}\{SI\}$ 2- $C_{sk}\{TD,AC\}$
Digital signatures type	Static, valid for each transaction.	Dynamics, valid for only one transaction.	Dynamics, valid for only one transaction.
Symmetric/Asymmetric	Symmetric cryptography	Asymmetric cryptography	Asymmetric cryptography
Protection against sniffing attacks	Not Protect	Not Protect	Not Protect
Protection against replay attacks	Not Protect	Protect	Protect

4.4. Summary of the Chapter

Obtaining sensitive information from the EMV contactless cards is quite easy to perform by fraudsters using off-the-shelf hardware and software. The information could be used to launch different attacks as explained above. This due to the wireless connectivity of such cards and the one-way authentication protocol. Thus, we propose in Chapter 5 the use of a token replace the actual PAN in the EMV contactless card transactions in order to stop the fraudsters from getting any advantage of the skimming information.

It can be argued that the card authentication methods all share a fundamental flaw related to the trust model between the POS and the EMV contactless card. We showed that the POS enforces strict rules to ensure the authenticity of the card, the integrity of its information, and even getting the card to sign the TD and the final decision. In contrast, the chapter showed that the contactless cards release sensitive information to anyone with NFC enabled readers/smartphones without any checks whatsoever. Our simulations showed that all what it takes is to send the commands in the correct format as per the EMV publicly available specifications. Therefore, we propose in Chapter 6 the use of a mutual authentication protocol between the contactless cards and the POSs to ensure that such cards do not leak any sensitive information unless it gets the right level of assurance that the POS is a genuine one.

5. Chapter 5: The Proposed Tokenisation Protocol to Improve the Security of EMV Contactless Cards

This chapter focuses on the EMV contactless cards and its vulnerability of leaking sensitive information such as the cardholder name, PAN and the expiry date of the EMV cards. Such data can be sniffed using off-the-shelf hardware or software without the knowledge of the genuine cardholder. The chapter proposes a tokenisation approach to replace the EMV contactless card's PAN with a token to protect the genuine data from being sniffed by an attacker and used in the CNP attack or any other attacks.

The proposal was inspired by the implementation of tokenisation in the EMV mobile payment such as Apple, Google and Samsung mobile payments [83]. We argue that the proposed tokenization technique is easy to adopt and cost-effective to implement by EMV payment protocol as it does not require any changes to the infrastructure of existing payment systems. A vital feature of the proposal is that all the changes in the EMV protocol are at the personalisation phase of the EMV card. The chapter presents a successful implementation of the tokenisation approach using the Java contactless card to represent EMV contactless cards to demonstrate its effectiveness in improving the security and protecting the privacy of the card's information.

The chapter is organized as follows. Section 5.1 details the problem statement and the motivations behind our proposal. Next, Section 5.2 explains and analyses the EMV mobile tokenisation payment and highlights the relevance to our proposal. Then, Section 5.3 describes the proposed tokenisation technique for the EMV contactless card and explains the implementation details across the three different phases, namely personalisation, transaction and authorisation. Later, Section 5.4 shows our trials and implementation of the tokenisation technique leaving Section 5.6 to conclude the work.

5.1. Problem Statement & Motivation

The PAN tokenisation for mobile payments such as the Apple Pay, Google Pay, and Samsung Pay is based on the EMV specification for the mobile payment, in which the mobile does not send the actual PAN to the POS [84][85][86]. Instead, the phone sends a token of 16- digit to represent the actual PAN. It can be argued that the tokenisation implemented based on the EMV mobile payment specification offers a higher level of security than the contactless card specification. Furthermore, the EMV mobile payment needs to be initialized by the cardholder by either entering a password, PIN or even by using a biometric to activate the EMV mobile payment [87]. On the other hand, the EMV contactless card specifications offer no cardholder verification at all. Moreover, sensitive information is sending in plaintext to the POS or any NFC reader, which be easily obtained by an attacker. Therefore, we propose a tokenisation technique to protect the PAN and improve the privacy of the cardholder during the EMV contactless card payment.

The main motivation behind proposing our EMV contactless card tokenisation is the EMV mobile payment tokenisation. The EMV mobile payment tokenisation was proposed by the EMV protocol to minimize the risk of unauthorised use of the PAN in such kind of transactions. We did ask ourselves the question "why this is a tokenisation technique implemented on EMV mobile payment while there is not on the EMV contactless card?". Moreover, there is a cardholder verification on the EMV mobile payment while there is no such verification on the EMV contactless card payment.

5.2. Analyses of the EMV Mobile Payment Tokenisation

To minimize the changes to the existing EMV payment infrastructure, the proposal reuses some of the components in mobile payment tokenisation. This section details the EMV mobile payments and focuses on the three phases of the EMV tokenisation system, which are the token generation, identification & verification and processing during the transaction phases. This section also explains the Payment Account Reference (PAR) term that is used in such transactions.

5.2.1. EMV Mobile Payment Tokenisation Phases

The EMV mobile payment tokenisation system and all the parties that are involved in the processing such as the IB, Token Service Provider (TSP), cardholder, Cardholder's Smartphone (SP), Token Requester (TR), POS and the AB as shown in Figure 5-1.

Token Requester (TR) represents the payment wallets such as Apple, Samsung and Google Payment wallets that installed at the cardholder’s smartphone or any gadgets such as smartwatches. When the cardholders want to use EMV mobile payment, they need to input their EMV card’s details (PAN, cardholder name, card expired date and the card verification code (CVC)) to the payment wallet that the cardholders wish to use as shown in step (1) in Figure 5-1.

The Token Service Provider (TSP) represents a role within the EMV payment ecosystem. That’s mean each TSP should be registered with the EMV ecosystem to generate a token for each PAN that been requested by the TR. The generated token cannot be reverse engineered to determine the actual PAN. In addition, the TSP could be the IB itself or even a third party that the IB has a contract with it to manage the tokens generation and the D-Tokenisation process on behalf of the IB.

Token Generation

In this phase, the PAN token is requested by the Token Requester (TR) and generated by as illustrated in Figure 5-1. The TSP then verifies the cardholder to make sure that the token is generated to the genuine cardholder who owns the PAN. The PAR is signed to each PAN by the TSP. Then the TR sends all the card’s details to the TSP to request a token to surrogate the actual PAN and use this token in the mobile payment as shown in step (2) in the same figure.

When the TSP receives the TR’s request, the TSP forwards the request to the appropriate IB to make sure all the card details are valid as shown in step (3). If the IB approves that all the card’s details are valid, and the cardholder is a rightful cardholder. Then, the TSP generates a token to replace the actual PAN and sends the generated token to the TR as shown in step (7) in Figure 5-1. This token is used in the EMV mobile payment instead of the real PAN.

The EMV tokenisation specifications offer the IBs and TSPs the right to choose how to generate the token in any way that ensures the security of the generated token and the generated token cannot be reverse engineered to determine the actual PAN. These some common ways are used to token generation [88]:

- A mathematically reversible cryptographic function based on a known robust cryptographic algorithm and a secure cryptographic key.

- A one-way non-reversible cryptographic function such as hash function with secret and robust seed.
- Assignment through an index function such as the use of lookup table, sequence number or a randomly generated number that not mathematically derived from the actual PAN.

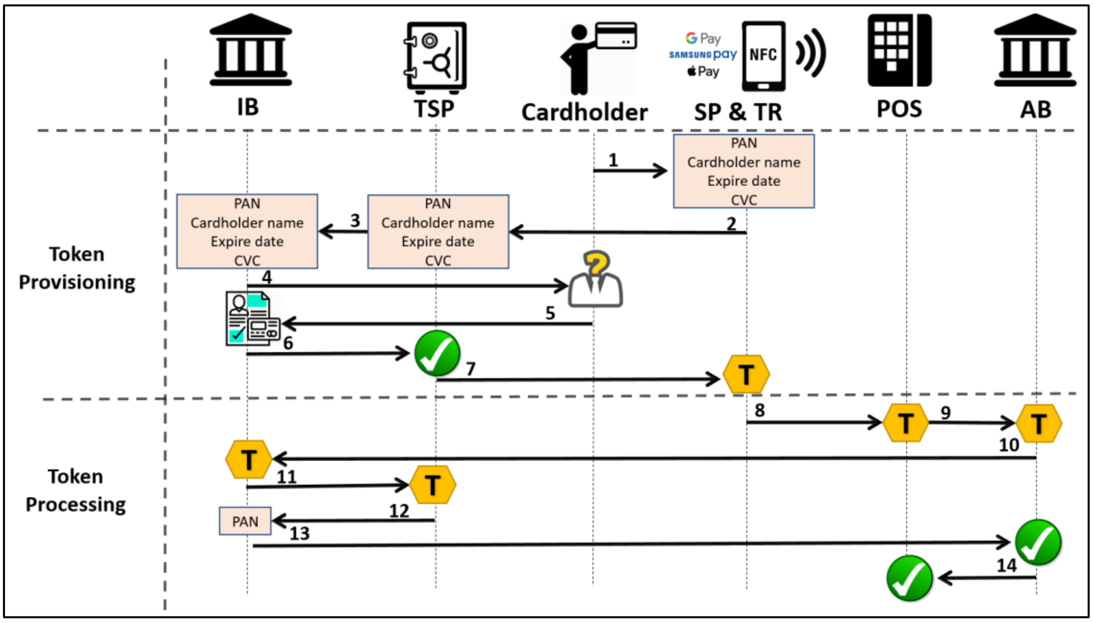
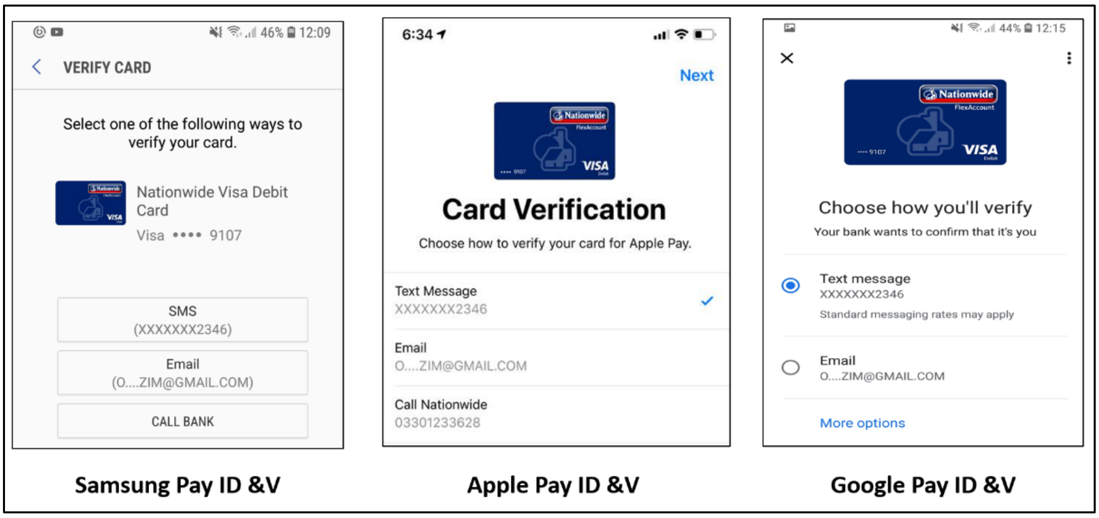


Figure 5-1: EMV Mobile Payment Tokenisation System

Cardholder Identification & Verification (ID&V)

When the IB receives the Card's details from the TSP as shown in step (3) in Figure 5-1, the Identification and Verification (ID &V) phase starts to make sure that the person who requested the token is the genuine cardholder of the EMV card. This could be done by using many different verification methods such as SMS, Email and even calling from the IB as shown in Figure 5-2.

For example, both email and phone numbers in Figure 5-2 were used at the ID &V methods are requested and stored by the IB at the beginning of opening the account of the EMV card. That's mean the IB sends an activation code to either the email or the phone number of the cardholder as shown in step (4) in Figure 5-1. If the cardholder successfully verified by the IB as shown in step (5) in the same figure, then, the IB sends an approval to the TSP to approve the process of generating a token as shown in step (6). Then, the TSP generates a token and token related data to replace the PAN in the mobile payment.



VERIFY CARD

Select one of the following ways to verify your card.

Nationwide Visa Debit Card
Visa •••• 9107

SMS
(XXXXXXX2346)

Email
(O....ZIM@GMAIL.COM)

CALL BANK

Samsung Pay ID &V

Next

Card Verification

Choose how to verify your card for Apple Pay.

Text Message ✓
XXXXXXX2346

Email
O....ZIM@GMAIL.COM

Call Nationwide
03301233628

Apple Pay ID &V

Choose how you'll verify

Your bank wants to confirm that it's you

● Text message
XXXXXXX2346
Standard messaging rates may apply

○ Email
O....ZIM@GMAIL.COM

More options

Google Pay ID &V

Figure 5-2: Identification & Verification Methods in EMV Mobile Payment

Token Processing During the Transaction

The token Processing phase starts after the process of generating the token and its related data at the provisioning phase. In other words, this phase takes place at the mobile transaction time. The token and its related data are sent to the POS instead of the actual PAN at the transaction time as shown in Figure 5-1 step (8). Then, in case of the online mobile-transaction, the POS sends the transaction information along with the token in real-time to the IB through its AB as shown in both steps (9 & 10) in the same figure. While in case of the offline mobile transaction these data are sent to the IB at the end of the day to authorise the transaction [89]. When the IB receives the token, it forwards it and its related data to the TSP to request the actual PAN which associate with the token as shown in step (11). Next, the TSP checks the token and its related data to ensure its validity by verifying the related data of the token. If the TSP verifies the token as a genuine one, then, the TSP sends the actual PAN back to the IB as shown in step (12) in the same figure. Finally, the IB performs several checks on the account that belongs to the actual PAN to decide on whether to approve or decline the mobile transaction. The decision is then sent back to the POS through the AB as shown in steps (13 & 14) in Figure 5-1.

5.2.2. Payment Account Reference (PAR)

The EMV mobile payment tokenisation introduces one significant security advantage in the EMV payment ecosystem by replacing the PAN with a token which minimizes the risk of compromising the PAN and using it in different attacks such as the CNP attack. However, the PAN tokenisation technique makes it possible for multiple tokens of the same PAN to be generated and used in different payment wallets or devices as shown in Figure 5-3, where our genuine EMV contactless card was registered with three different mobile payment wallets. As a result, the PAN was liked with three different tokens.

This usually makes it challenging for retailers and merchants to keep a link with the historical transactions of the actual PAN. Therefore, the EMV specification for mobile payment has introduced the Payment Account Reference (PAR) since 2014 to address the above challenge in addition to security and regulatory reasons such as risk analysis and anti-money laundering [90]. The PAR can be used to link cardholder’s payment tokens with their PAN without the need to use the underlying account number, which enables retailers to move away from dependence on the PAN as the primary linkage. The PAR consists of 29 characters where the first four characters represent the Bank International Number (BIN) while the other 25 characters are a unique value for each PAN.

In addition, the EMV mobile payment specification recommends that the PAR should be available to the POS each time a mobile transaction is being processed i.e. at the time of the transaction, the PAR should be sent to the POS as part of the response of reading record command[91]. To test the idea, we have registered our EMV contactless card to three different mobile payment wallets (Google, Samsung and Apple pay) as shown in Figure 5-3. Each of these wallets generates a different token to replace the same PAN. Moreover, we showed that PAR is the same for all the tokens.

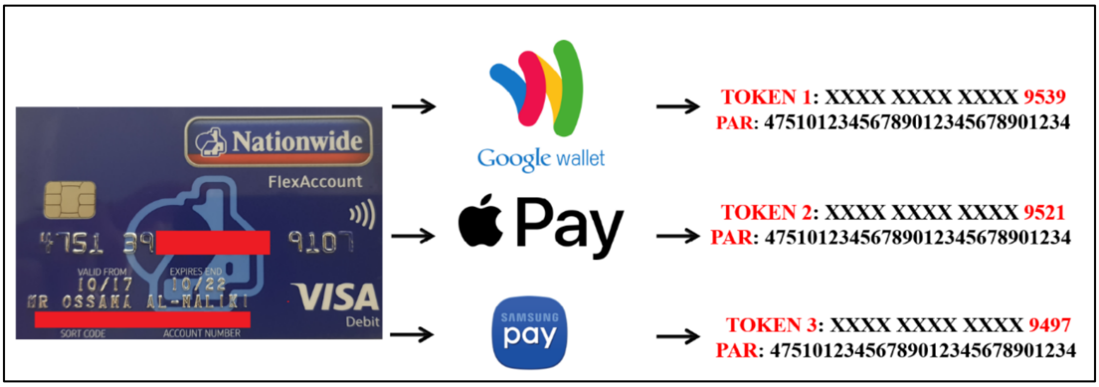

Nationwide
FlexAccount
[redacted]
VALID FROM 10/17
EXPIRES END 10/22
[redacted]
VISA
Debit
SORT CODE
ACCOUNT NUMBER
Google wallet
Pay
SAMSUNG pay
TOKEN 1: XXXX XXXX XXXX 9539
PAR: 47510123456789012345678901234
TOKEN 2: XXXX XXXX XXXX 9521
PAR: 47510123456789012345678901234
TOKEN 3: XXXX XXXX XXXX 9497
PAR: 47510123456789012345678901234

Figure 5-3: Payment Account Reference (PAR)

5.3. The Proposed Tokenisation Technique for Contactless Cards

As stated earlier, the EMV mobile tokenisation described in the previous section was one of the principal motives of the proposed technique. This section presents the required changes to the existing EMV protocol of contactless cards at the three key phases, namely card personalisation, transaction, and authorisation phases. We designed our proposed tokenisation in such a way that it does not require extra hardware or software to the EMV payment ecosystem and ensuring that the changes to the EMV protocol of contactless cards at all the three phases are implemented by reusing some of the mobile payment infrastructure components.

5.3.1. The Proposed Personalisation Phase

The contactless card tokenisation approach starts at the personalisation phase (card-issuing phase) as shown in Figure 5-4. In the original EMV personalisation phase, the IB uploads the EMV card within the cardholder's name and assigns it with a unique 16-digits PAN along with a card expiry date and various information. The IB then signs the above information by its private key (IB_{sk}). Finally, the IB uploads the EMV card with the signed version of the card's information. The card is then ready to be handed over to the cardholder.

Table 5-1 shows some examples of different token generation methods where the token consists of alphabetic and numeric characters or just numeric characters to replace the actual PAN as shown in Table 5-1. Here in this proposal, we propose the use of the third method where the generated token consists of the same first 6 digits of the actual PAN. The reason behind that is to make it possible to the POS to identify the IB by the use of the Bank Identification Number (BIN) which is provided by the token (which is represented by the six digits). On the other hand, the remaining of the 10 digits are unique for each PAN and these digits could not be reverse engineered to find the actual PAN, we suggest using a lookup table to link each token with its related PAN as shown in Figure 5-4.

Table 5-1: Examples of different token generation methods

PAN	Token	Comment

4751 3911 1111 9107	5A43 8BCC 79AE F1E8	Random generation of 16 alphabetic and numeric characters
4751 3911 1111 9107	1234 5678 9012 3456	Random generation of 16 numeric characters only
4751 3911 1111 **9107**	**4751 39**12 34 **9107**	Token consists of truncated PAN (first 6, last 4 of PAN are the same), the rest consist of randomly generated numbers

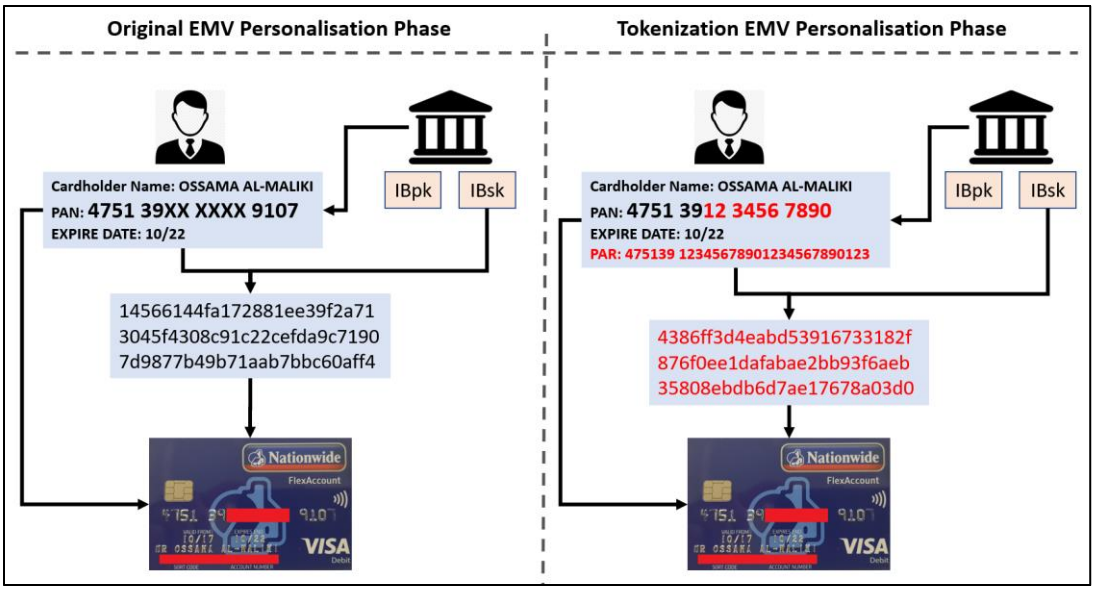

Original EMV Personalisation Phase
Tokenization EMV Personalisation Phase
Cardholder Name: OSSAMA AL-MALIKI
PAN: 4751 39XX XXXX 9107
EXPIRE DATE: 10/22
IBpk
IBsk
14566144fa172881ee39f2a71
3045f4308c91c22cefda9c7190
7d9877b49b71aab7bbc60aff4
Nationwide
FlexAccount
VISA
Debit
Cardholder Name: OSSAMA AL-MALIKI
PAN: 4751 3912 3456 7890
EXPIRE DATE: 10/22
PAR: 475139 1234567890123456789O123
IBpk
IBsk
4386ff3d4eabd53916733182f
876f0ee1dafabae2bb93f6aeb
35808ebdb6d7ae17678a03d0
Nationwide
FlexAccount
VISA
Debit

Figure 5-4: Original VS. Tokenisation EMV Personalisation Phase

5.3.2. The Proposed Transaction Phase

Although most of the key proposed changes to the existing EMV contactless card protocol are at the personalisation phase, the tokenisation technique brings minor but vital changes to the message exchanges between the POS and the EMV contactless card at the transaction phase as shown in Figure 5-5. The figure shows that the POS's commands are the same without any changes. However, the EMV contactless card's responses need to be updated in just two commands, namely the "Read Record" and the "Generate AC" APDU commands as showed in the same figure in steps 11 & 15.

In the "Read Record" APDU command, the Original EMV contactless card returns all card's information such as the cardholder's name, PAN, expiry date, and the signed version of the data while the card in the proposed tokenisation approach responds with the cardholder name, expiry date, the PAN's token (created at the personalisation phase) and the PAR along with the signed version of all the data. For the "Generate AC" APDU command, the original EMV contactless card returns either the DDA or the CDA generated based on the actual PAN. The proposed solution generates both DDA and CDA based on the token instead of the actual PAN where the POS verifies both DDA and CDA in the same way that the original EMV contactless card does.

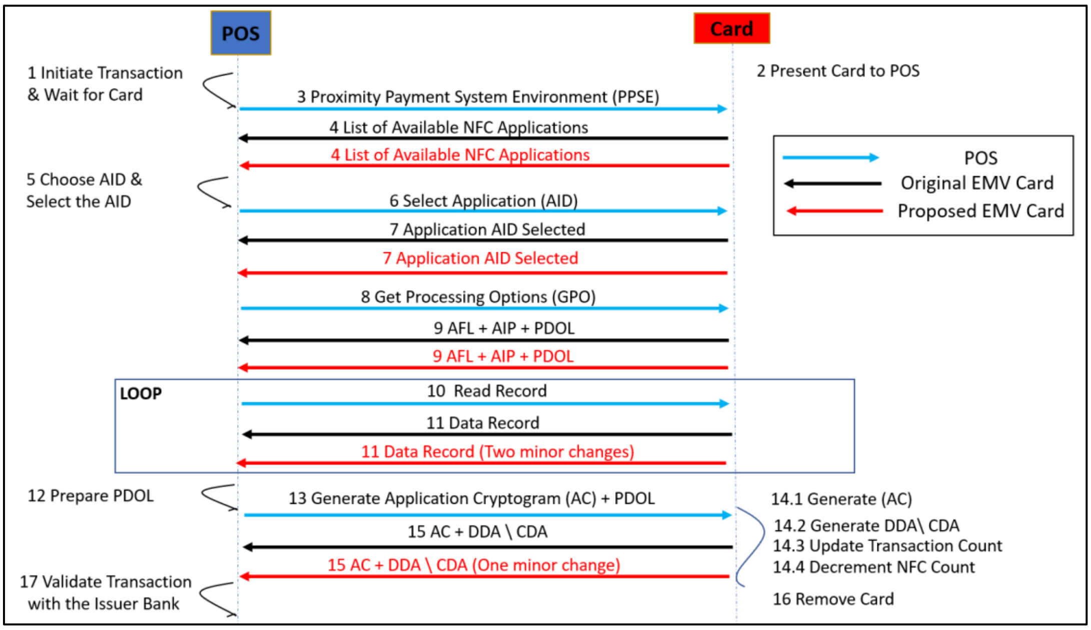

POS
Card
1 Initiate Transaction & Wait for Card
2 Present Card to POS
3 Proximity Payment System Environment (PPSE)
4 List of Available NFC Applications
4 List of Available NFC Applications
POS
Original EMV Card
Proposed EMV Card
5 Choose AID & Select the AID
6 Select Application (AID)
7 Application AID Selected
7 Application AID Selected
8 Get Processing Options (GPO)
9 AFL + AIP + PDOL
9 AFL + AIP + PDOL
LOOP
10 Read Record
11 Data Record
11 Data Record (Two minor changes)
12 Prepare PDOL
13 Generate Application Cryptogram (AC) + PDOL
14.1 Generate (AC)
14.2 Generate DDA\ CDA
14.3 Update Transaction Count
14.4 Decrement NFC Count
15 AC + DDA \ CDA
15 AC + DDA \ CDA (One minor change)
17 Validate Transaction with the Issuer Bank
16 Remove Card

Figure 5-5: Original VS. Tokenisation EMV Transaction Phase

5.3.3. The Proposed Authorisation Phase

In the original EMV protocol of contactless cards at this phase, the POS sends the actual PAN with the transaction's details to the IB through the AB to approve or decline the transaction. The tokenisation solution, on the other hand, proposes that the POS sends the token instead of the actual PAN to the IB through the AB. The IB then checks the token against a lookup table to find the associated actual PAN. If the PAN was verified by the IB and the account has enough credit, then the IB approves the transaction and sends the decision back to the POS through the AB as shown in Figure 5-6. Otherwise, the IB declines the transaction.

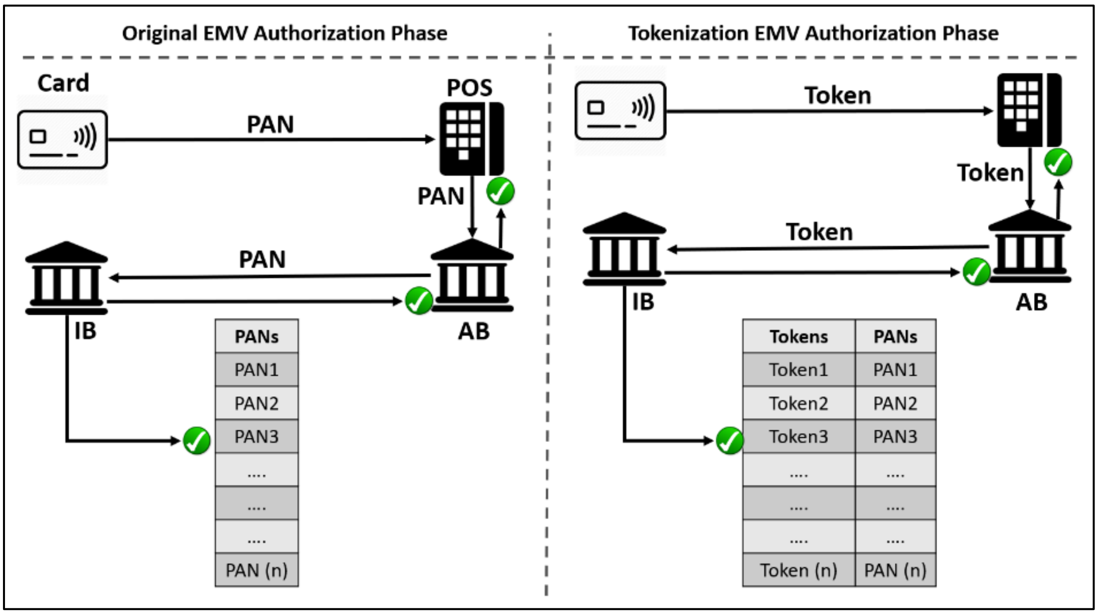


Figure 5-6: Original VS. Tokenisation EMV Authorisation Phase

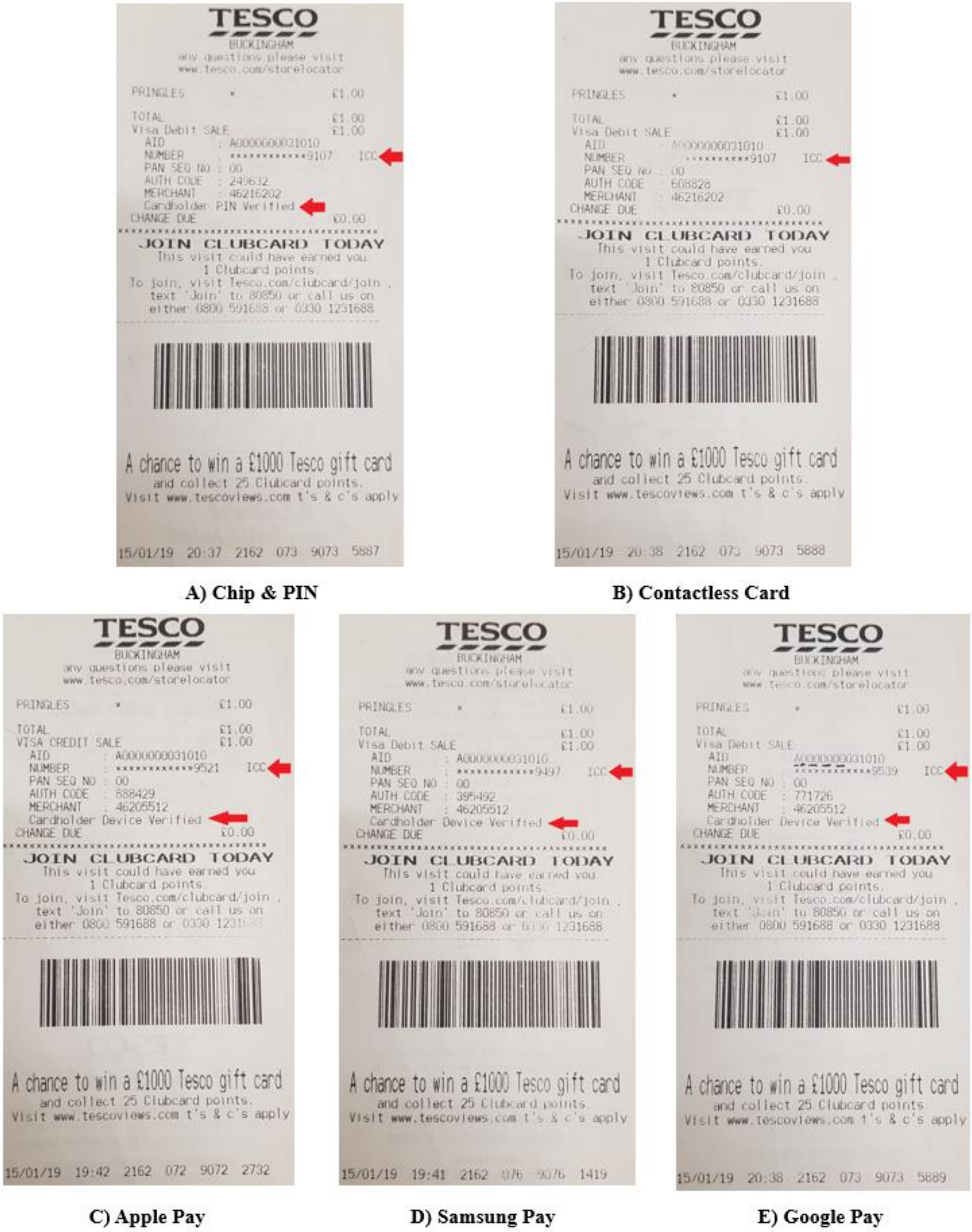
5.4. Trials, Implementations and Discussion

The aim of this section is to demonstrate the feasibility of the proposal by presenting relevant trials and implementations in addition to explaining the underlying hardware and software used for generating the results.

The same hardware and software in Chapter 3 were used to do the trials and the tokenisation protocol's implementations.

5.4.1. Tokenisation Trials

The aim of the trials is to uncover the practical details of mobile tokenisation techniques so they can be reused in the proposed solution to minimize the changes to EMV payment systems. To do so, we have registered a genuine EMV contactless card in three different mobile wallets: Apple, Google and Samsung Pay to illustrate the mechanism of the underlying EMV mobile payments and compare it to other payment mechanisms. Moreover, we did five different transactions as shown in Figure 5-7.



```
TESCO
BUCKINGHAM
any questions please visit
www.tesco.com/storelocator

PRINGLES          *              £1.00

TOTAL                            £1.00
Visa Debit SALE                  £1.00
  AID         : A0000000031010
  NUMBER      : ************9107    ICC
  PAN SEQ NO  : 00
  AUTH CODE   : 249632
  MERCHANT    : 46216202
  Cardholder PIN Verified
CHANGE DUE                       £0.00
**************************************
  JOIN  CLUBCARD  TODAY
   This visit could have earned you
        1 Clubcard points.
To join, visit Tesco.com/clubcard/join ,
   text 'Join' to 80850 or call us on
   either 0800 591688 or 0330 1231688

A chance to win a £1000 Tesco gift card
   and collect 25 Clubcard points.
Visit www.tescoviews.com t's & c's apply

15/01/19  20:37  2162  073  9073  5887
```

A) Chip & PIN

```
TESCO
BUCKINGHAM
any questions please visit
www.tesco.com/storelocator

PRINGLES          *              £1.00

TOTAL                            £1.00
Visa Debit SALE                  £1.00
  AID         : A0000000031010
  NUMBER      : ************9107    ICC
  PAN SEQ NO  : 00
  AUTH CODE   : 608828
  MERCHANT    : 46216202
CHANGE DUE                       £0.00
**************************************
  JOIN  CLUBCARD  TODAY
   This visit could have earned you
        1 Clubcard points.
To join, visit Tesco.com/clubcard/join ,
   text 'Join' to 80850 or call us on
   either 0800 591688 or 0330 1231688

A chance to win a £1000 Tesco gift card
   and collect 25 Clubcard points.
Visit www.tescoviews.com t's & c's apply

15/01/19  20:38  2162  073  9073  5888
```

B) Contactless Card

```
TESCO
BUCKINGHAM
any questions please visit
www.tesco.com/storelocator

PRINGLES          *              £1.00

TOTAL                            £1.00
VISA CREDIT SALE                 £1.00
  AID         : A0000000031010
  NUMBER      : ************9521    ICC
  PAN SEQ NO  : 00
  AUTH CODE   : 888429
  MERCHANT    : 46205512
  Cardholder Device Verified
CHANGE DUE                       £0.00
**************************************
  JOIN  CLUBCARD  TODAY
   This visit could have earned you
        1 Clubcard points.
To join, visit Tesco.com/clubcard/join ,
   text 'Join' to 80850 or call us on
   either 0800 591688 or 0330 1231688

A chance to win a £1000 Tesco gift card
   and collect 25 Clubcard points.
Visit www.tescoviews.com t's & c's apply

15/01/19  19:42  2162  072  9072  2732
```

C) Apple Pay

```
TESCO
BUCKINGHAM
any questions please visit
www.tesco.com/storelocator

PRINGLES          *              £1.00

TOTAL                            £1.00
Visa Debit SALE                  £1.00
  AID         : A0000000031010
  NUMBER      : ************9497    ICC
  PAN SEQ NO  : 00
  AUTH CODE   : 395492
  MERCHANT    : 46205512
  Cardholder Device Verified
CHANGE DUE                       £0.00
**************************************
  JOIN  CLUBCARD  TODAY
   This visit could have earned you
        1 Clubcard points.
To join, visit Tesco.com/clubcard/join ,
   text 'Join' to 80850 or call us on
   either 0800 591688 or 0330 1231688

A chance to win a £1000 Tesco gift card
   and collect 25 Clubcard points.
Visit www.tescoviews.com t's & c's apply

15/01/19  19:41  2162  076  9076  1419
```

D) Samsung Pay

```
TESCO
BUCKINGHAM
any questions please visit
www.tesco.com/storelocator

PRINGLES          *              £1.00

TOTAL                            £1.00
Visa Debit SALE                  £1.00
  AID         : A0000000031010
  NUMBER      : ************9539    ICC
  PAN SEQ NO  : 00
  AUTH CODE   : 771726
  MERCHANT    : 46205512
  Cardholder Device Verified
CHANGE DUE                       £0.00
**************************************
  JOIN  CLUBCARD  TODAY
   This visit could have earned you
        1 Clubcard points.
To join, visit Tesco.com/clubcard/join ,
   text 'Join' to 80850 or call us on
   either 0800 591688 or 0330 1231688

A chance to win a £1000 Tesco gift card
   and collect 25 Clubcard points.
Visit www.tescoviews.com t's & c's apply

15/01/19  20:38  2162  073  9073  5889
```

E) Google Pay

Figure 5-7: Five Different Transactions by the Same EMV Contactless Card

- Transaction (A) is chip & PIN transaction where the actual PAN (just the last 4-digits as the rest 12-digits have been masked) is showing in the customer's receipt and the cardholder has been verified by the PIN.

- Transaction (B) is a contactless card transaction where the actual PAN is showing in the customer's receipt and there is no cardholder verification.

- Transactions (C, D, and E) are mobile transactions for Apple, Samsung and Google pay respectively. The figure confirms that the three transactions were accomplished based on three different tokens used to replace the one actual PAN.

Here in this chapter, we argue the cardholder in the last three mobile transactions had to use his biometrics on the smartphone to verify himself to activate the payment wallet to release the token, which is less sensitive than the actual PAN. On the other hand, a contactless card releases the actual PAN and sends it in plaintext to any POS or any contactless reader without any kind of cardholder verifications. As stated earlier, the above argument was the critical motive proposing the EMV tokenisation for contactless cards to add an extra level of security to the PAN.

5.4.2. Implementations of the Tokenisation Proposed Protocol

The main goal behind the implementation is to illustrate the feasibility of the proposed tokenisation technique for the EMV contactless cards with only a few minor changes to the existing EMV payment protocol. To do this, we used the JCIDE to develop two Java contactless applets, one is a replica of an original EMV contactless card as already detailed in Chapter 3. While the other applet used to implement and simulate the proposed tokenisation techniques for the contactless cards.

The applet of the original EMV uses the actual PAN in the contactless transaction whereas the applet of the proposal uses a token instead of the actual PAN in the simulated contactless transactions. We used a lockup table (as shown in Figure 5-4) to link the PANs with their tokens and PARs. Both applets follow the personalisation phases illustrated in Figure 5-4 where the first applet follows the original EMV personalisation phase while the second follows the proposed personalisation phase by employing a token instead of the actual PAN and incorporating the use of the PAR. Moreover, both applets support the main five APDU commands and their responses. The supported commands are “PPSE”, “Select AID”, “GPO”, “Read Record” and “Generate AC” APDU commands as shown in Figure 5-5 with the blue arrows.

After uploading and installing the two applets into the two Java contactless cards, both applets were testing using the NFC ACR 122U reader. The NFC reader acts as a genuine POS that sends EMV commands to our Java contactless identical to those sent by a real-life POS to standard EMV contactless cards. We also implemented the two applets in such a way that they can respond with proper EMV responses as shown in Table 5-2. The table shows the simulation results of all the five POS EMV commands and their responses for both the original EMV and the proposed tokenisation technique.

The output shown in Table 5-2 does not only demonstrate the successful implementation and simulation of the proposed tokenisation technique using the Java contactless card and the NFC reader but also illustrates how minimal the required changes to the existing EMV protocol were. Both applets respond to the POS EMV first three commands (“PPSE”, “Select AID” and “GPO”) with responses identical to those of the original EMV protocol, However, the last two responses for both of “Read record” and “Generate AC” commands were updated in the proposed tokenisation applets as shown in Table 5-2.

- The first record in the Original EMV contactless applet consists of three tags to represent the cardholder name with tag 5F20, the PAN with a tag number 5A, and the expiry date of the EMV card with tag 5F24. In the proposed applet, the first record consists of the same three tags with a token stored at the personalisation phase instead of the actual PAN. Moreover, the first record consists of the tag 9F24 to represent the PAR for the reason explained in Section 5.2.2.
- The second record contains the IB public key certificate, where the IB public key was signed by the certificate authority private key at the personalisation phase. Therefore, the record is the same in both the original and the proposal applets.
- The third record in the proposed applet is different from the original EMV one as it contains the signed version of the EMV card’s details including the introduced token instead of the PAN, and the PAR that was not used in the original EMV contactless card.
- The last responses of the last POS command “Generate AC” are different in each applet for the same reason stated above with one difference that the data must be signed by the session key between the IB and the EMV card.

Table 5-2: Applets testing for Tokenisation Implementations

EMV Commands	POS APDU commands	Original Applet Responses	Proposal Applet Responses
PPSE	00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 00	6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90	6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90
Select Application (AID)	00 A4 04 00 07 A0 00 00 00 03 10 10 00	6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 90 00	6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 90 00
Get Processing Options (GPO)	80 A8 00 00 23 83 21 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	77 4D 82 02 20 00 94 04 10 02 05 00 57 13 47 51 39 05 86 29 91 07 D2 21 02 21 99 20 00 00 00 00 0F 5F 34 01 00 9F 10 07 06 01 0A 03 A0 00 00 9F 26 08 F4 E1 96 C3 0D 93 CD 81 9F 27 01 80 9F 36 02 00 5E 9F 6C 02 2E 00 9F 6E 04 20 70 00 00 90 00	77 4D 82 02 20 00 94 04 10 02 05 00 57 13 47 51 39 05 86 29 91 07 D2 21 02 21 99 20 00 00 00 00 0F 5F 34 01 00 9F 10 07 06 01 0A 03 A0 00 00 9F 26 08 F4 E1 96 C3 0D 93 CD 81 9F 27 01 80 9F 36 02 00 5E 9F 6C 02 2E 00 9F 6E 04 20 70 00 00 90 00
Read Record 1	00 B2 01 0C 00	24 5F 20 16 4F 53 53 41 4D 41 20 41 4C 2D 4D 41 4C 49 4B 49 5A 08 47 51 39 XX XX XX 91 07 5F 24 03 22 10 23 90 00	36 5F 20 16 4F 53 53 41 4D 41 20 41 4C 2D 4D 41 4C 49 4B 49 5A 08 47 51 39 12 34 56 78 90 5F 24 03 22 10 23 9F 24 15 47 51 39 12 34 56 78 90 12 34 56 78 90 12 30 90 00
Read Record 2	00 B2 02 0C 00	90 80 7F 58 2F 18 1C 14 8B 11 54 12 C9 2C 2A D0 64 A6 AF B3 93 84 5A EB 71 C0 A9 27 0D D0 B1 F0 86 06 54 1F 1E 95 F5 22 9D F8 F4 54 48 CC E7 9C DA D9 68 00 63 D5 44 71 AC 2C 02 00 7C 53 FF 8A AC BA DB 44 FF 1D 2E 4A 09 C9 6C 9B 72 01 B4 E8 EB A1 F2 2E 03 04 AC B7 00 F5 0C 06 A3 DF 0A A9 59 F5 84 02 3B C0 B1 C3 12 08 26 49 EA C3 03 42 EA E7 4F 2C 80 07 9C 3D 59 CA 80 00 0C 1C 8B 3B 0C 90 00	90 80 7F 58 2F 18 1C 14 8B 11 54 12 C9 2C 2A D0 64 A6 AF B3 93 84 5A EB 71 C0 A9 27 0D D0 B1 F0 86 06 54 1F 1E 95 F5 22 9D F8 F4 54 48 CC E7 9C DA D9 68 00 63 D5 44 71 AC 2C 02 00 7C 53 FF 8A AC BA DB 44 FF 1D 2E 4A 09 C9 6C 9B 72 01 B4 E8 EB A1 F2 2E 03 04 AC B7 00 F5 0C 06 A3 DF 0A A9 59 F5 84 02 3B C0 B1 C3 12 08 26 49 EA C3 03 42 EA E7 4F 2C 80 07 9C 3D 59 CA 80 00 0C 1C 8B 3B 0C 90 00
Read Record 3	00 B2 03 0C 00	93 80 7F 96 90 BD B7 03 B1 1E FA B8 53 C6 70 A2 61 0B 42 3C CE CE 99 B9 75 F4 57 AF 8F 6E 59 4A E8 AF 33 1D E6 A8 22 B1 AA 5D 54 C8 25 18 15 01 97 AB E8 C1 68 59 BB D8 42 7F 4C D4 90 42 36 F9 DB 87 B7 76 FB 4E 77 9C 86 CA 91 74 7D CA 35 51 49 B5 37 33 6C 38 3C 0E DC 3F FE DB 2A 39 48 07 B3 4C C5 69 34 87 3B CD AC 46 7F 3E 8E 8C A5 48 5C 22 9E 0F 32 D0 9B C5 05 34 7E 54 D2 AE FD E1 A0 90 00	93 80 7F 68 DA 7D 06 3B A1 99 C3 64 47 DD A0 25 EF 26 48 F3 86 0C AC 37 85 E4 55 B0 85 F6 D9 12 E6 58 68 D9 1A 82 DF 41 AE 58 D6 22 27 78 5F 97 48 9E 74 37 70 B8 02 A0 55 93 AF EF 18 A1 3F 9D 38 9E 4E 8D 52 7B 88 F6 6D E6 09 5F BA BC E6 7A 49 9D EE 4C BA AD 2C 7A E1 51 5B 8A 63 57 98 01 F8 4C CF 52 86 50 DE 65 EE 47 85 40 79 9F 62 5A 89 1D 50 AA 00 55 18 A8 2B 2C 86 A4 E6 F3 75 A3 C2 90 00
Generate Application Cryptogram (AC)	80 AE 40 00 1D 81 00 00 00 05 9F 04 00 00 00 00 9A 20 10 17 9F 21 04 00 00 5F 2A 82 60 9F 37 12 34 56 00	98 65 41 D2 7F 79 95 6F B5 55 34 86 74 F4 B1 ED 6F CA F4 78 BB 72 AD 09 3C 2F 8B 20 26 79 86 11 4C 19 85 CD 0D F7 3F C5 5A CC 44 40 6B 33 68 9D 79 11 A9 9C 4F 18 E7 75 11 A3 20 48 BF DF 6B 71 75 0B 90 00	98 65 41 D2 7F 79 95 6F B5 55 34 86 74 F4 B1 ED 6F CA F4 78 BB 72 AD 09 3C 2F 8B 20 26 79 86 11 4C 19 85 CD 0D F7 3F C5 5A CC 44 40 6B 33 68 9D 79 11 A9 9C 4F 18 E7 75 11 A3 20 48 BF DF 6B 71 75 0B 90 00

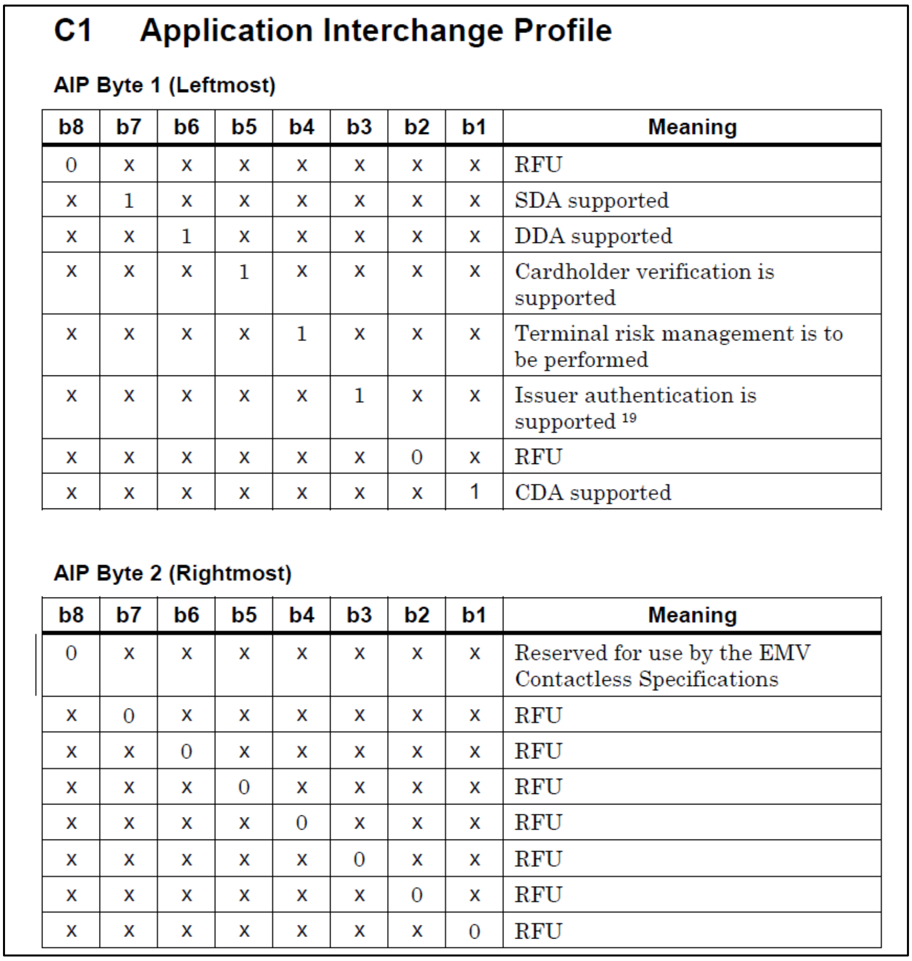
5.5. Discussions Related to the Tokenisation Protocol

The main advantage of deploying the proposed solution on EMV contactless card transactions is to stop attackers from sniffing the actual PAN using off-the-shelf hardware or software. This was mainly achieved by replacing the actual PAN with a token to be used for contactless card transactions. It could be argued that the token has a much lower security implication if it was captured by an attacker, as it cannot be used for online shopping or in lunching a CNP attack.

The chapter demonstrated the feasibility of implementing the proposal with minimal changes to the existing EMV payment infrastructure, which is one of the key advantages of the proposal. The deploying process only required a minor change into both the EMV card personalisation and the authorisation phases as discussed in Section 5.3. Therefore, the proposal requires no extra hardware or software as it reuses some of the infrastructure components that are already used by the EMV mobile payment systems.

We have implemented our proposed tokenisation approach to represent Visa EMV kernel 2 as shown in Figure 5-5. However, our proposed solution could be implemented successfully with all the other EMV contactless kernels even the Fast DDA kernel [35], due the proposed solution does not require any changes in the APDU commands and responses. Therefore, our proposal is ready to be deployed in all the seven different kernels that the EMV contactless card specifications are supported.

It can be argued that introducing any changes to the EMV protocol payment ecosystem would affect the existing 7.1 billion EMV based cards around the world [92]. Therefore, we recommend deploying the proposal on new EMV cards while the old cards continue to work using the current EMV protocol to avoid disturbing the whole EMV ecosystem. To achieve that, we propose the use of one bit from the "reserved for future" two bytes of the Application Interchange Profile (AIP). Using the bit as a flag to inform the POS of whether the EMV contactless card supports the proposal or not so the POS can respond accordingly. We recommend using bit number 8 (leftmost) of the second byte of the AIP as it is already reserved for the use of EMV contactless specifications according to EMV Book 3 as shown in Figure 5-8 [28].



C1 Application Interchange Profile

AIP Byte 1 (Leftmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	x	x	x	x	x	x	x	RFU
x	1	x	x	x	x	x	x	SDA supported
x	x	1	x	x	x	x	x	DDA supported
x	x	x	1	x	x	x	x	Cardholder verification is supported
x	x	x	x	1	x	x	x	Terminal risk management is to be performed
x	x	x	x	x	1	x	x	Issuer authentication is supported [19]
x	x	x	x	x	x	0	x	RFU
x	x	x	x	x	x	x	1	CDA supported

AIP Byte 2 (Rightmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	x	x	x	x	x	x	x	Reserved for use by the EMV Contactless Specifications
x	0	x	x	x	x	x	x	RFU
x	x	0	x	x	x	x	x	RFU
x	x	x	0	x	x	x	x	RFU
x	x	x	x	0	x	x	x	RFU
x	x	x	x	x	0	x	x	RFU
x	x	x	x	x	x	0	x	RFU
x	x	x	x	x	x	x	0	RFU

Figure 5-8: Application Interchange Profile

5.6. Summary of the Chapter

One key vulnerability in EMV contactless cards is the leakage of the PAN along with other sensitive information, which can be easily accessed by any unauthorised NFC enabled readers/smartphones. The chapter argued that the PAN with other sniffed information from a contactless card could be used by an attacker to launch one of the "Card Not Present" attacks. The chapter presented a simple but effective tokenisation technique to protect the PAN from being compromised in the EMV contactless card inspired by the EMV mobile tokenisation.

We argued that the proposed solution is cost-effective and easy to adopt by the EMV payment protocol due to the reuse of existing EMV infrastructures. To demonstrate that, we developed two applets based on two Java contactless cards to emulate the original EMV contactless cards and the proposal.

In addition to demonstrating a successful implementation and simulation of the proposed tokenisation technique using the Java contactless card and the NFC reader, the chapter illustrated how minimal the required changes to the existing EMV protocol were.

6. Chapter 6: The Proposed Mutual Authentication Protocol

The tokenisation proposal described in Chapter 5 stops the attackers from skimming the PAN and use it to launch various of attacks such as CNP attack. However, skimming and obtaining most of the sensitive information of such cards are still possible using off-the-shelf hardware. This is because the EMV card authenticates itself as a genuine card to the POS in each transaction while the reverse is not happening. An attacker can take an advantage of such vulnerability in the EMV specifications especially in contactless cards due to the wireless connectivity between the cards and POSs.

This chapter proposes a cost-effective mutual-authentication solution that relies on two-way challenge-response between the EMV contactless cards and the POSs in order to prevent sniffing attacks launched by NFC enabled readers or smartphones. To demonstrate the viability of the proposed authentication protocol, we present a Java framework to illustrate the practicality of the proposed solution. The chapter argues that the proposed protocol can be easily integrated into the EMV infrastructure with minor changes at the personalisation and transaction phases.

The chapter starts with the problem statement in Section 6.1. Then, Section 6.2 presents most of the related literature reviews. Next, the proposed mutual authentication between the EMV contactless cards and POS is discussed in Section 6.3. Later, Section 6.4 details the implementations and results of the proposal. Next, Section 6.5 presents three discussion points related to the proposal. Finally, Section 6.6 concludes the chapter.

6.1. EMV Authentication Problem Statement

As mentioned in Section 2.1.2.1, the EMV payment protocol supports three different card authentication methods namely SDA, DDA, and CDA. The main aim of these methods is to prove the authenticity of the EMV cards to POS/ATM whether in chip & PIN or contactless card transactions [27]. Nevertheless, the EMV payment protocol does not have any sort of authentication method to prove the authenticity of POS/ATM to the EMV cards. This vulnerable point in the EMV payment specifications allows unauthorised NFC enabled readers/smartphones to skim the EMV cards especially the EMV contactless cards due to the wireless connectivity of such cards. The skimmed information could be used to launch other attacks such as CNP and cloning attacks. Therefore, proposing a mutual authentication protocol is a must to handle the mentioned vulnerable point and prevent skimming such cards by unauthorised NFC enabled readers/smartphones.

6.2. Mutual Authentication Existing Protocols

Existing works proposed solutions based on mutual authentication between an NFC mobile payment and POSs for EMV mobile payment to ensure the confidentiality of the sensitive information of the EMV mobile payment [93] [94][95]. The first work was done by N. El Madhoun, F. Guenane, and G. Pujolle in [96] which proposed the use of a cloud-based secure authentication protocol for the EMV mobile payment. The protocol uses asymmetric cryptography to ensure mutual authentication and encryption of the payment information during the transaction. At the time of the transaction, the NFC mobile requests from a trusted cloud infrastructure to verify the POS before sending any sensitive payment information to the POS. If the POS was authenticated successfully, the cloud generates a session key to be used by NFC mobile and POS to encrypt the transaction details to ensure the confidentiality of the transaction data. Other existing proposal was proposed by U. B. Ceipidor, C. M. Medaglia and A. Moroni in [97] which was based on Needham-Schroeder protocol to ensure mutual authentication and confidentiality between the NFC mobile payment and POS. The protocol uses an authentication server to verify both, the NFC mobile and the POS and provide them with a session key to be used for encrypting transaction data.

Another mutual authentication solution was proposed by J. H. and A. van M. Martin Emms and Budi Arief in [98] for the EMV contactless cards in which the POS has its own pair of RSA public/private key along with the POS public key certificate. The POS needs to sign the EMV contactless card nonce by its own RSA private key and sends it back along with the POS's public key certificate to the EMV card. Then, the card retrieves the POS public key to verify its own nonce and authenticate the POS. However, the main limitation of such proposals is the size of the “APDU commands” where the maximum size of a single APDU command is 256 bytes. However, the size of the two certificates that are required to be sent to the EMV contactless card by the POS exceeds the maximum size of the APDU.

The solutions [96] and [97] for EMV mobile payments depend on the capability of the NFC mobile phone to connect to the cloud or the authentication server to authenticate the POS. Obviously, such capabilities is not available in EMV contactless cards. As a result, a reasonable mutual authentication protocol for the EMV contactless cards should depend on the EMV contactless cards themselves without the need for external servers. Another consideration when designing a mutual authentication for EMV contactless card is the size of the APDU commands and responses in addition to the contactless card transaction time as the EMV specifications allow up to 500 milliseconds to process an overall contactless card transaction [2]. However, it is worth mentioning that some valid EMV contactless card transactions could take a slightly longer time than 500 ms if the contactless card is not close enough to the POS [57].

6.3. Mutual Authentication Protocol

Herein, we describe a cost-effective dual-authentication proposal that relies on two-way challenge-response between the EMV contactless cards and POSs. We argue that the proposal does not require any change in the existing EMV payment protocol infrastructures as all required changes are done at the personalisation and transaction phases. The mutual authentication scheme is basically dependent on generating one-time random challenges generated for each transaction and exchanged based on a shared secret key of a symmetric cipher such as the Advance Encrypt Standard (AES) between the EMV card and the POS. This secret key is distributed among all the parties at the card's personalisation phase as details in the next subsection. At the transaction stage, the key is used to encrypt and decrypt challenges between the POSs and EMV cards as will explain next.

6.3.1. Keys Distribution & Personalisation Phases

The proposed solution requires that all EMV contactless cards and the POSs must have a shared secret key in order to process a mutual authentication between them and prevent reading the EMV contactless cards by unauthorised NFC enabled readers/smartphones. To minimise the changes in the EMV infrastructures, we suggest the use of existing methods that the EMV payment protocol is currently using to distribute the keys between different EMV components. As shown in Figure 6-1, the original processing of the EMV keys distribution is shown with the sold-bordered boxes while our additional steps are shown with the dash-bordered boxes.

In the original EMV keys distribution process, the CA sends its own public key (CA_{pk}) to both IB and AB so that CA_{pk} can be uploaded into the POSs. However, in the mutual authentication solution, we propose that:

- The CA distributes another key to both IB and AB in the same way that the CA_{pk} is being distributed. We refer to this key by CA Shared Secret (CA_{ss}).
- No changes are proposed to the IB public key certificate (IB_{pk}) as shown in Figure 6-1 where the IB sends IB_{pk} to the CA.
- The CA then signs the IB_{pk} with its own private key (CA_{sk}) to generate the IB public key certificate.
- The AB uploads the CA_{sk} in all the POSs.

In the original EMV card personalisation phase, the IB uploads the EMV card with a different type of data based on the type of the underlying EMV authentication methods (SDA, DDA or CDA). The SI, such as the PAN, expiry date and cardholder name, are uploaded by the IB into the EMV card. The signed version of the SI is also uploaded into the card along with the IB public key certificate as detailed in 4.1.3.

We propose one minor change to the original EMV personalisation phase, that is uploading the CA_{ss} into the EMV contactless card. As a result, both the POS and EMV cards are uploaded with the same CA_{ss} at the end of both keys distribution and personalisation phases.

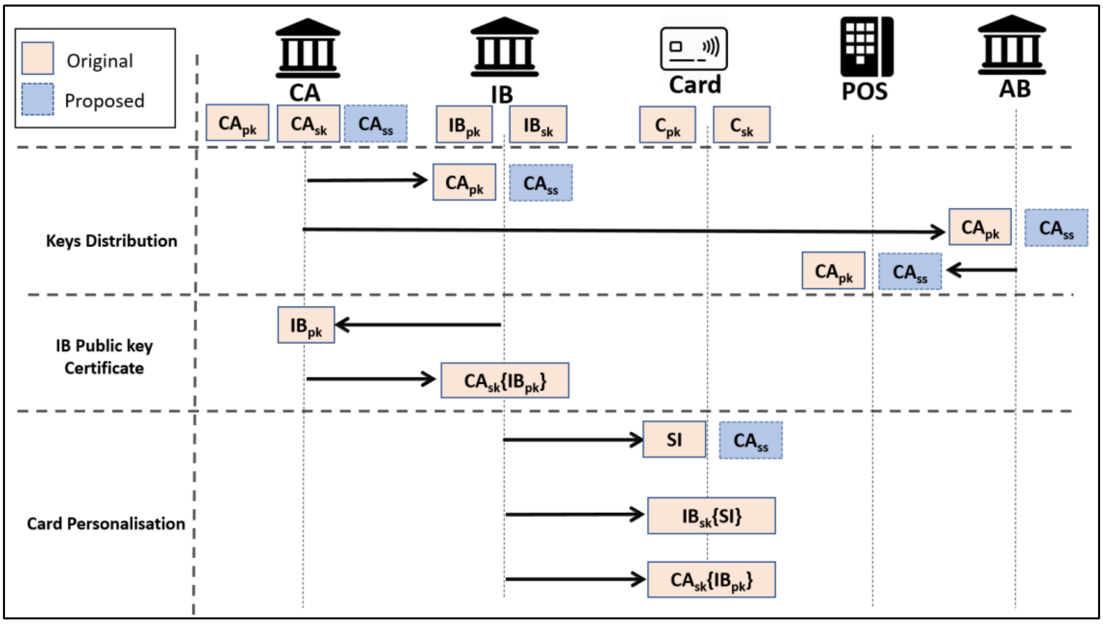


Figure 6-1: The Keys Distribution in the Proposed Scheme

6.3.2. Transaction Phase

The proposed mutual authentication relies on generating one-time random challenges for each transaction and employing the shared secret key, distributed to both EMV cards and the POSs as explained earlier. To stop skimming attacks, we propose that the EMV contactless card must authenticate the POS before the “GPO” response. The reason behind that is to prevent the EMV contactless card from revealing any sensitive data such as PAN, cardholder name and expiry date before the authentication process is completed. As such sensitive data is revealed in the original EMV transaction in both responses of “GPO” and “Read Record” Commands.

Figure 6-2 shows the main POS’s commands and EMV contactless card’s responses where the original EMV transaction steps are shown in black and the additional steps for the proposed mutual authentication are shown in red and can be summarised as follows.

- The first proposed change is shown in Figure 6-2 in step 5.3 in which the POS generates its own 8 bytes of Random Challenge (POS_RC) and sends it to the card along with the Application Identification (AID) as illustrated in Figure 6-2 in both steps 5.3 and 6.
- Once the card receives the POS_RC, it generates its own 8 bytes random challenge (Card_RC) as shown in step 7.1 in the same figure.

- The EMV contactless card calculates A as shown in Figure 6-2 in step 7.2 as following

  A= (POS_RC) XOR (Card_RC)

- The card then uses the symmetric key CA_{ss} (uploaded into the card at the personalisation phase by the IB as explained in the previous section) to encrypt A and send it back to the POS along with the "Select AID" response as shown in Figure 6-2 in both steps 7.3 and 8.
- When the POS receives the encrypted version of A, it uses the CA_{ss} (uploaded at the key distribution phase by the AB) to decrypt and obtain A as shown in step 9.1.
- Then the POS uses its own POS_RC (generated at the previous command) to obtain the Card challenge by doing the following

  Card_RC'= A XOR POS_RC

- The POS encrypts the Card_RC' by its own copy of CA_{ss} and sends it to the EMV card at the "GPO" command as shown in Figure 6-2 in both steps 9.4 and 10.
- When the EMV contactless card receives the encrypted version of Card_RC', it uses its own copy of CA_{ss} and decrypts it to obtain the Card_RC' as shown in step 11.1.
- Finally, the EMV card compares Card_RC and Card_RC'. If they are identical, the card authenticates the POS as a genuine one and it continues with the transaction as normal. Otherwise, the transaction should be aboard by the card as the POS fails to authenticate itself to the card as shown in steps 11.2.

The above steps show how the EMV contactless card authenticates the POS before releasing any information. The POS, on the other hand, authenticates and verifies the card based on the original EMV card authentication methods supported by the EMV payment protocol [99]. As a result, incorporating the proposed solution into any of the EMV card authentication methods (SDA, DDA and CDA) leads to establishing a mutual authentication between the POSs and EMV cards.

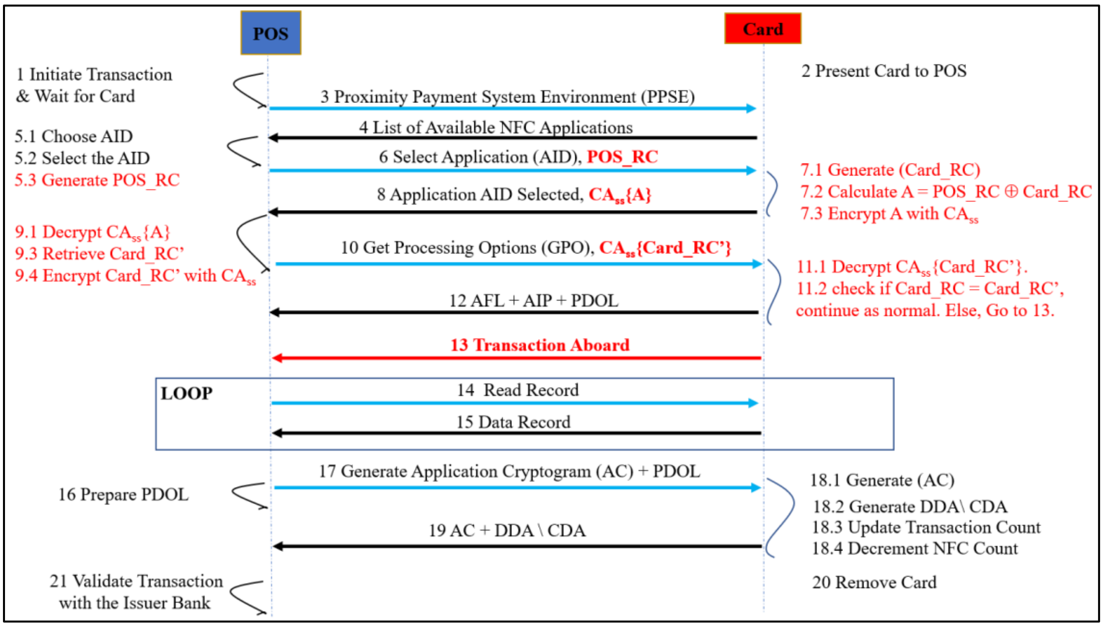

POS
Card
1 Initiate Transaction
& Wait for Card
2 Present Card to POS
3 Proximity Payment System Environment (PPSE)
4 List of Available NFC Applications
5.1 Choose AID
5.2 Select the AID
5.3 Generate POS_RC
6 Select Application (AID), POS_RC
7.1 Generate (Card_RC)
7.2 Calculate A = POS_RC ⊕ Card_RC
7.3 Encrypt A with CA_ss
8 Application AID Selected, CA_ss{A}
9.1 Decrypt CA_ss{A}
9.3 Retrieve Card_RC'
9.4 Encrypt Card_RC' with CA_ss
10 Get Processing Options (GPO), CA_ss{Card_RC'}
11.1 Decrypt CA_ss{Card_RC'}.
11.2 check if Card_RC = Card_RC',
continue as normal. Else, Go to 13.
12 AFL + AIP + PDOL
13 Transaction Aboard
LOOP
14 Read Record
15 Data Record
16 Prepare PDOL
17 Generate Application Cryptogram (AC) + PDOL
18.1 Generate (AC)
18.2 Generate DDA\ CDA
18.3 Update Transaction Count
18.4 Decrement NFC Count
19 AC + DDA \ CDA
20 Remove Card
21 Validate Transaction
with the Issuer Bank

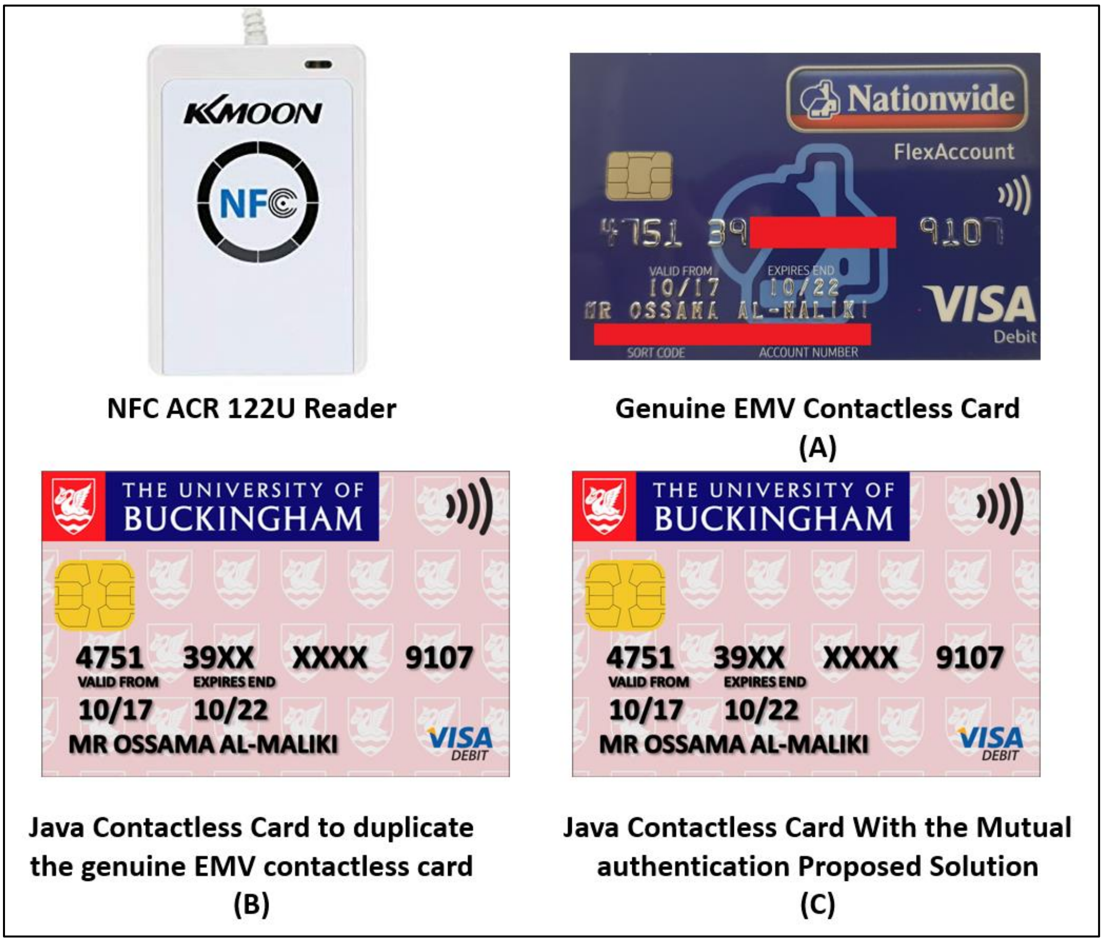
Figure 6-2: Transaction Phase in the Proposed Scheme

6.4. Implementation of The Mutual Authentication Protocol

This section demonstrates the practicability of the proposed mutual authentication solution to withstand sniffing attacks on EMV contactless cards launched by unauthorised NFC enabled readers/smartphones. The section explains the hardware and software used to implement the proposed solution. Then, it presents implementation and simulation results of the proposed scheme.

The same hardware and software which were detailed in Chapter 3 have been used in order to implement the mutual authentication protocol. We used two Java contactless cards, one to represent an original EMV contactless card and the other Java contactless card to represent the proposed mutual authentication protocol.

Moreover, a genuine EMV contactless card was used to duplicate its APDU responses to both Java contactless cards. This EMV contactless card was also used to calculate the times of the APDU commands and responses in order to compare its time with the time of the proposed solution as explain next. Figure 6-3 shows all the hardware used in our implementation.



NFC ACR 122U Reader

Genuine EMV Contactless Card
(A)

Java Contactless Card to duplicate the genuine EMV contactless card
(B)

Java Contactless Card With the Mutual authentication Proposed Solution
(C)

Figure 6-3: Hardware used in the Implementations & Simulations

6.4.1. Java Framework for Implementing the Mutual Authentication Protocol

As explained in the previous section, all the required changes for the mutual authentication proposed solution were done at both the "Select AID" and "GPO" APDU commands and responses. Since the EMV sensitive information is revealed by the EMV contactless cards at the response of "GPO" command, the proposal forces the POS to authenticate itself to the card before the card's response to the "GPO" command in order to withstand sniffing attacks. However, the proposed solution could modify easily in order to be implemented in other EMV contactless cards kernels such as the Fast DDA (FDDA).

To demonstrate and evaluate the proposal, we used the JCIDE to develop two Java applets to represent both an original EMV contactless card and the proposed protocol. Both applets were developed to respond to the first three APDU commands to serve the purpose of the implementation.

The first Java applet duplicates the original EMV contactless card shown in Figure 6-3 while the second applet was designed to simulate the proposed solution. The PyApduTool was used to connect both of the NFC ACR 122U reader and the three cards that are used in the implementation. The PyApduTool was also used to send the first three APDU commands namely "PPSE", "Select AID" and "GPO" to the three cards according to the EMV specifications [28]. All the three APDU responses are static i.e. the responses are the same for each transaction as the three commands and responses serve the purpose of handshaking between the POS and the EMV contactless card. Table 6-1 confirms the correctness of implementing the first applet as it precisely duplicates the same APDU responses of the original card.

For the second applet, the applet loads the Java contactless card with the CA_{ss} key at the personalisation phase. We developed the applet in a way that gives the researchers the ability to upload the Java card with AES keys of three different key sizes (128,192 and 256 bits) in order to simulate the solution. As stated earlier, no changes are required at the "PPSE" APDU command and response. Therefore, the second applet returns a similar response to the original one as shown in Table 6-1.

Table 6-1 shows all the three APDU commands ("PPSE", "Select AID" and "GPO") and their responses. The table shows the "APDU" commands of the original EMV and those of the proposed mutual authentication protocol. Two differences between the two APDU commands can be seen in Table 6-1 at the "Select AID" command, the POS sends its own 8 bytes of random number(POS_RC), and 2), at the "GPO" command, the POS sends 16 bytes of the encrypted version of Card_RC.

Moreover, Table 6-1 shows that there are three kinds of "APDU" responses. The first APDU responses belong to the original EMV contactless card while the second APDU responses belong to the first applet that duplicates the original EMV contactless card and hence both APDU responses are identical. The third APDU responses, on the other hand, belong to the second applet that represents the proposed protocol where one difference compared to the original APDU response. This difference can be seen when the card sends 16 bytes of the encrypted version of XORing POS_RC and Card_RC at the response of "Select AID" command.

Table 6-1: APDU commands & responses for all the Tested Contactless Cards

Name of APDU Command	**POS Original APDU Commands**	**POS Proposed APDU Commands**	**Card (A) APDU Responses**	**Card (B) APDU Responses**	**Card (C) APDU Responses**
PPSE	00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 00	00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 00	6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90 00	6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90 00	6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90 00
Select AID	00 A4 04 00 07 A0 00 00 00 03 10 10 00	00 A1 04 00 0F A0 00 00 00 03 10 10 **11 22 33 44 55 66 77 88** 00	6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 90 00	6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 90 00	6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 **06 7E 25 89 0C 24 83 BD 3E C0 F6 5D C4 9B EF EF** 90 00
GPO	80 A8 00 00 23 83 21 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	80 A8 00 00 33 83 21 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 **CA 97 8C 5D A9 95 0E 0E 05 B7 44 9E 56 83 01 D2** 00	77 4D 82 02 20 00 94 04 10 02 05 00 57 13 47 51 39 05 86 29 91 07 D2 21 02 21 99 20 00 00 00 00 0F 5F 34 01 00 9F 10 07 06 01 0A 03 A0 00 00 9F 26 08 92 FB E4 3F 5B D5 3D B6 9F 27 01 80 9F 36 02 00 1B 9F 6C 02 2E 00 9F 6E 04 20 70 00 00 90 00	77 4D 82 02 20 00 94 04 10 02 05 00 57 13 47 51 39 05 86 29 91 07 D2 21 02 21 99 20 00 00 00 00 0F 5F 34 01 00 9F 10 07 06 01 0A 03 A0 00 00 9F 26 08 92 FB E4 3F 5B D5 3D B6 9F 27 01 80 9F 36 02 00 1B 9F 6C 02 2E 00 9F 6E 04 20 70 00 00 90 00	77 4D 82 02 20 00 94 04 10 02 05 00 57 13 47 51 39 05 86 29 91 07 D2 21 02 21 99 20 00 00 00 00 0F 5F 34 01 00 9F 10 07 06 01 0A 03 A0 00 00 9F 26 08 9B 47 D9 91 EC 49 A7 12 9F 27 01 80 9F 36 02 00 1C 9F 6C 02 2E 00 9F 6E 04 20 70 00 00 90 00

6.5. Discussions Points Related to The Proposed Mutual Authentication Protocol

This section presents three key discussion points that address the following questions: 1) is the proposed solution efficient enough to meet the 500ms threshold?, 2) does the proposal satisfy the current EMV requirements related to the APDU size of commands and responses between the card and the POS? and 3) why the proposal is secure against sniffing attacks and what are the limitations?

6.5.1. Time & Computational Overhead

As mentioned earlier the transaction time is a vital point when assessing the proposed mutual authentication protocol for the EMV contactless card transactions. Such transactions should be processed very quickly. Therefore, any proposal that takes a long time to be processed must be rejected automatically.

We used PyApduTool to calculate the overall time to perform all the three APDU commands and responses. We repeated these three commands 10 times and reported the average time on the three different cards that are used in the implementation individually. Figure 6-4 shows the times for the genuine EMV contactless (Card A) and the first Java contactless card that was designed to duplicate the genuine card (Card B) while the second Java contactless card (card C) is designed to implement the proposed scheme.

As shown in Figure 6-4, the three cards (A, B and C) are spending almost the same time in order to process the first APDU command "PPSE" within almost 58 ms. This is an important observation not only to illustrate that the proposal does not change this specific APDU command and response but also to confirms that the reported execution times in this chapter are comparable to those in real-life i.e. the running of the same commands on Java and cards and an original bank card are comparable.

The figure also shows that both the original EMV contactless card (Card A) and the first Java contactless card (Card B) spend in average around 70 ms to process the second APDU response "Select AID". However, the proposed scheme (Card C) spends around 155 ms to process the same command i.e. an extra 85 ms as an overhead time of processing all the required steps in the proposed mutual authentication scheme.

Moreover, in the third APDU response of "GPO command", the proposed solution takes around 25 ms more than Card A and B. This extra time is required to process the steps of the proposed scheme. That makes the total extra time to process the proposed mutual authentication scheme is around 110 ms. It can be argued that the extra 110 ms is a reasonable overhead to be added to the EMV contactless card transactions as some of the genuine EMV contactless cards can take longer than 500 ms [57]. In fact, it has been shown that the POS usually tolerate contactless transactions even if they take more than 650 ms and the POS process and accept the contactless cards transactions within maximum time of 2500 ms [77].

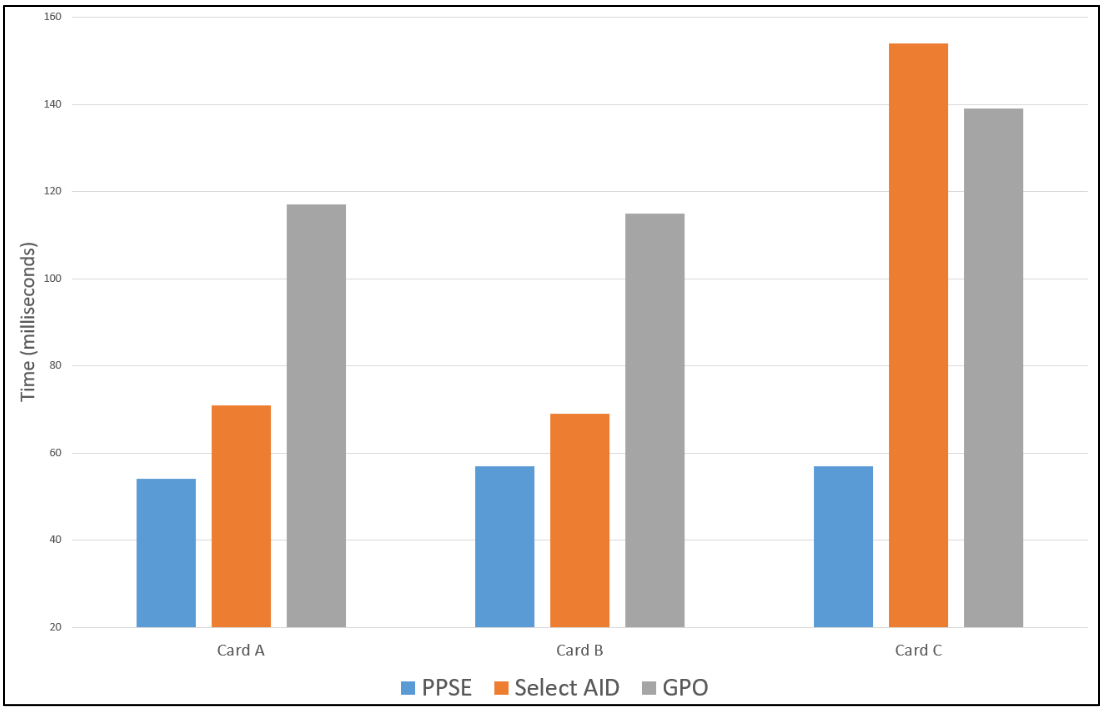


Figure 6-4: Timing Results for the Mutual Authentication scheme

6.5.2. APDU Size & Required Storage Size

The second consideration that the proposed mutual authentication protocol must consider to be integrated into the existing EMV infrastructure is the APDU size. The previous section showed that the whole process of the proposal requires just 40 bytes to be added to the APDU commands and responses of the original EMV contactless transactions. As shown in Table 6-1, there are 8 bytes added to the "Select AID" APDU command that represents the POS random number. Furthermore, there are 16 bytes added to the response of the same command "Select AID". The last 16 bytes are added on the “GPO” command. Therefore, these 40 bytes consider as a light added bytes to the original EMV APDU size.

Furthermore, the proposed mutual authentication scheme requires to store the CA_{ss} into the second Java contactless card (Card C) at the personalisation phase. This additional AES and key storage are about 3 KB. This added storage size is acceptable as the Java card used in the implementation had 72.8 KB of storage size. The way that this extra storage size is measured in the implementation is by comparing the size of the Java applet of Card B and Card C. As the applet Java size for Card B is 12 KB while it is 15 KB for Card C.

6.5.3. Security Analysis

The whole security of the proposed mutual authentication scheme rests on the security of CA_{ss} uploaded into the EMV card’s chip and the Secure Access Module (SAM) of the POS. The EMV chip itself considers as a secure environment to store sensitive information such as the card RSA private key and the Shared key of IB and the EMV chip. The SAM is currently also used to store all the CA certificates and several EMV applications. To the best of the authors’ knowledge, no successful skimming attacks on the EMV card’s chip or the SAM were reported in the literature.

If an attacker eavesdrops the communication between the POS and the EMV contactless card during the time of the contactless card transaction (practically hard but possible), can the attacker obtain the CA_{ss}?. In the proposed scheme, the CA_{ss} is not being sent by either POSs or the EMV contactless cards during the transaction. Therefore, even if the attacker eavesdrops to the communication, it is currently computationally hard to brute-force the CA_{ss} if it is a 256-bit key [100].

Attackers could also launch a Man-In-The-Middle (MITM) attack during the EMV contactless card transactions and modify the APDU commands and responses between the POSs and the EMV contactless cards. The POS_RC could be modified by an attacker which results to failure of the proposed scheme. However, this attack is possible in theory while in the real-life scenario, the attack is very difficult to achieve due to the limited distance (maximin of 10 cm) between the POSs and the EMV contactless cards during the contactless card transactions.

We appreciate that proposing a new solution to the existing EMV payment protocol affects almost 7.1 billion EMV based cards around the world [101]. Therefore, we suggest that the proposal can be deployed in newly issued EMV based cards while the existing EMV cards continue to work as usual using the current EMV payment protocol with the one-sided authentication protocol. To achieve that, we propose the use of one bit from the two bytes of the Application Interchange Profile (AIP) [28]. Using the bit as a flag to inform the POS of whether the EMV contactless card supports the proposal or not so the POS can respond accordingly.

6.6. Summary of the Chapter

One of the biggest threats on the EMV contactless cards is the skimming attack that could lead to other attacks such as CNP, replay and cloning attacks. Launching a skimming attack on contactless cards is quite easy to perform by fraudsters using off-the-shelf hardware and software. The main vulnerable point in the EMV payment specifications that led to a skimming attack is the one-way authentication. This specification forces the EMV card to authenticate itself to the POS while the reverse is not happening. Therefore, Any NFC reader or even NFC based smartphone can obtain most of the sensitive information of the EMV contactless card by launching a skimming attack.

This chapter presented a mutual authentication protocol between the EMV contactless cards and the POSs. The proposed protocol is cost-effective and easy to adopt by the EMV payment protocol with minor changes in the exiting EMV infrastructures. To prove that, we developed two Java applets on two Java contactless cards to simulate the original EMV contactless card and the proposed mutual authentication protocol. Moreover, the results showed that our proposed protocol is easy to fit with the EMV specifications and infrastructures.

The results also showed that the proposed protocol would not take a long time to be processed by the card and most of the additional data will not exceed the APDU maximum size. To conclude, the Mutual authentication protocol for the EMV contactless cards is must to be deployed by the EMV specification in order to prevent skimming attacks on such cards.

7. Chapter 7: Incorporating Gyroscope Sensor into EMV Contactless Cards

The PAN tokenisation, presented in Chapter 5, and the mutual authentication protocol, presented in Chapter 6, are sufficient to stop CNP attack and prevent the EMV contactless cards from being read by unauthorised NFC enabled readers/smartphones. However, attacks such as relay and stolen/lost attacks are still possible. Therefore, this chapter proposes a new solution to withstand the skimming and relay attacks on EMV contactless cards in such a way that the cards cannot be read by any POS or even any NFC enabled readers/smartphones without the cardholder's knowledge. The proposed solution depends on integrating a gyroscope sensor into the EMV contactless cards so that the cards can be activated and started engaging in a wireless transaction

The proposal can be utilised in two main scenarios:

- Activation: the card's movement is used to activate the card.
- Verification: the card's movement is used to authenticate the cardholder in addition to activating the card.

The simulation results show that the use of a simple movement works perfectly to activate the contactless cards and prevent skimming and relay attacks. However, more complex movement should be used to achieve acceptable level of accuracy when verifying the cardholder.

The chapter argues that the proposal can be easily integrated into the existing EMV infrastructure, as the only required change is the card's hardware and software by using a passive microcontroller with a built-in gyroscope sensor. The chapter discusses the challenges associated with transactions' time, range limitation, and the general usability of the proposed approach.

This chapter starts by explaining the proposed solution in Section 7.1. Then, Section 7.2 presents the implementations setup for the proposal. Next, Section 7.3 details the results of the implementations and discussion of the proposed solution. Finally, Section 7.4 concludes the chapter.

7.1. The Proposed Gesture Recognition Approach

The proposal aims to withstand both the skimming and the relay attacks on the EMV contactless cards by employing built-in gyroscope sensor. The sensor provides the EMV card during the contactless transaction with specific readings generated form a specific movement by the cardholder at the POS. If the cardholder performs the correct movement that can be recognized by the EMV contactless card, then the EMV card unlocks itself and starts engaging with the POS's RFID communications as normal. The contactless transaction is then carried out according to the existing EMV specifications as highlighted in Figure 7-1. Otherwise, the EMV contactless card rejects the RFID communication with the POS or any NFC enabled card readers/smartphones, and then sends an error message or instructs the POS to change the transaction interface to chip & PIN instead of a contactless transaction as shown in Figure 7-1. The black-dashed arrows in the figure represent the original EMV steps while the red-solid arrows represent the additional steps required to process our gesture recognition-based solution.

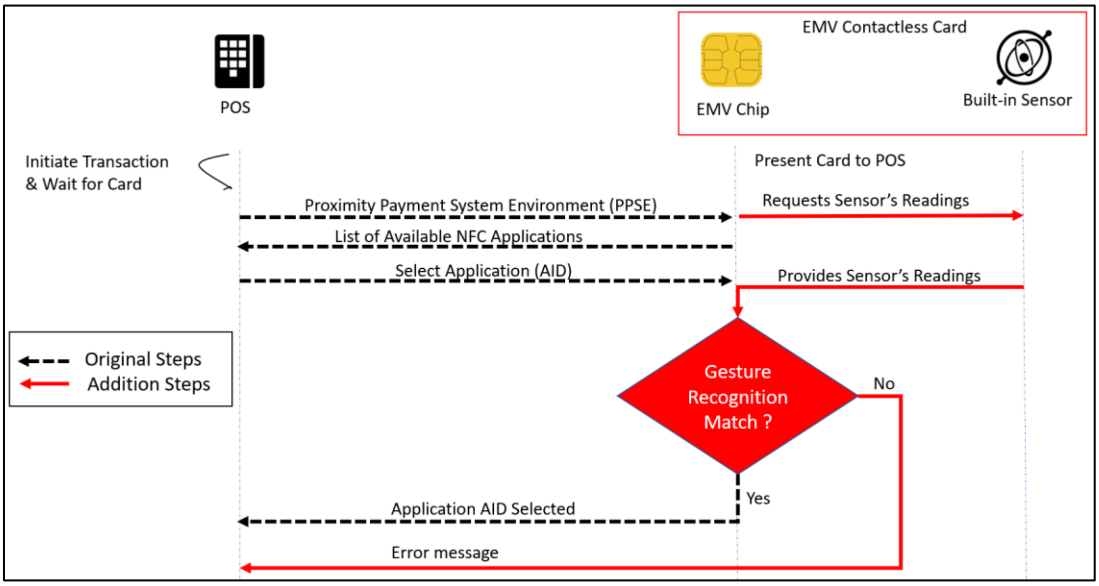

POS
EMV Contactless Card
EMV Chip
Built-in Sensor
Initiate Transaction & Wait for Card
Present Card to POS
Proximity Payment System Environment (PPSE)
Requests Sensor's Readings
List of Available NFC Applications
Select Application (AID)
Provides Sensor's Readings
Original Steps
Addition Steps
Gesture Recognition Match ?
No
Yes
Application AID Selected
Error message

Figure 7-1: Gesture Recognition Based Proposed Solution for the EMV Contactless Cards

As shown in Figure 7-1, the POS starts a contactless transaction by preparing all the information required for the transaction such as the amount, date, time and the currency. If the transaction amount can be processed as a contactless transaction (£45 or less), the POS activates its own RFID field by sending electromagnetic waves to power up the EMV contactless card’s chip. When the EMV contactless card enters the POS’s RFID range, it harvests the POS’s electromagnetic waves to power up its own chip. Then, the POS starts to send APDU commands and the EMV contactless card responses with a suitable response to each of these commands according to the EMV contactless card specifications [26].

The main goal of the gesture recognition-based solution is to allow the EMV contactless cards to engage with RFID wireless communication if and only if a specific movement is performed on the card by the genuine cardholder while presenting it at the POS. In other words, the EMV contactless cards must reject any kind of wireless communication if a particular movement is not performed. The main aim behind this proposal is to prevent the EMV contactless cards from being read by any unauthorised NFC enabled readers/smartphones so that skimming and relay attacks on such cards can be prevented.

To achieve that goal, we propose to incorporate a built-in gyroscope sensor into the EMV card to sense the movement. The card then compares the sensed movement against a signature movement stored on the card at the card-issuing stage (personalisation stage). Based on a similarity score set at the card-issuing stage, the card decides whether to allow or stop engaging with the RFID wireless communications. It is important to highlight that such checks must happen before releasing any sensitive information to prevent skimming attacks. Therefore, we decided to include the checks before the responding to the second APDU response (“Select AID” response as shown in Figure 7-1).

The first and the second APDU commands and responses are used as a handshake between the POS and EMV cards to agree on how to process the EMV transaction without sharing any of the sensitive information by the EMV card. Therefore, enabling the EMV card to decide whether to engage or not with any sort of wireless communication at these two first APDU commands and responses is critical to the proposed solution in order to prevent such cards from revealing their sensitive information to any unauthorised NFC enabled readers/smartphones.

7.1.1. Activation vs. Verification Scenarios

As explain earlier, the proposed solution can be deployed in two different scenarios, namely the activation and verification, based on the use of built-in gyroscope sensor to control the wireless connectivity. The activation scenario aims to control the EMV contactless card by preforming a specific movement by the cardholder in order to activate it and engage in RFID communication with the POS. On the other hand, the verification scenario attempt to verify the cardholder and activate the EMV contactless card by the use of a specific movement performed by the cardholder at the time of the EMV contactless card transaction.

7.1.2. Considerations & Challenges

In the proposed solution to build a gesture recognition by the use of built-in gyroscope sensor to control the wireless connectivity of the EMV contactless cards in order to prevent both skimming and relay attacks on such cards, we must design the proposed scheme to address and consider a number of points to ensure the effectiveness and the robustness of the proposed solution. These points are:

Time: our proposed solution should not take long to be processed. This is due to the nature of EMV contactless card transactions i.e. they should be processed within around 500 milliseconds according to the EMV specifications. Therefore, we should design our proposal in a way that requires minimum additional time.

Usability: The proposal has to be user-friendly by ensuring that the gyro’s gesture can be easily performed by the cardholder within the permitted physical range.

Cost-Effectiveness: the proposal should introduce minimal changes to the original EMV payment infrastructures.

Security Countermeasure: the proposal should provide a countermeasure for both skimming and relay attacks on the EMV contactless cards.

Distance Limitation: to ensure the effectiveness of the proposal, , we need to make sure that the movement of the activation gesture can be performed within the 10 cm of the POS.

False Rejection Rate (FRR): our proposal should make the False Rejection Rate (FRR) as low as possible. That means allowing the genuine cardholder to do EMV contactless card transactions from the first go with high probability.

False Acceptance Rate (FAR): our proposal should have a very low False Acceptance Rate (FAR). That means stopping the EMV card from engaging with any POS or NFC reader without the knowledge of the cardholder while they are doing a daily activity such as walking and sitting.

7.2. Implementation Requirements Setup for the Gyro Proposed Protocol

This section aims to illustrate the feasibility of the proposed solution and list the hardware and software used in order to implement the proposal. It also explains our dataset and the chosen movements. Moreover, the section shows the different algorithms which are used in the implementation.

7.2.1. Hardware & Software

Three hardware devices were used to implement the proposed scheme. The first hardware is the Classic Circuit Playground that is used to represent the built-in gyroscope sensor on the EMV contactless card to capture the card movement [102]. It is a microcontroller with a processor of 8 MHz and has different kinds of sensors such as light, temperature, sound and motion (gyroscope) sensors. In the implementations, we only used the motion sensor to provide us with the triple parameters X, Y and Z. The parameters give readings of the rotational and orientational motions. While the rest two hardware are the NFC ACR 122U reader and Java contactless card as detailed in Chapter 3.

The main objective behind using the NFC reader is to test that our prototype is working with a limited range distance of 10 cm (the RFID range of normal POSs) as all the movements testing are done within the range of this NFC reader. In addition, the main aim of using the Java card is to attach physically the microcontroller on it to test different motions within the range of the NFC reader. All the hardware devices are shown in Figure 7-2.

Arduino IDE was used to write simple codes for the microcontroller in order to collect different movements from the participants[103]. Moreover, we used PyApduTool software to check the connectivity between the NFC reader and the Java contactless card attached to the sensor.

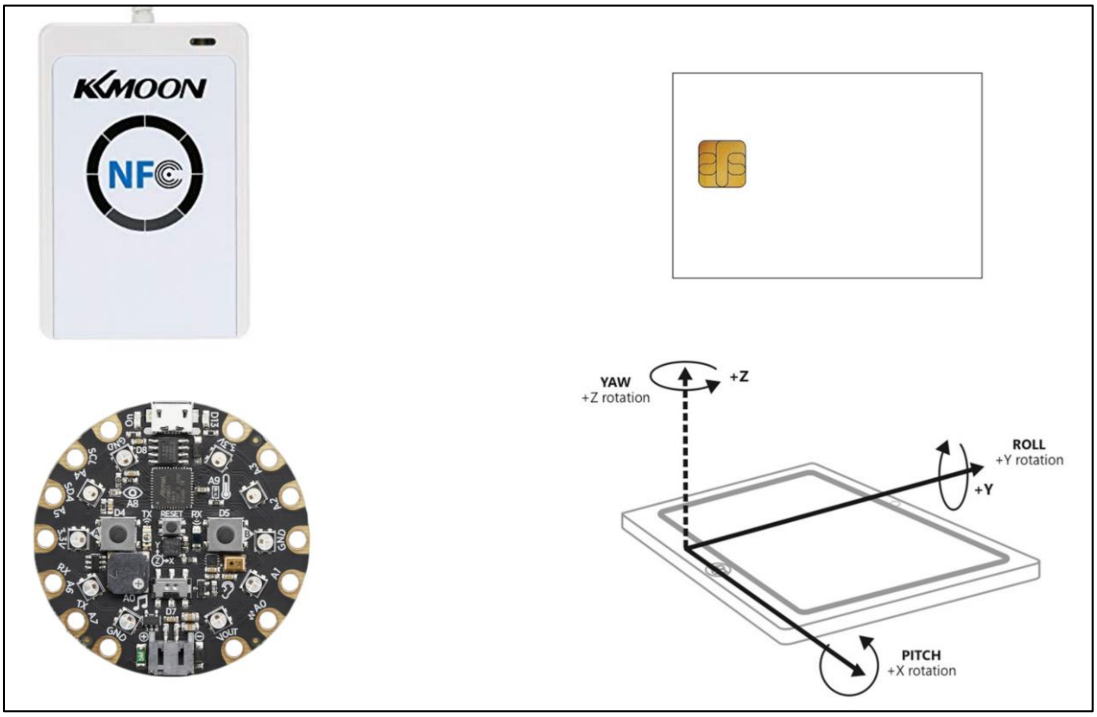


Figure 7-2: Hardware required to implement the Built-In Gyroscope Sensor Proposal

7.2.2. Dataset and Movements

To investigate the effectiveness of the proposed solution, we created our own dataset by collecting samples from ten different candidates, where each one was asked to perform five different movements (listed below). Each of the five moves was performed ten times by each candidate. i.e. 500 different samples representing the ten candidates performing the five movements were collected. We also collected samples that represent a number of everyday activities such as walking, standing and sitting moves, where each of the 10 candidates was asked to perform the movements.

The five different movements are:

1. Tap-Up-Down: taping the EMV contactless card on the POS.
2. Twist-Right: twisting the card slightly to the right-hand side on the POS.
3. Twist-Left: twisting the card slightly to the left-hand side on the POS.

4. Flip-Face-Up: Flipping the EMV contactless card from the face-up to the face-down. NB the face-up is the side of the card that the PAN is printed on.
5. Flip-Face-Down: flipping the card from the face-down to face-up. NB, the face down is the side of the card that the CVC is printed on.

The five movements listed above, were carefully selected because they are easy enough to be performed by a cardholder within the limited distance between the POS and the EMV contactless card during the transaction. Furthermore, the movements are unlikely to match everyday activities such as walking and sitting as illustrated by our results. This means that the cards do not get accidently activated in the pocket of the cardholder if an attacker places a card reader next to them.

7.2.3. Experimental Protocol

The Nearest Neighbour (1-NN) classifier was used to estimate the accuracy of the proposed scheme in which we used the Euclidean Distance (ED) and Dynamic Time Warping (DTW) to calculate the score similarity. The DTW algorithm is used to measure the score of similarity between two vectors of different sizes [104]. The DTW is used in many different applications such as speak recognition, signature recognition and shape matching [105]. The reason behind choosing the DTW to determine the similarity score is the variations in the speed of performing the movements across different candidates and different samples of the same candidate. The proposal's results are evaluated by reporting the Equal Error Rate (EER).

The Euclidean Distance (ED) algorithm [106] was used to work out the similarity between the movements and to test both the activation and verification scenarios as explained in the next subsections. The ED algorithm was applied on fixed-size feature vector, which means some vectors have zeros values as padding.

The ED was used as a benchmark against the DTW algorithm, where the ED was used on the original collected movements' samples (before pre-processing) as they were fixed size for all movements. However, the DTW was used on these movements after doing some sort of pre-processing on them. The pre-processing was done by taking out the less significant vectors (which do not form the specific movement) of each movement's sample in order to end up with only the most significant vectors which are forming each movement.

7.3. The Gyro Proposal Results & Discussions

The section shows the implementation and results of the activation and verification scenarios in addition to presenting a number of discussion points.

7.3.1. The Activation Scenario

Herein, we use the collected data to address the following two questions:

- Can the card get activated as a result of performing a random daily activity i.e. what is the probability of accident activation if the cardholder was walking, standing-up or sitting-down? It is unrealistic to assume that the card will be stationary in the pocket of the cardholder and , therefore, one need to assess the possibility of an attacker placing a card-reader next to the card while the cardholder is moving around.
- Can the card get activated if a different movement is performed? This provides an insight into the performance if a stricter approach to activating the card is followed i.e. what if we want the activation movement to be different not only from everyday activities but also from other movements. As mentioned earlier, the card movement is compared against a signature movement or "templet" before activation. Ideally, we would like the card to activate only if the intended movement is performed.

7.3.1.1. Robustness Against the Movements of Daily Activities

The goal of the first part of the activation scenario is to test how each movement reacts to everyday activities such as walking and sitting. As the ultimate aim of the proposed solution of the use of the built-in gyroscope sensor in the EMV contactless cards is to prevent attackers from being able to communicate with the EMV contactless cards without the genuine cardholder's knowledge. Hence, it is very important to test the five chosen movements against everyday activities. As the attackers could try to communicate with the victim's card while the victim is walking around or even sitting in a bus or a train. In order to test the movements against everyday activities, we collected 200 samples of two of the most common everyday activities namely walking and sitting to test against. Figure 7-3 shows the overall results of the testing the five moves against everyday activities.

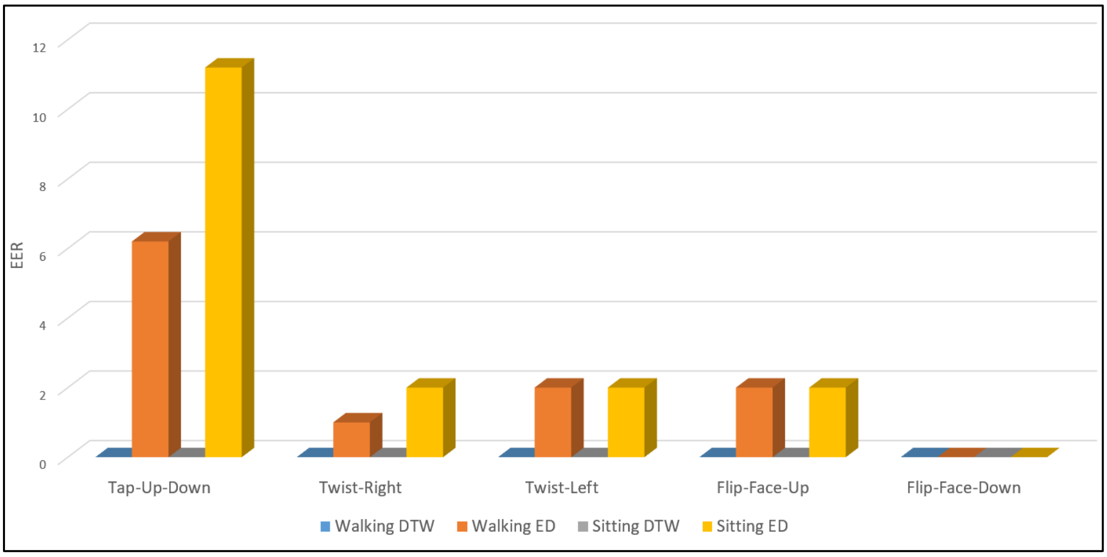
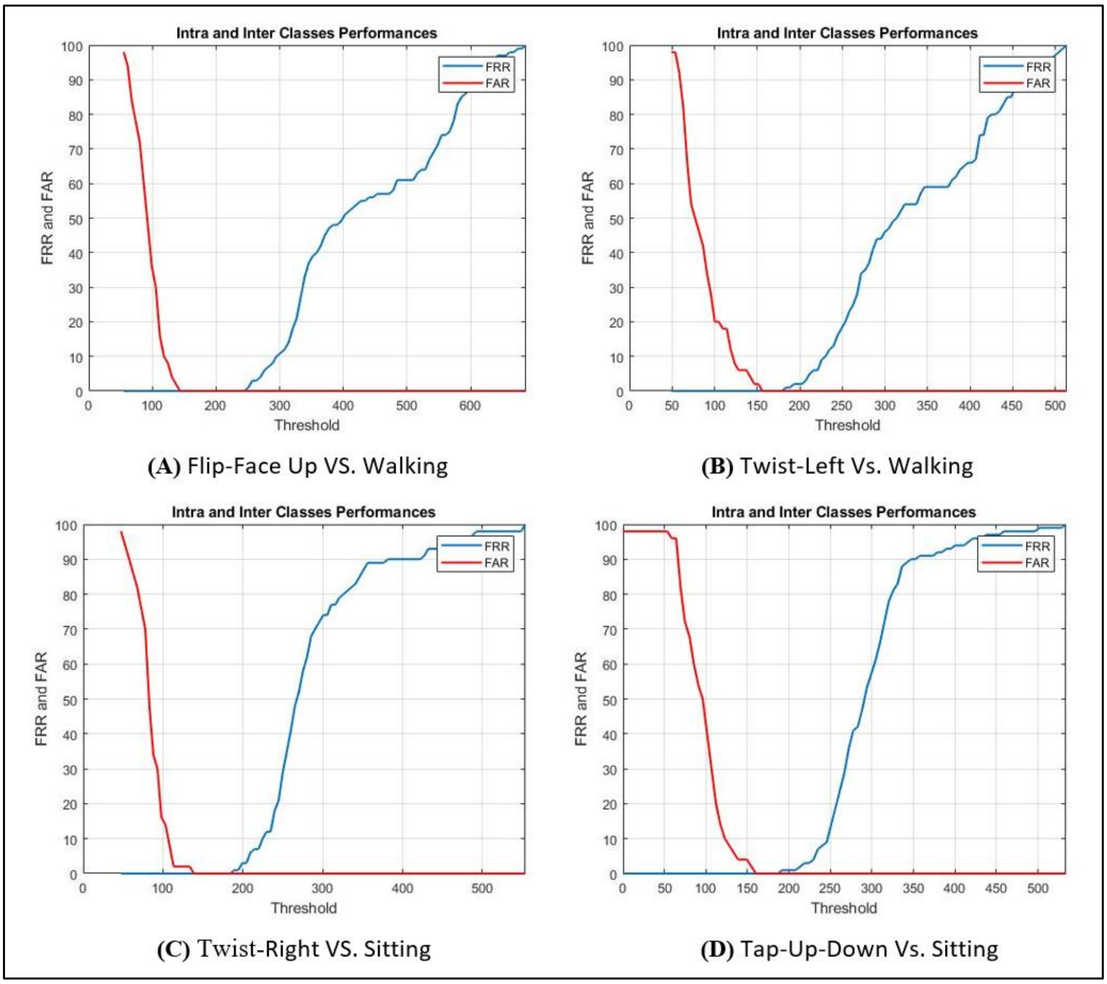


Figure 7-3: Testing each Movement against Everyday Activities Overall Results

As shown in Figure 7-3, all of the five chosen movements scored 0% EER using the DTW algorithm when they are compared against walking and sitting samples' movements. The results are a positive indication of the suitability of the proposed solution. However, the ED algorithm did not provide as good level of accuracy as the DTW algorithm. For example, the Tap-Up-Down movement has 6.2% and 11.2% EER using the ED algorithm when compared against walking and sitting samples respectively. While both Twist-Left and Flip-Face-Up movements scored 2% EER when they were compared against walking and sitting samples.

Figure 7-4 shows examples of the implementation results for testing the five movement against everyday activities, where A and B in Figure 7-4 show how both Flip-Face-Up and Twist-Left movements performances when they are compared with all the walking samples using the DTW algorithm to calculate the Intra and inter classes. Moreover, C and D in the same figure show how both Twist-Right and Tap-Up-Down movements performances when they are compared with all the sitting samples using the DTW algorithm to calculate the Intra and inter classes.



(A) Flip-Face Up VS. Walking

(B) Twist-Left Vs. Walking

(C) Twist-Right VS. Sitting

(D) Tap-Up-Down Vs. Sitting

Figure 7-4: Results of four Movements Against both Walking & Sitting Movements' Samples

As shown in the top figure that all the movements have scored 0% EER when these movements are compared against “walking” and “sitting” movements’ samples. That means the attackers will fail each time they try to activate the EMV contactless card while the genuine cardholder is sitting or walking around. The reason behind that is that our selected movements’ samples of all the ten users are totally unique when they are compared with the walking and sitting samples.

7.3.1.2. Robustness Against other Movements

The main goal of this part of the activation implementation is to find how each movement preforms compared with the rest movements. This part also aims to identify the best and the worst movement among all the chosen movements in terms of recognition accuracy in order to recommend it in the proposed scheme.

We designed the implementation where we tested each of the samples of each movement against samples of all other movements. For instance, we tested all the samples of the Tap-Up-Down movement from all the 10 users against all the samples of the other four movements of all users. Figure 7-5 shows all the overall results of the simulations.

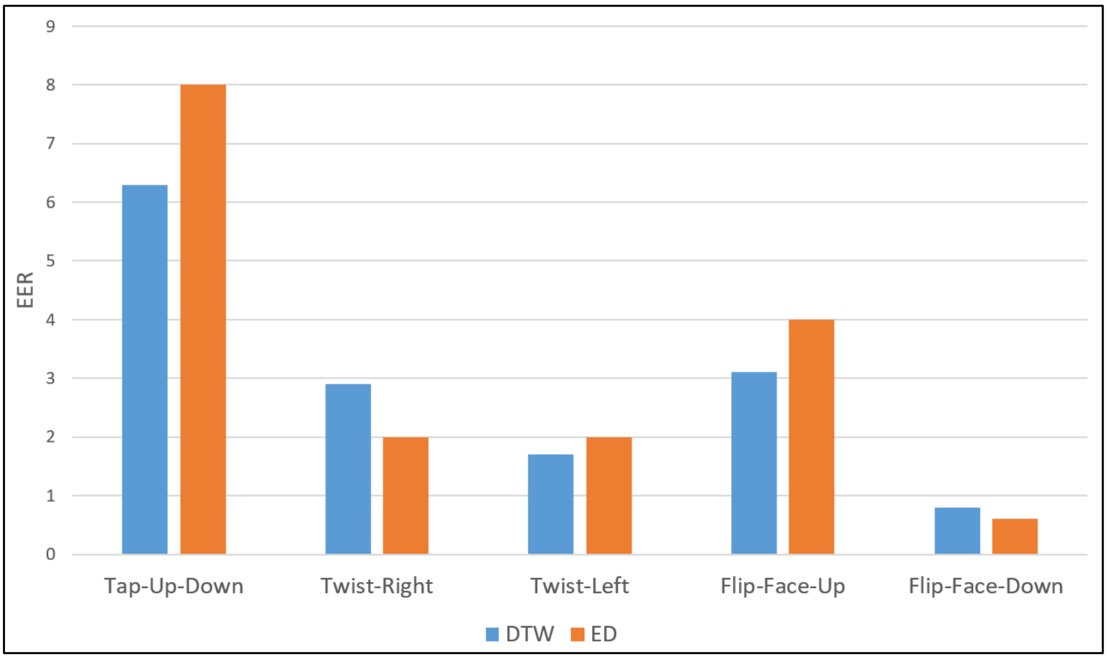
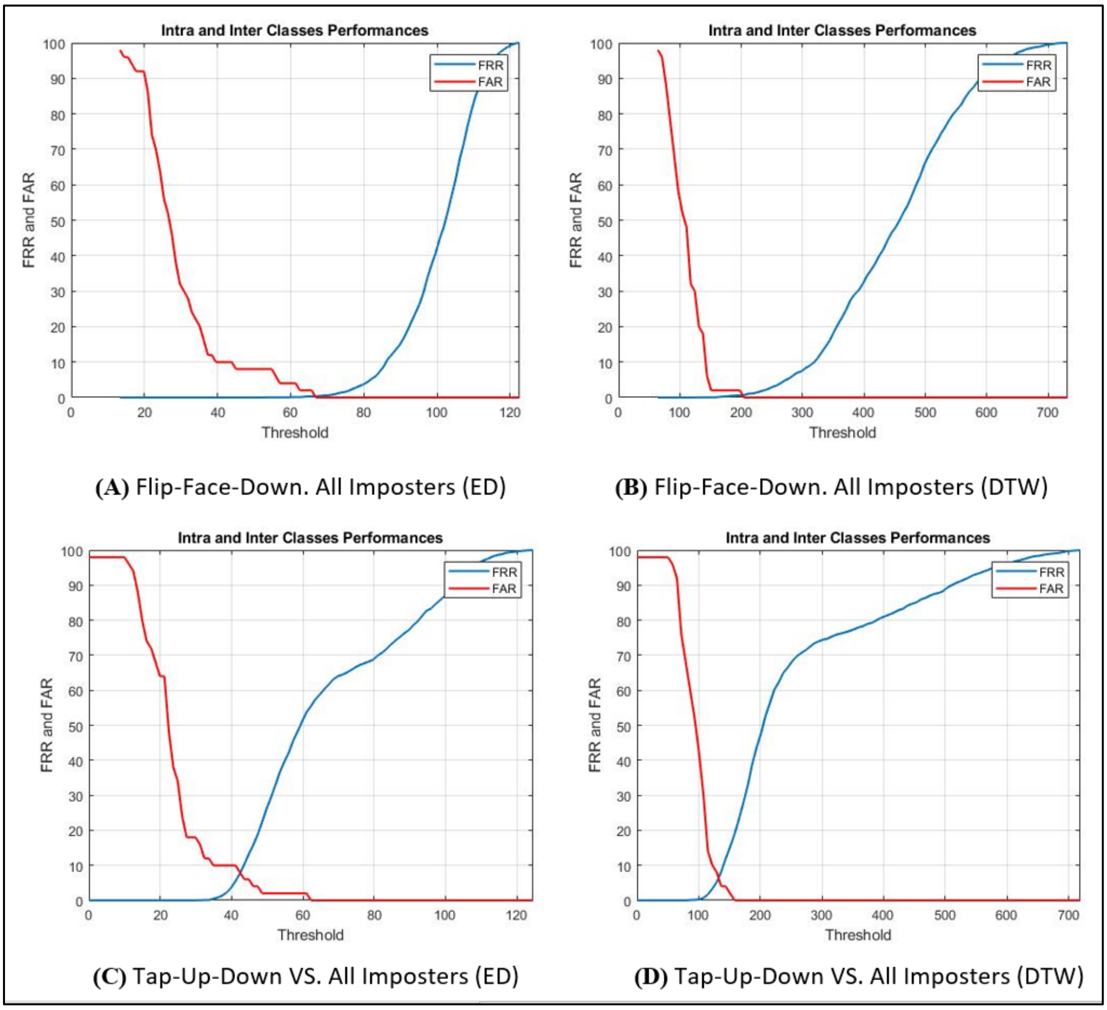


Figure 7-5: Overall Results for Testing each Movement against all Movements

Figure 7-5 shows that the Flip-Face-Down movement scored the lowest EER among the other movements in this this scenario scoring 0.8% and 0.6% of EER using DTW and ED algorithms respectively. However, the Tap-Up-Down movement scored the highest EER among the other movements as shown in the same figure.



(A) Flip-Face-Down. All Imposters (ED)

(B) Flip-Face-Down. All Imposters (DTW)

(C) Tap-Up-Down VS. All Imposters (ED)

(D) Tap-Up-Down VS. All Imposters (DTW)

Figure 7-6: Testing Results of both Flip-Face-Down & Tap-Up-Down Against all Imposters

Figure 7-6 shows the same results as above but with the details of the trade-off between FARs and FRRs. The main reason behind the above results might be that the simplest movement among all the other movements (Tap-Up-Down) has the worst performance. The results show that the more complex the movement the better the performance (lower EER) and the vice versa. Therefore, Flip-Face-Down has scored the best results in this scenario because it is the most complex movement among the rest while the Tap-Up-Down scored the worst in the same scenario due to its simplicity compared to other movements.

After that, we wanted to test each movement against the rest movements individually to have a better understanding of the results. To do so, we tested the samples of each movement against the samples of each movement separately. For example, we tested all the samples of the Tap-Up-Down movement from the 10 users against the samples of each of the other four movements separately as shown in Figure 7-7.

Figure 7-7 shows that the Flip-Face-Down movement scored the lowest EER among the other movements when it tested against all the four other movements. As shown in the same figure that this movement scored 0% EER by using both DTW and ED, when tested against the first three movements (Tap-Up-Down, Twist-Right and Twist-Left). However, the same movement scored higher of 2% EER using the DTW algorithm and 1.2% EER using the ED algorithm when it tested against the Flip-Face-Up movement. The reason behind that is both movements (Flip-Face-Down and Flip-Face-Up) are slightly similar when they are performed by the user in comparison with the rest of the movements.

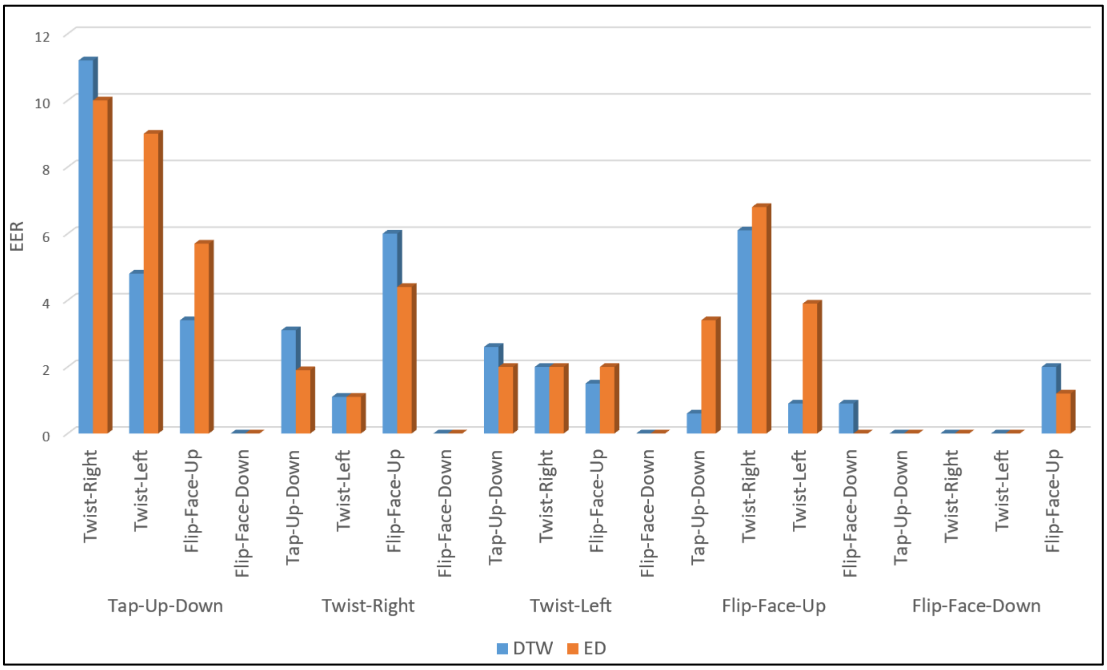
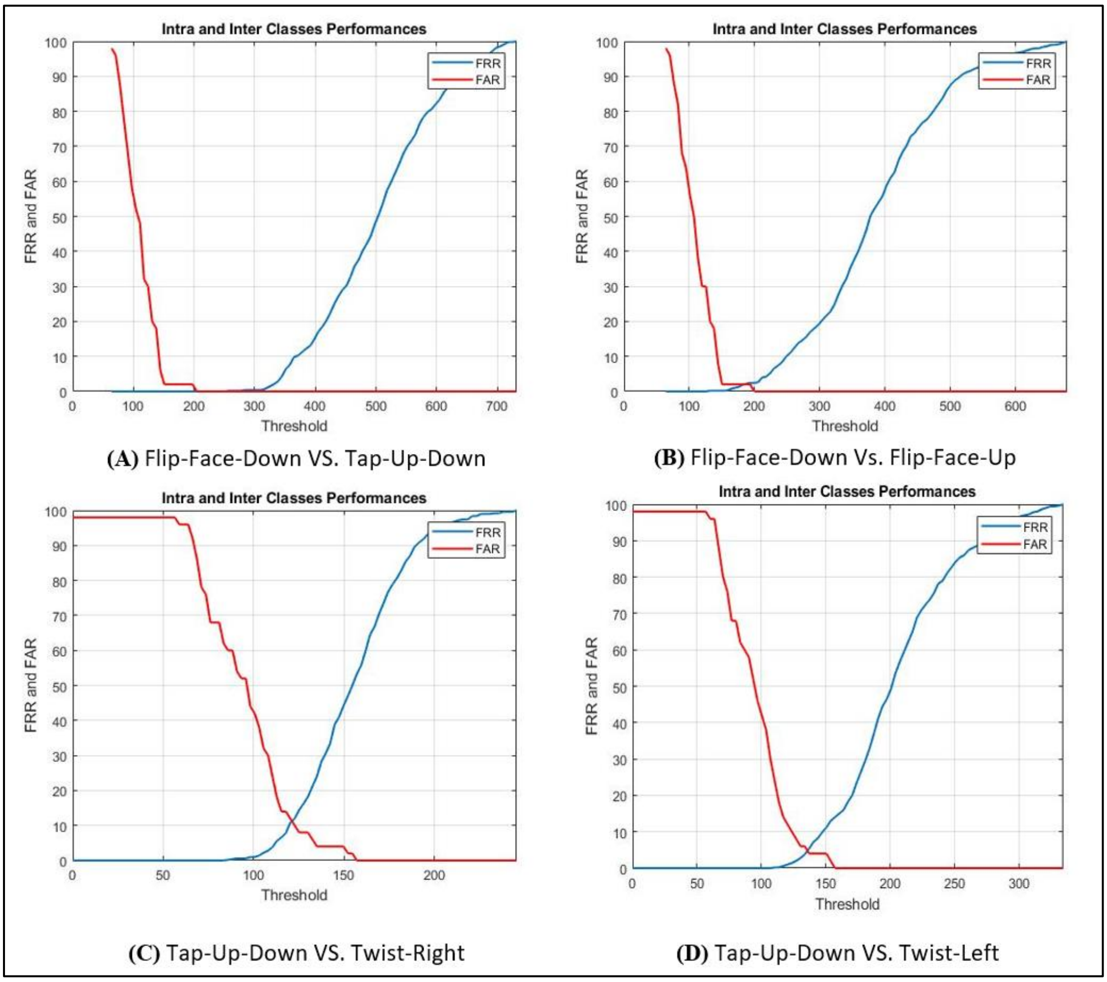


Figure 7-7: Overall Results for the Activation Scenario to Test the Movements Performance with each other

The same figure shows that the first movement (Tap-Up-Down) scored the highest EER (worst performance) among the other movements. This is due to simplicity of this movement in comprise with the other movements. However, when testing the Tap-Up-Down against the Flip-Face-Down, it scored 0% EER using both the ED and DTW due to how different the feature vectors of the collected samples for each of these two movements were.

The results in Figure 7-7 confirm the results in Figure 7-5 where the Flip-Face-Down movement can be considered as the best movement among the other four movements to be used for activating the EMV contactless cards. The Flip-Face-Down movement scored 0% EER against the everyday activities as shown in Figure 7-3 and has the lowest EER when tested against all the other movements as shown in Figure 7-5 and Figure 7-7. Therefore, we can certainly recommend the Flip-Face-Down to be used in the proposal.



(A) Flip-Face-Down VS. Tap-Up-Down

(B) Flip-Face-Down Vs. Flip-Face-Up

(C) Tap-Up-Down VS. Twist-Right

(D) Tap-Up-Down VS. Twist-Left

Figure 7-8: Example of Testing specific Movement against other Movement

Figure 7-8 shows details of the trade-off between FARs and FRRs of testing the movements against each other, where both of A and B in the figure show how the Flip-Face-Down movement performances when compared with both Tap-Up-Down and Flip-Face-Up using the DTW algorithm. Figure 7-8 C and D show how the Tap-Up-Down movement performs when compared with both of Twist-Right and Twist-Left movements.

7.3.2. The Verification Scenario

The second main scenario of the implementation is the verification scenario where we aim to check whether the movements could be used to verify cardholder i.e. can the proposal be used as a multifactor authentication for the contactless card transactions ? This could potentially give the cardholders more control over their EMV contactless cards as such cards do not have any kind of cardholder verification.

In order to test the verification scenario, this section evaluates the suitability of the five chosen movements to be used for verifying the users. To do so, we chose the strictest approach by designing the verification implementation in such a way that each movement of every user is tested against all the samples of the same movement of the other users. For example, to test the verification result of the Tap-Up-Down movement of user1, we compare the movement's samples of the Tap-Up-Down of user1 against the movement's sample of the Tap-Up-Down of the other nine users. Figure 7-9 shows the overall verification results of all the five chosen movements for the 10 users.

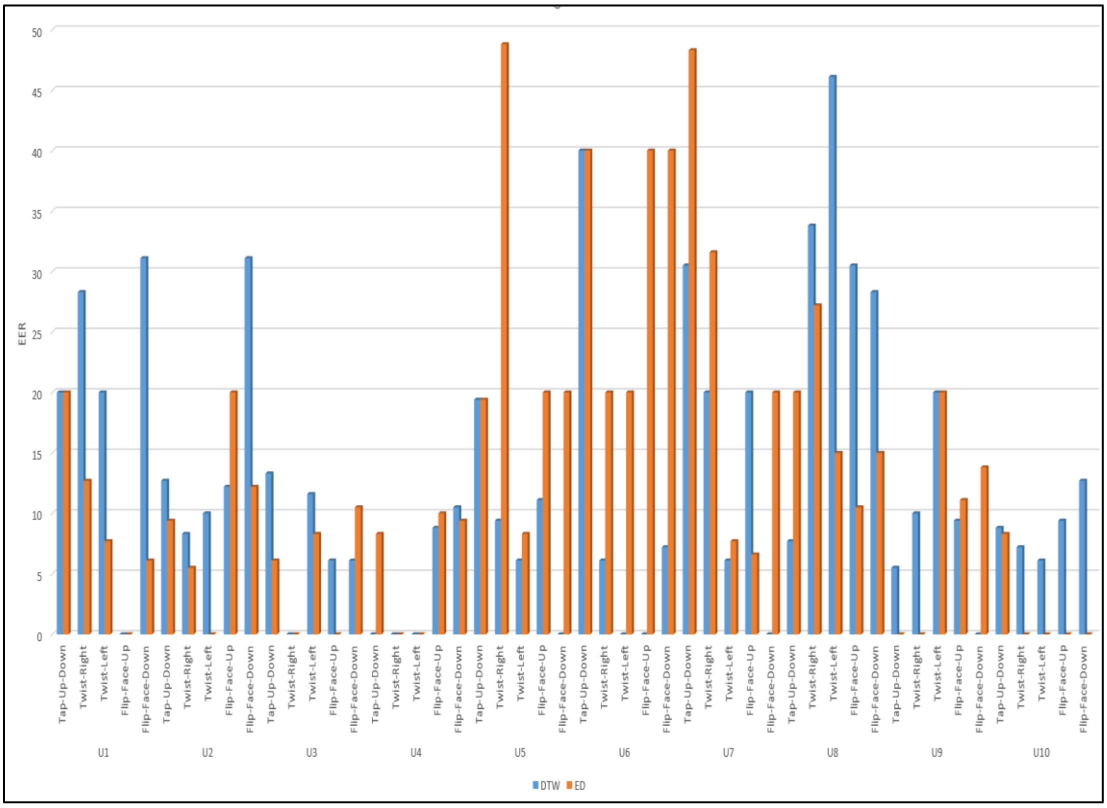
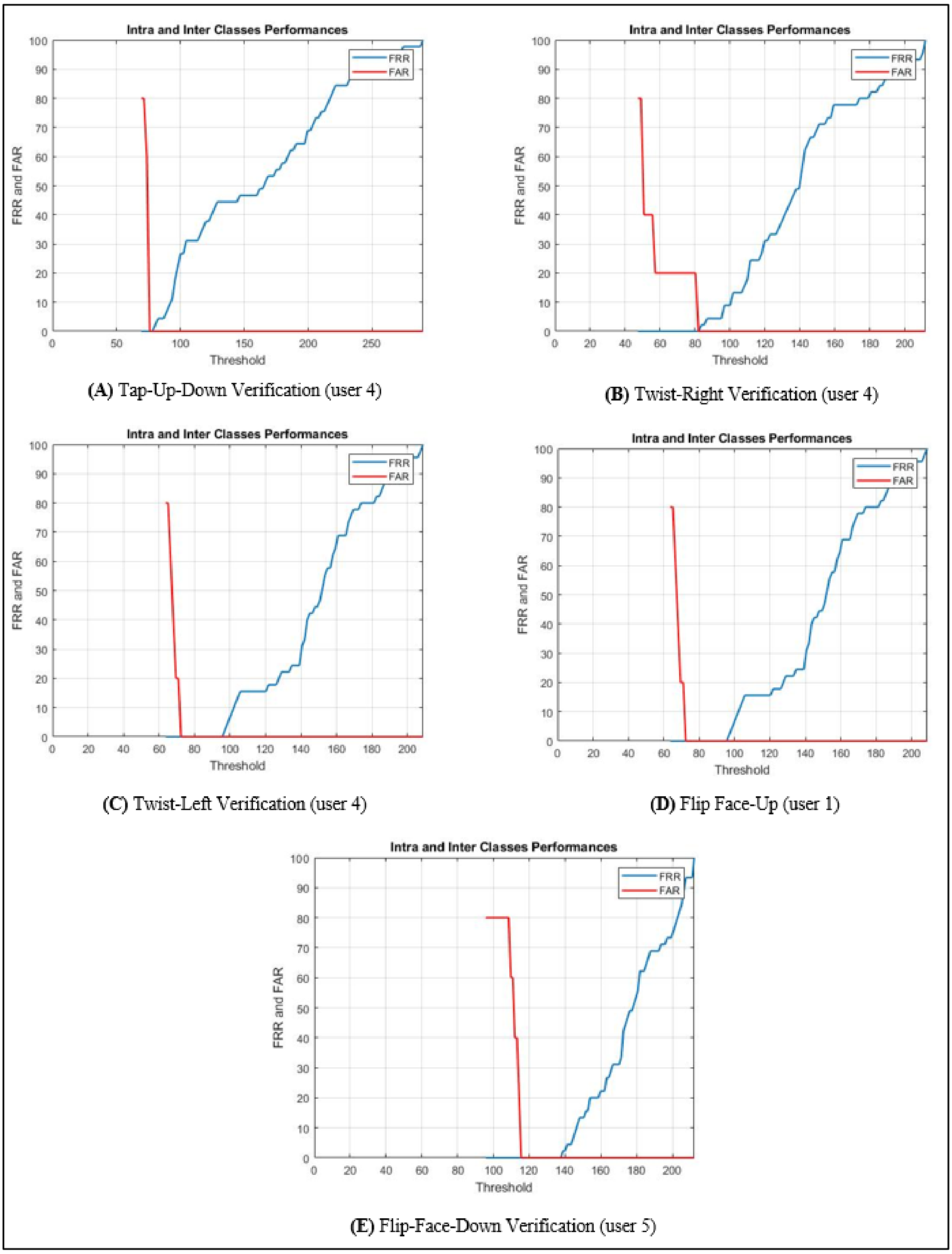


Figure 7-9: Verification Scenario Overall Results

As shown in Figure 7-9, the EER scores vary among individuals and also vary across different movement of the same individual. The figure shows that some individuals are better than the others in producing some movements. For example, the first user produced the most accurate samples of the Flip-Face-UP with zero EER while they produced less accurate samples for all the rest four movements. This applies differently to all other nine users as shown in Figure 7-9.



(A) Tap-Up-Down Verification (user 4)

(B) Twist-Right Verification (user 4)

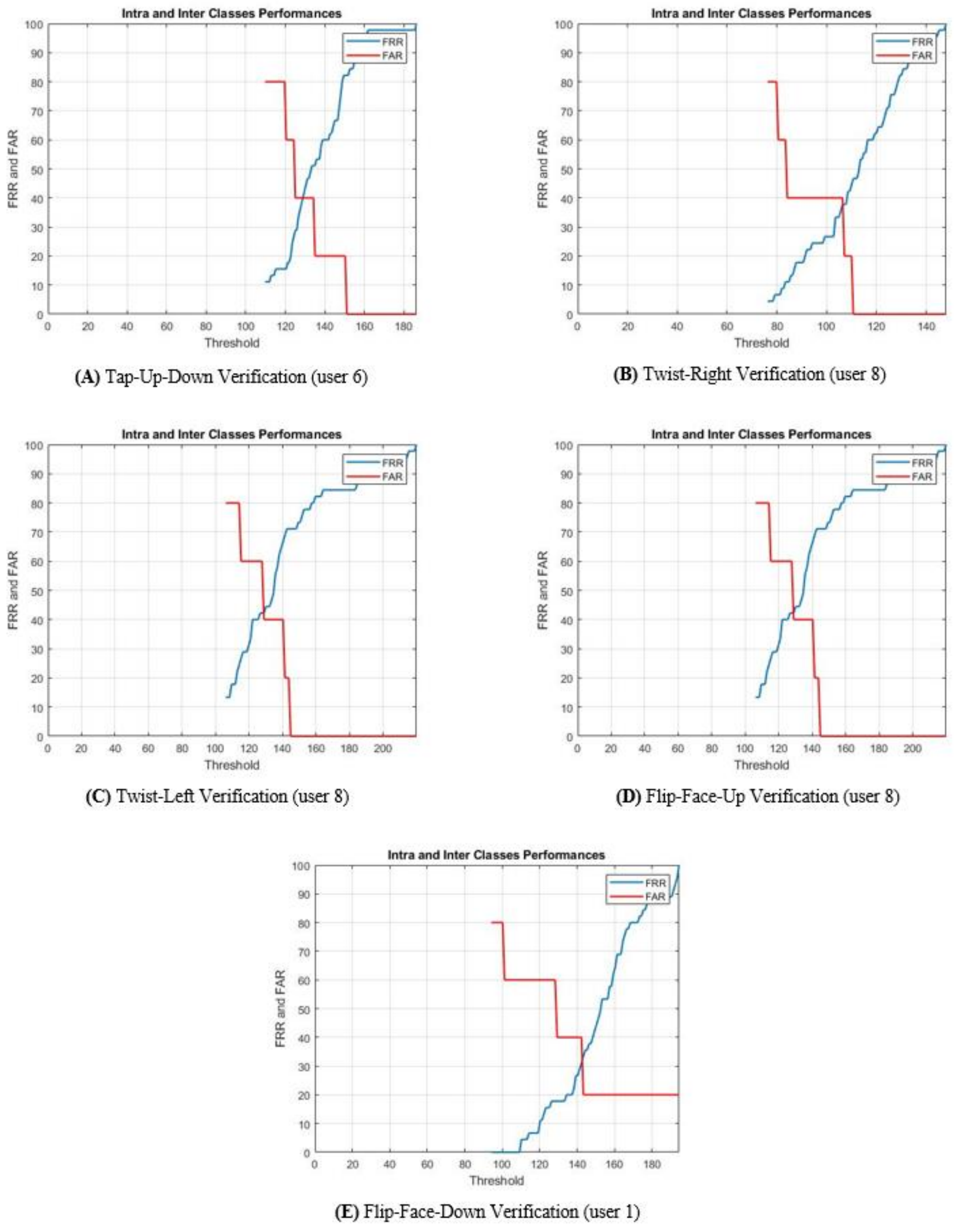
(C) Twist-Left Verification (user 4)

(D) Flip Face-Up (user 1)

(E) Flip-Face-Down Verification (user 5)

Figure 7-10: The Most Accurate Verification Results

Figure 7-10 shows the trade-off between FARs and FRRs for the users who produced the most accurate samples for all the five chosen movements while Figure 7-11 shows the same details for the users who produced the lowest accurate sample for all the five chosen movements.



(A) Tap-Up-Down Verification (user 6)

(B) Twist-Right Verification (user 8)

(C) Twist-Left Verification (user 8)

(D) Flip-Face-Up Verification (user 8)

(E) Flip-Face-Down Verification (user 1)

Figure 7-11: The Lowest Accurate Verification Results

As shown in both of Figure 7-10 (A) and Figure 7-11 (A) that the highest and lowest scores of EER for the Tap-Up-Down movement where user 4 scored the lowest EER among the other users while user 6 scored the highest EER. This means that user 1 produced unique Tap-Up-Down samples that could successfully be used to verify them from the other users with 0% EER. However, user 6 did not produce the same level of quality and had a score of 40% EER.

We can argue, based on the verification results, that the five chosen movements might not be complex enough to verify the users although some users produced unique samples for specific movement that could be used to verify them among the rest 9 users. Therefore, we do believe that using more complex movements such as a movement that is formed from a combination of simple movements or any sort of signature movement could deliver more accurate results.

7.3.3. Gyro Proposed Protocol-Discussion Points

The activation results presented in the previous section showed that the Flip-Face-Down movement produced the highest accuracy results compared with the other four movements. However, the Tap-Up-Down movement produced the lowest accuracy results in comparison with the other chosen movements. This due to the complexity of the Flip-Face-Down movement while the Tap-Up-Down movement considers as the simplest movement among all the chosen movements. Therefore, we do believe when the movement is more complex, the activation results are more accurate.

Moreover, all of the five movements produced low EER when they were tested against both of the two most popular everyday activities, namely walking and sitting. We particularly recommend the use of the Flip-Face-Down movement (achieved 0% ERR) in order to activate the EMV contactless card and prevent both skimming and relay attacks on such cards.

One of the interesting observations that we noticed from the verification results that the users vary in producing quality samples as shown in Figure 7-9. For instance, the fourth user produced three out of five of the best samples that scored the lowest EER in three different movements namely Tap-Up-Down, Twist-Right and Twist-Left as shown in Figure 7-10 (A,B and C) respectively. On the other hand, user number eight produced the least accurate samples among the other users by producing three out of five of the worst samples that scored the highest EER in three deferent movements namely Twist-Right, Twist-Left, and Flip-Face-Up as shown in Figure 7-11 (B,C and D) respectively.

In the verification scenario results, we believe that some movements might have the potential to provide a solid verification method to verify the user. The results could be improved significantly if the movements are more complex than the five tested movements. A signature movement could consist of multiple movements that the user needs to perform next to the POS could be used to verify the user in case of the EMV contactless card transaction. However, any complex movement increases the transaction time dramatically in comparison with a simple movement such as the five chosen movements that we tested on. Another reason that we thought about is the usability of performing a complex movement by the cardholder.

7.4. Summary of the Chapter

Using built-in sensor in the EMV contactless cards in order to prevent them from being skimmed or engaged in contactless transaction without the knowledge of the cardholder could be a promising solution. Our proposed solution does not require any change into the POS infrastructure or even the back-end payment system (Issuer Bank and Acquirer Bank) as most of the required modifications are done at the card side.

We created our own dataset in order to test the proposal which contains five different movements and two everyday activities movements (walking and sitting). Each movement was performed ten times by the each one of the ten users. Then, we tested two different implementation scenarios namely activation and verification scenarios our dataset.

The activation scenarios results showed that the activation implementation is working well with the chosen movements where each movement was tested against the rest of other movements and everyday activities. The results showed that the score of EER against walking and sitting movements samples was zero. That means the fraudsters will not be able to skim the EMV contactless card when the card is inside the cardholder wallet. Moreover, the Flip-Face-Down movement was the best movement among all the chosen movements which we recommend using to activate the EMV contactless cards.

On the other hand, the verification results were not as good as the activation results. The reason behind that was due to the simplicity of the chosen movements that we tested where we made sure to use simple movements in order to make our proposed method as easy as possible to the cardholders to do. We suggest that using more complex movements will enhance the verification results to successfully verify the cardholders and stop both skimming and relay attacks on the EMV contactless cards.

Proposing the use of gyro sensor into the EMV contactless cards in order to prevent skimming and relay attacks on such cards can introduce extra financial impact to the IBs due to the added cost of the gyro sensor. This extra require fees could make some IBs not willing to deploy the proposed protocol. Therefore, in the next chapter, we are proposing a new protocol which prevent both skimming and relay attacks on such card without the need of extra hardware to the EMV contactless cards. That's make the next proposal easier to be deployed by the IBs as no extra hardware is required and no required financial impact to the IBs.

8. Chapter 8: The Proposed Protocol of Using NFC Enabled Smartphone to Improve the Security of EMV Contactless Cards

In the previous chapter, we proposed the use of gyro sensor in order to prevent skimming and relay attacks on the EMV contactless cards by giving the cardholders more control on their cards. Nevertheless, the gyro proposal requires adding the sensor to the EMV contactless cards to implement the proposal.

This chapter gives cardholders the control over their cards without the need of adding any sort of hardware to these kinds of cards. To do so, we propose the use of the cardholders' NFC enabled smartphones in order to send an activation request to manage the NFC communication of the EMV contactless cards.

The chapter starts by showing the problem statement and several motivations. It then explains the proposed scheme in both activation phase and the transaction phase. Next, the chapter presents the implementations and results of the proposal scheme. Finally, it provides discussion points about the proposal and follows by the conclusion.

8.1. Problem Statement & Motivations

The no CVMs in the EMV contactless card transactions has led to various kinds of attacks as detailed in Section 2.2.1 due to the missing link between the genuine cardholders and their EMV contactless cards. This makes it possible for attackers to launch different kinds of attacks, such as skimming and relay attacks. Moreover, lost/stolen EMV contactless cards could be used by fraudsters months after they reported to the IBs by the genuine cardholders, as explained in Section 2.2.1.2.

Smartphones have become gradually part of people's life in various aspects of modern life. According to the Office for National Statistics in the UK, approximately 95% of the British people own at least one smartphone [107]. Moreover, the Scientia Mobil reported in its latest report in 2019 that the number of NFC enabled smartphone worldwide has grown by 19% since 2015. The percentage of the NFC enabled smartphones was 54% in the last quarter of 2015, while it increased to 73% in the last quarter of 2018 [108]. One of the reasons behind this growth of the numbers of the NFC enabled smartphones around the world is the smartphones' manufactures are started to enable all their new smartphones with NFC technology. For example, Apple's iPhone smartphones have NFC support since the iPhone 6, wherein 2015, almost 36% of iOS devices supported NFC while the 96% supported in 2018.

These above figures regarding the popularity of the smartphones and the NFC enabled smartphones motivated us to use the cardholders' NFC enabled smartphone in order to control the EMV contactless cards and prevent any possible attacks on such cards.

8.2. The Proposed Protocol

The chapter proposes that the cardholders need to activate their contactless cards before using them in contactless card transactions. By doing this, the proposed protocol prevents such cards from being skimmed by any unauthorised NFC enabled readers/smartphones without the knowledge of the cardholders. Also, the proposed protocol will prevent relay attack on such cards due to the added control of the genuine cardholders on their cards. Therefore, both of skimming and relay attacks will be stopped by this proposal. Moreover, the proposed protocol will not require any extra hardware to be added to the card's side which make it requires less cost comparing with the proposed protocol in Chapter 7.

The activation process is done by the use of the cardholders' NFC enabled smartphones. Most of the IBs offer their cardholders the ability to use the mobile banking applications to allow them to manage their account by transfer money between different accounts, pay people, check balance, change cash machine limit, view transaction history and statements, report of stolen/lost cards, and several more different features according to each IB [109][110].

Therefore, we suggest that the IBs add another feature to their mobile banking applications that allows cardholders to activate or deactivate their EMV contactless cards. The proposed scheme works as shown in Figure 8-1 and can be summarised as follows.

- First, the genuine cardholder needs to launch the IB mobile banking application as shown in step 1 in Figure 8-1, where the IB mobile applications verify the cardholder by a password, PIN, or a fingerprint. We propose that the application should turn on the NFC on the smartphone automatically in order to make it possible to communicate with the EMV contactless card later on.

- Second, the cardholder needs to launch the activation option in the IB mobile banking application in order to send an activation request to the EMV contactless card as shown in step 2 in Figure 8-1. Once done, the cardholder should place the NFC enabled smartphone next to the EMV contactless card.
- Next, when EMV contactless card enters the RFID filed of the NFC enabled smartphone, the smartphone starts sending the required APDU commands that aim to activate the EMV contactless card as shown in step 3.
- The EMV contactless card then receives the APDU commands and processes them according to the proposed scheme as shown in step 4. As in this point, the EMV contactless card verifies that the activation request is issued by its own genuine cardholder’s NFC smartphone, as explained in the next subsection. If the verification process is successful, the EMV contactless card confirms the activation process and updates its status and becomes ready for contactless card transactions. Otherwise, the activation request is denied, and an error message is sent back to the NFC enabled smartphone.
- At the time of contactless card transactions, the POS starts processing the contactless card transactions as usual as there is no change in the POS APDU commands. However, the only change in this proposed scheme is happening at the side of the EMV contactless card. The card checks its status, to decide to engage with the POS or not. If the status is active, the transactions carry as normal as shown in step 7 in Figure 8-1. Otherwise, an error message is sent by the EMV contactless card to the POS or a request of change the transaction interface to chip & PIN transaction instead of the contactless card transactions as shown in step 8.

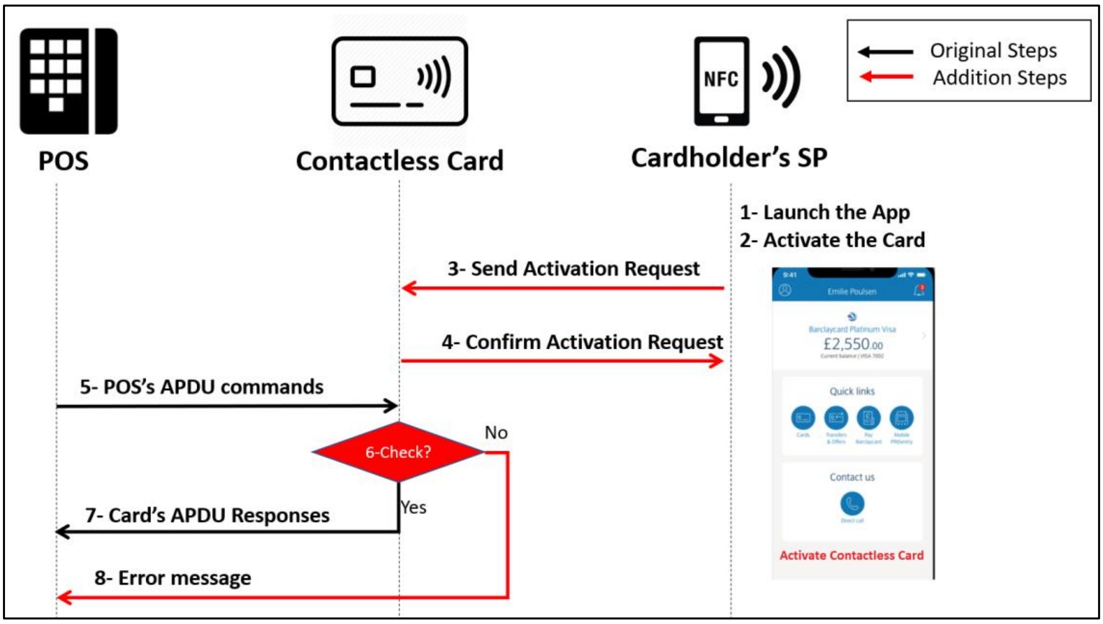


Figure 8-1: NFC Smartphone to Control the EMV Contactless Cards Proposed Scheme

8.2.1. Scripts APDU Commands to Update the EMV Cards

The proposal requires to update the NFC status of the EMV contactless cards by the use of the cardholders' NFC enabled smartphones. The updating process needs to be done by the IBs of the cards or any other authorised device such as the cardholder's smartphone through the mobile banking application. To do so, we propose the use of the EMV issuer scripts APDU commands in order to update the status of the EMV contactless cards in this proposed scheme.

The EMV issuer script commands allow the IBs to update and change EMV cards' parameters while they are still in the POS's field [111].The EMV payment protocol specifications supports several issuer scripts APDU commands that allow to update various EMV card's parameters which were fixed at the personalisation phase, block and unblock the application, block the card, reset the PIN try counter, change the PIN and many more [28]. For example, when cardholders need to change the PIN, they need to use an ATM to do so. Then, a script command is sent by the IB to the EMV card in order to update and change the old PIN with the new one.

Therefore, to minimize the necessary change in order to process the proposed scheme, we propose the use of two scripts issuer APDU commands, namely "Get Challenge" and "External Authenticate" in order to serve our aim in this proposal.

The “Get Challenge” APDU command is used to obtain 8 bytes of a random number from the EMV card, and these 8 bytes are valid only in the next issued APDU command. Moreover, the “External Authenticate” APDU command is used to request the EMV card to verify a cryptogram. Both the “Get Challenge” and “External Authenticate” APDU commands are coded in the EMV payment specifications, as shown in Table 8-1 [35].

To make sure that the scripts APDU commands are issued by the authentic IB or any other party that was authorised by the IB such as the mobile banking application, we propose the use of the EMV card Master Key (ICCMK) in order to prevent any attempt to use these scripts APDU commands by attackers or any unauthorised party. This key is a shared key between the IBs and the EMV cards.

Table 8-1: The Coded of both Get Challenge and External Authenticate APDU commands

Code	Get Challenge	External Authenticate
CLA	00	00
INS	84	82
P1	00	00
P2	00	00
LC	00	8-16 bytes
Data	None	8-16 bytes
LE	00	00

8.2.2. Activation Request Phases

In the proposed scheme, it is vital to make sure that only the genuine cardholder’s NFC enabled smartphone is the only NFC enabled device that could send the genuine activation request to the EMV contactless card. By doing this, we prevent any attempt to activate the EMV contactless cards by an unauthorised person such as attackers. Therefore, we propose that the activation request APDU commands and responses between the cardholder’s NFC enabled smartphone and the EMV contactless card are managed as follows:

- The genuine cardholders start by launching their IB mobile banking application by entering either password, PIN, or use a fingerprint. Then, the cardholder selects the activation request and then places his EMV contactless card next to the NFC enabled smartphone, as shown in both steps 1 and 2 in Figure 8-2.
- Next, the cardholder’s NFC enabled smartphone sends the first APDU command, namely “PPSE” in order to request all NFC applications that are supported by the EMV contactless card. Then, the EMV contactless card responds by sending all the NFC applications including the proposed activation NFC application.
- The NFC enabled smartphone chooses and selects the NFC application and sends the “Select AID” APDU command to the EMV contactless card. Then, the EMV contactless card confirms the AID selection, as shown in steps 5, 6, and 7 in the same figure.
- Once the NFC AID application is selected, the smartphone sends the “Get Challenge” APDU command to request the EMV contactless card to generate 8 bytes of random challenge. Then, the card generates its own 8 bytes of random challenge (Card_RC) and encrypts it with the IICMK and sends it back to the NFC smartphone in the response of the “Get Challenge” APDU response as shown in steps 8, 9 and 10 in Figure 8-2.
- When the NFC smartphone receives the APDU response of the “Get Challenge”, it generates its own 8 bytes of random challenge (SP_RC) and decrypts the ICCMK{Card_RC} to obtain the Card_RC. Then, it calculates A as following

$$A = SP\_RC \oplus Card\_RC$$

- Then, the smartphone sends the "External Authenticate" APDU command along with the activation request and both of the encrypted versions of SP_RC and A, as shown in both step 12 in Figure 8-2.
- Next, when the EMV contactless card receives the "External Authenticate" APDU command, it obtains both of the SP_RC and A by decrypting both the ICCMK{SP_RC}and ICCMK{A} using its own ICCMK. Then, it calculates B, where B is calculated as follows:

$$B = Card\_RC \oplus A$$

- Finally, the card checks whether B matches to SP_RC. If both are matched, then the EMV contactless card confirms the activation request and changes its NFC status (True) and sends a confirmation message, as shown in step 14. Else, an error message is sent back to the smartphone as a result of failure authentication of a genuine smartphone, as shown in step 15 in Figure 8-2.

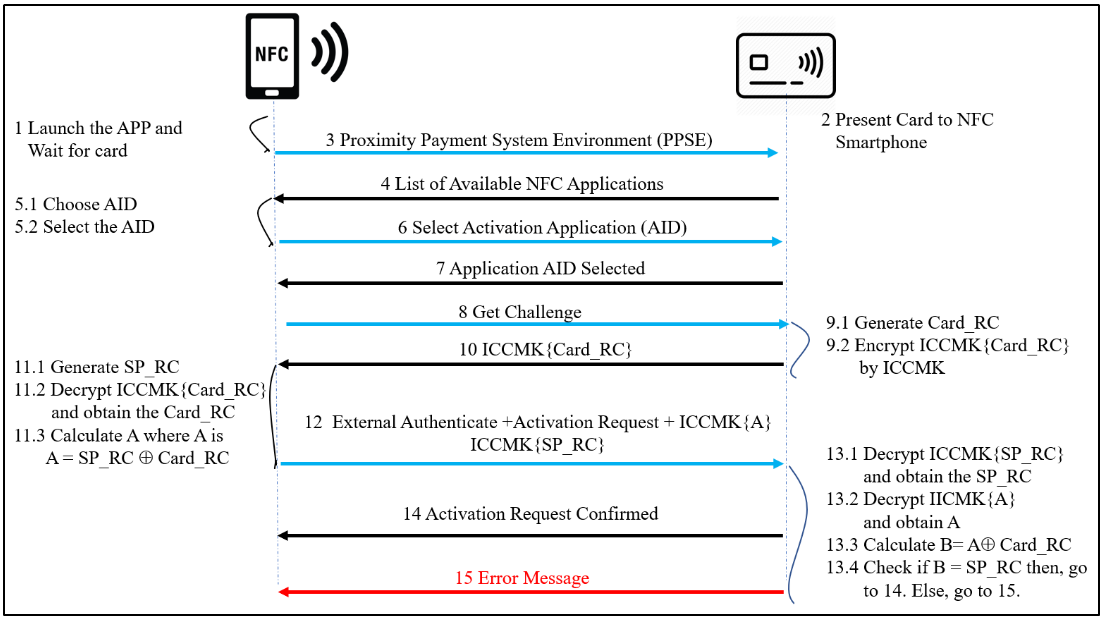

NFC
1 Launch the APP and Wait for card
2 Present Card to NFC Smartphone
3 Proximity Payment System Environment (PPSE)
4 List of Available NFC Applications
5.1 Choose AID
5.2 Select the AID
6 Select Activation Application (AID)
7 Application AID Selected
8 Get Challenge
9.1 Generate Card_RC
9.2 Encrypt ICCMK{Card_RC} by ICCMK
10 ICCMK{Card_RC}
11.1 Generate SP_RC
11.2 Decrypt ICCMK{Card_RC} and obtain the Card_RC
11.3 Calculate A where A is A = SP_RC ⊕ Card_RC
12 External Authenticate +Activation Request + ICCMK{A} ICCMK{SP_RC}
13.1 Decrypt ICCMK{SP_RC} and obtain the SP_RC
13.2 Decrypt IICMK{A} and obtain A
13.3 Calculate B= A⊕ Card_RC
13.4 Check if B = SP_RC then, go to 14. Else, go to 15.
14 Activation Request Confirmed
15 Error Message

Figure 8-2: The Proposed Scheme in the Activation Phase

8.2.3. The Proposed Scheme at the Transaction Phase

Figure 8-3 shows the proposed scheme in the EMV contactless card transaction phase. The figure shows that the proposed scheme does not change the original APDU commands of the POS in the whole EMV contactless card transactions. That is because all the required changes and processes are on the card's side, whether before the actual transaction, as explained in the earlier section or during the transaction, as shown in the red arrows in Figure 8-3.

In the first two “PPSE” and “Select AID” APDU commands, there is not require change in the original APDU responses of the EMV contactless card. Nevertheless, in the next three APDU commands namely “GPO”, “Read Record” and “Generate AC”, the EMV contactless card needs to check whether it was activated or not before responding to these APDU commands as shown Figure 8-3. If the card was activated by the cardholder’s NFC enabled smartphone, then the card continues as normal by responding the usual APDU responses for each of these three APDU commands. Else, the EMV contactless card responds with an error message, as shown in the same figure.

The reason behind choosing to perform the first check before responding to the “GPO” APDU command is because in the “GPO” APDU response, the EMV contactless card exposes several sensitive information such as the PAN, AFL, AIP, and PDOL.

Moreover, we intended to perform the EMV contactless card checks regarding of the card being active or not in all the three APDU commands, namely “GPO”, “Read Record”, and “Generate AC”. The three checks aim to prevent the attackers from changing the original sequence of the APDU commands. For instance, if the EMV card checks only in the “GPO” APDU command, the attackers could skip this command and send the “Read Record” commands and obtain most of the EMV contactless card’s sensitive information by launching skimming attack. Therefore, the three checks are vital in order to prevent this scenario.

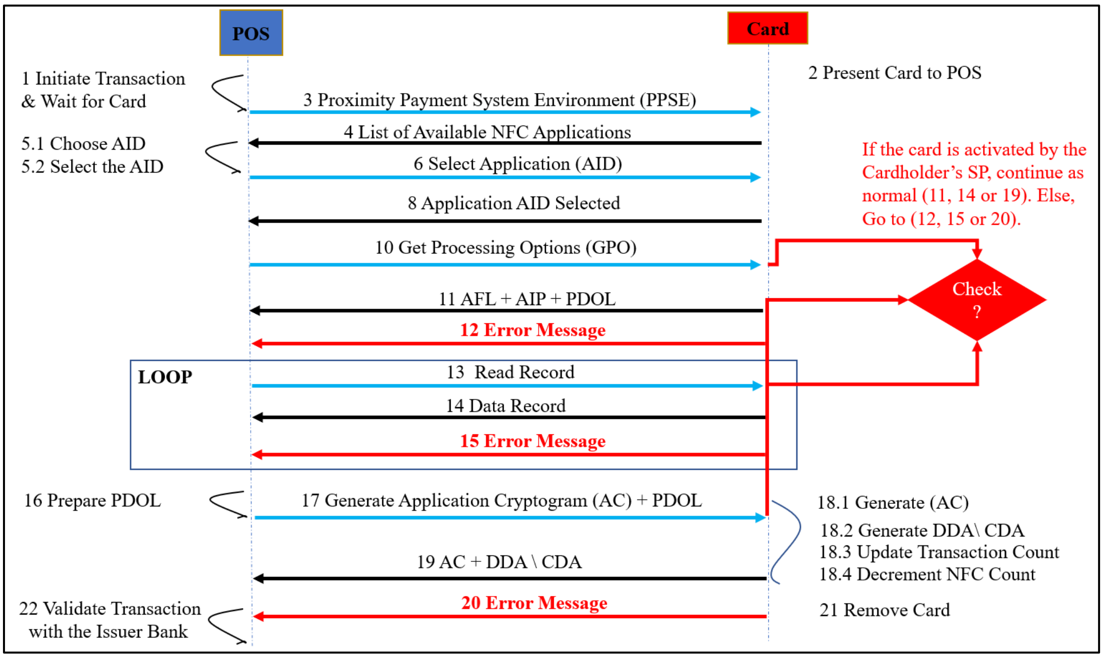

POS
Card
1 Initiate Transaction & Wait for Card
2 Present Card to POS
3 Proximity Payment System Environment (PPSE)
4 List of Available NFC Applications
5.1 Choose AID
5.2 Select the AID
6 Select Application (AID)
8 Application AID Selected
If the card is activated by the Cardholder's SP, continue as normal (11, 14 or 19). Else, Go to (12, 15 or 20).
10 Get Processing Options (GPO)
Check ?
11 AFL + AIP + PDOL
12 Error Message
LOOP
13 Read Record
14 Data Record
15 Error Message
16 Prepare PDOL
17 Generate Application Cryptogram (AC) + PDOL
18.1 Generate (AC)
18.2 Generate DDA\ CDA
18.3 Update Transaction Count
18.4 Decrement NFC Count
19 AC + DDA \ CDA
20 Error Message
22 Validate Transaction with the Issuer Bank
21 Remove Card

Figure 8-3: The Proposed Scheme in the Transaction Phase

8.3. Implementations & Results of the NFC Smartphone Activation Proposed Protocol

The section explains the implementation details of the proposal to illustrate its feasibility. We have used the same hardware and software which were detailed in Chapter 3 in order to implement this proposed protocol.

8.3.1. Implementation of the Activation by the NFC Smartphone Proposed Protocol

We designed a new Java applet in order to simulate all the required processing steps in the proposal, as explained in the previous section.

In this implementation, we set up the ICCMK as 16 bytes of 3DES as follows (0123456789ABCDEF0123456789ABCDEF). Moreover, 8 bytes of both of the Card_RC and the SP-RC were as (1122334455667788) and (8877665544332211) respectively. Table 8-2 shows all the required APDU commands and responses between the cardholder's NFC enabled smartphone and the EMV contactless card.

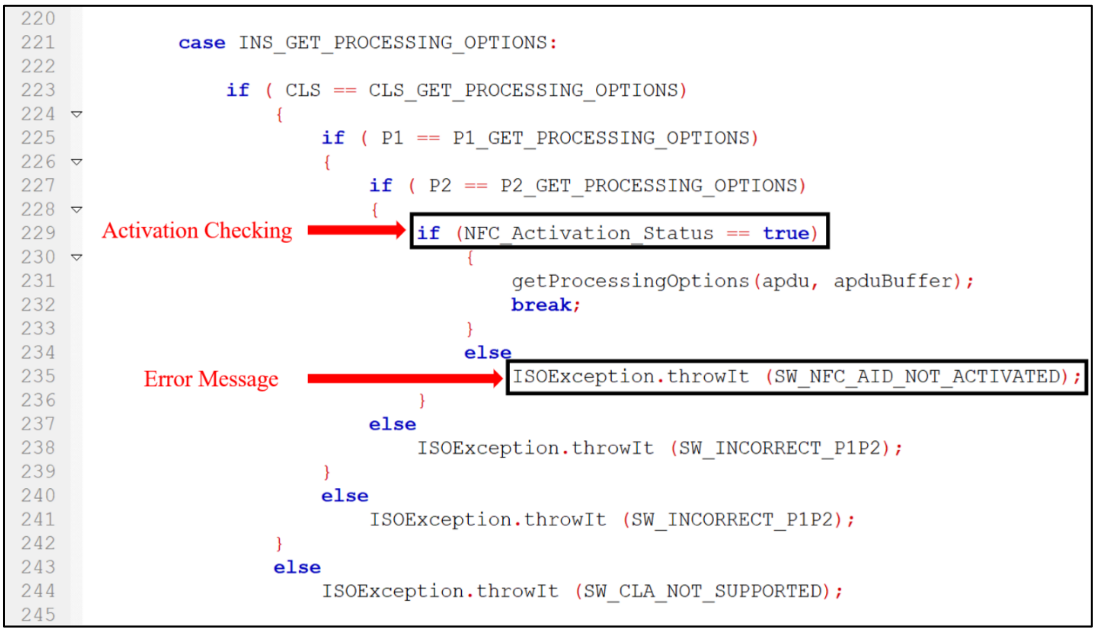
As shown in Table 8-2, the "PPSE" and "Select AID" APDU commands and responses are identical to the original APDU commands and responses in the EMV contactless card transactions as the proposal did not change these two commands. However, in the next APDU command “Get Challenge”, the cardholder’s NFC enabled smartphone sends this command as detailed in Table 8-1. Then, the EMV contactless card responds with the encrypted version of Card_RC (ICCMK{Card_RC}).

Furthermore, in the “External Authenticate” APDU command, the cardholder’s NFC enabled smartphone sends 16 bytes of both the ICCMK{A} and the ICCMK{SP_RC} as detailed in 8.2.2. Then, the EMV contactless card checks both of these received cryptograms and decides whether to activate the NFC communication or not according to the checking results.

Table 8-2: APDU Commands & Responses are required to Activate the EMV Contactless Card

Name of APDU Commands	**NFC Enabled Smartphone APDU Commands**	**Java Card APDU Responses**
PPSE	00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 00	6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90 00
Select AID	00 A4 04 00 07 A0 00 00 00 03 10 10 00	6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 90 00
Get Challenge	00 84 00 00 00 00	**B4 CC 3F D9 D8 D9 52 14** 90 00
External Authenticate	00 82 00 00 10 **77 14 F3 24 D1 B2 8D D4 58 77 2F EC 6A 80 6F 7F** 00	90 00

Figure 8-4 shows an example of a part of updating the original Java applet in order to implement the proposal. We added the checking processing as shown in Figure 8-4 before processing all of the “GPO”, “Read Record” and “Generate AC” APDU commands. The error message is also added to the Java applet in case of the NFC_Activation_Status is false.



```
        case INS_GET_PROCESSING_OPTIONS:

            if ( CLS == CLS_GET_PROCESSING_OPTIONS)
                {
                    if ( P1 == P1_GET_PROCESSING_OPTIONS)
                    {
                        if ( P2 == P2_GET_PROCESSING_OPTIONS)
                        {
                        if (NFC_Activation_Status == true)
                            {
                                getProcessingOptions(apdu, apduBuffer);
                                break;
                            }
                            else
                                ISOException.throwIt (SW_NFC_AID_NOT_ACTIVATED);
                        }
                        else
                            ISOException.throwIt (SW_INCORRECT_P1P2);
                    }
                    else
                        ISOException.throwIt (SW_INCORRECT_P1P2);
                }
                else
                    ISOException.throwIt (SW_CLA_NOT_SUPPORTED);
```

Figure 8-4: Example of the Java Applet Code Showing the Activation Check and the Error Message

8.3.2. Time Overhead Analysis

The time results of this proposal are divided into two main parts. The first part emphases the require time to process the four APDU commands and responses between the cardholder’s NFC enabled smartphone and the EMV contactless card in the activation request phase, as explained in 8.2.2. While the second part focuses on the additional required time during the contactless card transactions between the POS and the EMV contactless card. in both scenarios, we calculated the time average after repeating each APDU command 10 times.

In the first part, the overall reported time by PyApduTool was 295 ms for all the four required APDU commands and responses. The time is divided to 55ms to process the “PPSE” command, 70ms to process the “Select AID” command. The reported times match the ones in previous chapters such as the ones in Section 6.5.1, as we did not change these two APDU commands and responses. However, the time required to process the “Get Challenge” APDU command is around 65ms while it is almost 105ms for the “External Authenticate” APDU command.

In the second part, Figure 8-5 shows the timing results of the two Java applets, where the first Java applet represents the original EMV contactless card. In contrast, the second Java applet represents the proposed scheme. As shown in the figure, both of the “PPSE” and “Select AID” APDU commands in both of the Java applets spend the same time whereas the Java applet representing the proposal spends between 3 ms to 5 ms more than the original Java applet in the remaining APDU commands. The reason behind these extra milliseconds is due the checking processing, which was done before processing the APDU commands. The total time required is 23ms in all the APDU commands and responses.

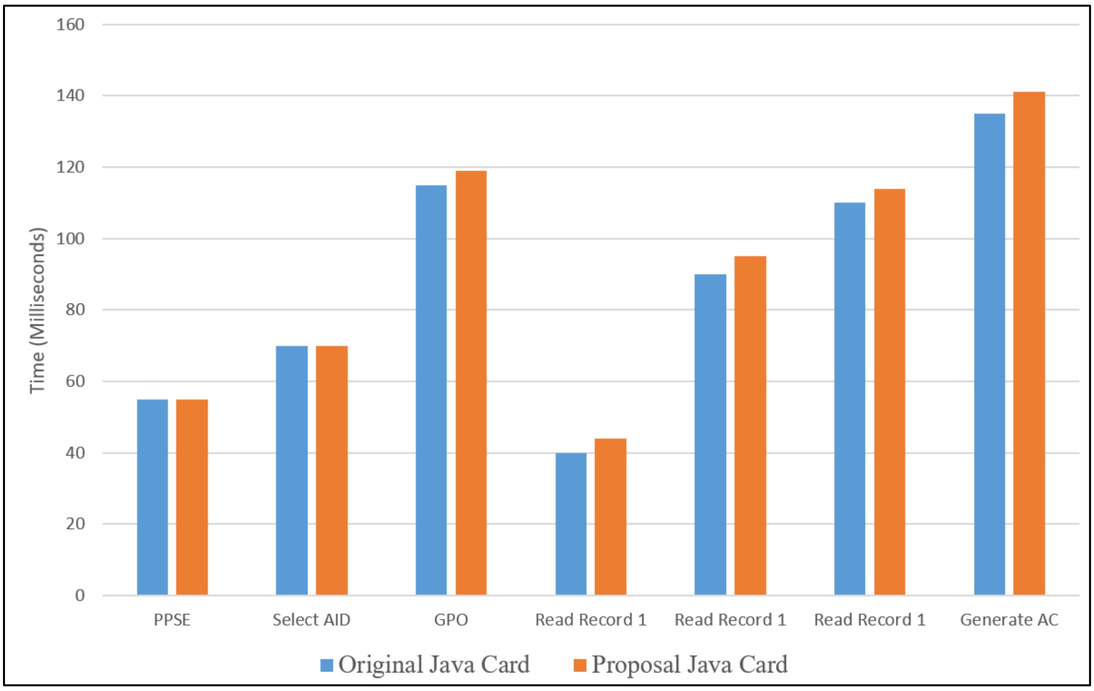


Figure 8-5:Comparing the timing of the Original Java Applet and the Proposal of Using the Cardholder’s Smartphone

8.4. Study and Limitations of the Proposed Protocol

The section points out several discussion points related to the proposed scheme, such as usability, user experience, internet connection, and security analysis.

8.4.1. Usability Vs. User Experience

Usability refers to how easily the end-users interact with products, objects, and tools. Therefore, the usability of any new proposal is vital for the end-users, where a successful and effective proposal needs to be accepted by the end-users. We are aware that this proposed scheme requires several actions that the cardholder needs to do before processing the contactless card transactions, as explained in the previous sections. These required actions affect the users' experience especially in this kind of transaction (contactless cards). As the main aim of these transactions is to be easy and quick to achieve by the cardholders. Thus, we propose deploying the proposed scheme by the EMV payment protocol as an optional feature to the cardholders. That means each cardholder can decide whether to use the proposed scheme or not. This could be achieved by the use of the IB mobile banking application, where this application offers the cardholder the ability to choose to use extra protection on his EMV contactless card or not.

8.4.2. Internet Connection Offline Scenario

Most of the IBs' mobile bank applications require the availability of the internet by any source such as Wi-Fi or mobile data usage or other internet sources in order to connect these applications with the IBs. As these kinds of applications need to update continuously to manage the different accounts and show the transaction history and various other services that require direct updates by the IBs. Our proposed scheme proposes the use of such applications to connect with the EMV contactless cards and send the activation request, as explained in Section 8.2.

In various scenarios in real life, internet services could be offline for any reasons. This could make the cardholder unable to access the IB's mobile banking application and activate his EMV contactless card in order to do contactless card transactions. To solve this problem, we suggest that the IBs offer the ability to use certain features in their mobile banking application in the offline mode, as some of the IBs are doing currently. The recommended offline mode offers the cardholders the ability to use the IB mobile bank application to send the activation request to the EMV contactless card, even if there is no internet connectivity in the cardholders' smartphones.

8.4.3. Security Analysis

The security of the proposed scheme relies on one key only, that is the ICCMK. This key is used to encrypt and decrypt the EMV contactless cards' challenges

The attackers can use off-the-shelf NFC enabled readers/smartphones to communicate with the victim's EMV contactless card without his knowledge and launch skimming or relay attacks on such cards as detailed previously in Section 2.2.1. However, the attackers will fail to fool the EMV contactless cards by sending the activation request APDU command by using any NFC enabled readers/smartphones. The reason behind that is the EMV contactless cards verify the APDU activation request command by challenging the sender. The verification process depends on the ICCMK which is used to encrypt and decrypt the challenge between the EMV contactless card and any sender of the activation APDU command request. The ICCMK key is stored at the EMV card's chip at the personalisation phase by the IB. It is a fact that the EMV card's chip is a secure environment to store this key and the attackers cannot obtain this key by launching any kind of attacks such as brute force attack.

8.5. Summary of the Chapter

The chapter presented our proposal to use the cardholders' NFC enabled smartphones to activate the EMV contactless cards in order to prevent both skimming and relay attacks on such cards. The proposed scheme gives the cardholders more control over their EMV contactless cards, so they can decide when to engage in any NFC communications.

This proposal does not require any extra hardware as we relied on the cardholders' NFC enabled smartphones to communicate with the EMV contactless cards. We also used two of the existing issuer scripts APDU commands namely "Get Challenge" and "External Authenticate" APDU commands in order to change NFC status of such cards. the implementation results showed that the time require to activate the EMV contactless card is around 295ms while the proposal needs an additional time between 20ms to 30ms depends on the number of records that are supported by the EMV card. It can be argued that the timing overhead is resendable for this kind of transaction.

On the other hand, we do admit that the proposal could affect the users' experience by asking the cardholders to do extra steps before doing a contactless card transaction. Therefore, we propose making our proposal as an option that the cardholders could choose to do in case if they feel the need an addition layer of security to be added to their contactless cards.

9. Chapter 9: Conclusion & Future Work

This chapter recaps on the main aim of this thesis and all the proposed contributions to achieve the aim. The chapter also highlights some possible future research directions.

9.1. Summary of the Thesis

As shown in Chapter 2, the EMV contactless card transactions can be subjected to different attacks such as skimming, eavesdropping, replay, and relay attacks. All these attacks and more are possible to be launched mainly due to the wireless connectivity between POS and EMV contactless cards along with a key vulnerability in the EMV payment protocol, that is the one-way authentication methods. Furthermore, the lack of CVMs leads to a missing link between the cardholders and their EMV contactless cards.

Chapter 3 detailed our first contribution where we developed and built a Java framework for the EMV contactless cards. The whole development and building process were according to the EMV payment specifications. The reason behind that, we wanted to duplicate the genuine EMV contactless cards in order to test the Java framework on our next proposed solutions. This Java framework helped us to understand more about each functionality of the EMV contactless card transactions. This Java framework used to implement and test most of our next contributions by uploading and modifying the Java framework in order to suit the required changes in the contributions.

In Chapter 4, we investigated the EMV card authentication methods namely SDA, DDA, and CDA. The investigation process went through the four main stages of these methods namely keys distribution, personalisation, transaction and authentication stages. These methods provide a one-way authentication protocol that serves the POS side only while the EMV cards do not have any option to authentication the POS. Implementations one of each of the three card authentication methods were presented in Chapter 4 using of the Java framework that developed in Chapter 3. We explained how each one of these methods reacted to both of sniffing and replay attacks. As a result of our investigation of the EMV card authentication methods, we had to come up with a mutual authentication protocol to replace the one-way authentication and give the EMV cards a chance to verify the authenticity of the POS.

To achieve the main aim of the thesis in improving the security of the EMV contactless card transactions, we proposed four protocols to withstand the mentioned attacks. The following are a summary of the contributions.

1- A tokenisation proposal in Chapter 5 was our first attempt to enhance the security of the EMV contactless cards. As detailed in Chapter 2, the EMV contactless cards can be a subject of a skimming attacks by the use of off-the-shelf hardware or software without the knowledge of the cardholder. Our tokenisation proposal relies on the idea of replacing the actual PAN that is currently being sent in plaintext during the transactions with a token in order to minimize the risk of skimming such information. We showed how the tokenisation technique works at three different phases personalisation, transaction, and authorisation, where minimum changes are required to deploy this proposed solution by the EMV payment protocol as there is no change in the exited original EMV infrastructure.

2- Chapter 6 explained our second attempt to improve the security of the EMV contactless card transaction by proposing a mutual authentication protocol between the POS and EMV contactless cards. We showed in our analysis of the EMV card authentication methods in Chapter 3 that the main goal of such methods is to authenticate the EMV cards to the POS while the EMV cards do not have the option to authentication the POS as a genuine one. This one-way authentication method protocol is one of the vulnerabilities besides the wireless connectivity that led to skim the EMV contactless card using any sort of NFC enabled readers/smartphones as showed in Chapter 2. Hence, we came up with the mutual authentication protocol in order to give such cards the chance to verify the authenticity of POSs. By doing so, the EMV contactless cards will no longer be read by unauthorised NFC enabled readers/smartphones. To minimise the changes required to deploy the proposed protocol, we did not add any extra APDU commands to the original EMV APDU commands. Furthermore, we showed that the proposed protocol has very little impact on the time of transactions.

3- In Chapter 7, we proposed the use of built-in gyroscope sensor in order to bring back some sort of a relationship between the EMV contactless cards and cardholders. As the EMV contactless card specifications have no CVMs in such transactions, the link between the cardholders and their EMV contactless cards was missing. The missing link has led to various attacks such as relay attacks. The main aim of the proposed solution was to prevent the EMV contactless card from being read by any POS or even NFC enabled readers/smartphones unless some sort of movement is performed by the genuine cardholder.

4- In Chapter 8, we proposed the use of the cardholder's smartphone in order to activate and allow the EMV contactless card to engage in wireless activities. The proposed solution took advantage of most of the modern smartphones are NFC enabled. This functionality makes it possible for smartphones to send and receive APDU commands and responses. The proposed technique aims to bring back some kind of control to the cardholder on his own EMV contactless card and prevent such card from being read or communicate with unauthorised POS or even NFC enabled readers/smartphones.

The first two proposals to improve the security of the EMV contactless cards required minimum changes in the original EMV payment infrastructures. For this reason, deploying both of these two solutions by the EMV payment protocol is possible and will not affect the original infrastructures. However, our last two solutions required more changes in comparison with the first two in the EMV original infrastructures whether the changes are based on hardware or software.

9.2. Future Work

More work should be done to make the EMV contactless card transaction more secure. The focus of our future work will be in two main points as follows:

9.2.1. Location as an Additional Authentication Factor

Using the location as an additional authentication factor to improve the security of the EMV contactless card transactions is an exciting approach. Figure 9-1 shows our suggested framework for this idea where the location of the cardholder is provided by two independent sources. The first source is the cardholder's smartphone while the second source is the Mobile Network Operator (MNO). The IB compares the two locations of the cardholder's smartphone that were provided by the two independent sources with the added location of the POS as shown in step number 4 in Figure 9-1. If all the locations are close enough to each other. Then the IB approves the transaction. Else, the transaction is declined by the IB. As shown in the same figure that most of the added steps and required processing are done at the IB end back system. Therefore, this will not affect the transaction time and not require any change in the original EMV infrastructure. However, the only change is the process of adding the location of the POS into the transaction details.

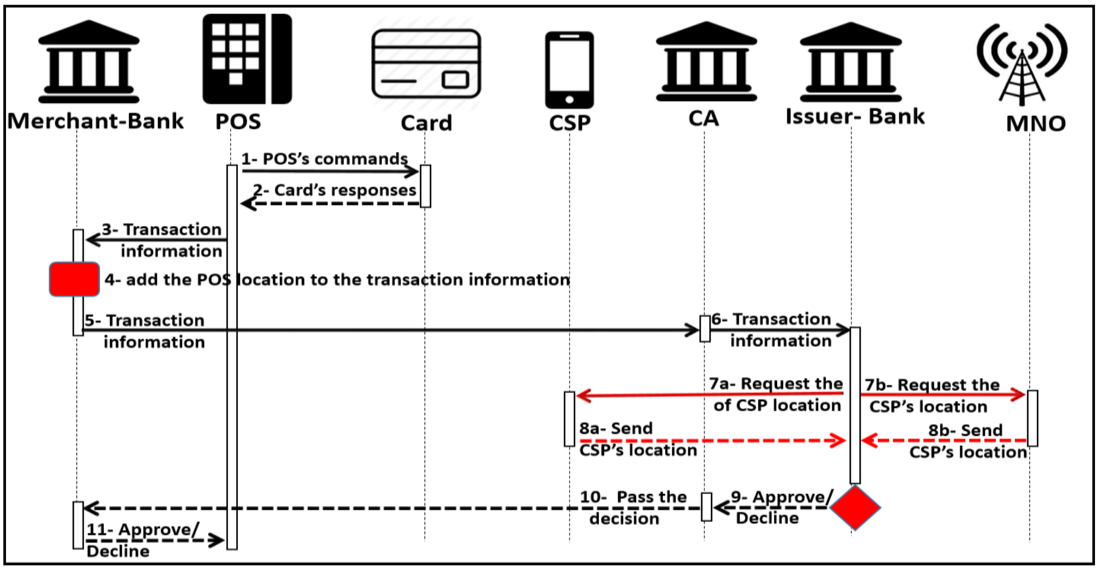

Merchant-Bank
POS
Card
CSP
CA
Issuer- Bank
MNO
1- POS's commands
2- Card's responses
3- Transaction information
4- add the POS location to the transaction information
5- Transaction information
6- Transaction information
7a- Request the of CSP location
7b- Request the CSP's location
8a- Send CSP's location
8b- Send CSP's location
9- Approve/ Decline
10- Pass the decision
11- Approve/ Decline

Figure 9-1: Location as Authentication Factor for EMV Contactless Cards

9.2.2. Lightweight Encryption of the APDU Commands & Responses

All the APDU commands and responses are sent in plaintext during the EMV transactions whether contact, contactless and mobile transactions. Because of that, downgrade attack is possible especially on the EMV contactless cards due to the wireless connectivity.

As detailed in Section 2.2.1.2 that some attackers gained the advantage of sending the APDU commands and responses in the clear without any sort of encryption. This helps the attacker to modify the APDU responses and fool the POS in order to serve the attacker's purposes as discussed in [63][65][66]. Therefore, proposing the use of some kind of encryption method to encrypt the APDU commands and responses during the transactions will prevent such attacks and enhance the security of the EMV payment protocol in general and the contactless card transaction in particular.

References

[1] D. Paret, “RFID and contactless smart card applications,” *IEEE Electrical Insulation Magazine*, vol. 22, no. 5. pp. 52–53, 2006, doi: 10.1109/MEI.2006.1705870.

[2] J. Van Den Breekel, D. A. Ortiz-yepes, E. Poll, and J. De Ruiter, “EMV in a nutshell,” in *Technical Report*, 2016, pp. 1–37.

[3] G. Jain and S. Dahiya, “NFC: Advantages, Limits and Future Scope,” *Int. J. Cybern. Informatics*, vol. 4, no. 4, pp. 1–12, 2015, doi: 10.5121/ijci.2015.4401.

[4] I. S. Iso, “Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 3: Initialization and anticollision,” vol. 2, 2001.

[5] ISO/IEC, *ISO/IEC 14443-4 - Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 4: Transmission protocol*. 2001.

[6] T. Trütsch, “The Impact of Contactless Payment on Spending,” *Int. J. Econ. Sci.*, 2014.

[7] The Uk Cards Association, “Guide for retailers: Accepting contactless and higher value contactless payments,” 2017.

[8] I. Lacmanović, B. Radulović, and D. Lacmanović, “Contactless payment systems based on RFID technology,” *MIPRO 2010 - 33rd Int. Conv. Inf. Commun. Technol. Electron. Microelectron. Proc.*, pp. 1114–1119, 2010.

[9] N. A. Chattha, “NFC - Vulnerabilities and defense,” in *Conference Proceedings - 2014 Conference on Information Assurance and Cyber Security, CIACS 2014*, 2014, doi: 10.1109/CIACS.2014.6861328.

[10] T. P. Diakos, J. A. Briffa, T. W. C. Brown, and S. Wesemeyer, “Eavesdropping near-field contactless payments: a quantitative analysis,” *J. Eng.*, vol. 2013, no. 10, pp. 48–54, 2013, doi: 10.1049/joe.2013.0087.

[11] S. Lefophane and J. Van Der Merwe, “A security review of proximity identification based smart cards,” in *Proceedings of the 10th International*

Conference on Cyber Warfare and Security, ICCWS 2015, 2015, pp. 534–541.

[12] S. Ghosh, J. Goswami, A. Kumar, and A. Majumder, "Issues in NFC as a form of contactless communication: A comprehensive survey," in *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, ICSTM 2015 - Proceedings*, 2015, doi: 10.1109/ICSTM.2015.7225422.

[13] P. Fillmore, "Overview of Contactless Payment Cards An overview of the EMV standards," *Black Hat*, pp. 1–9, 2015.

[14] J. Jumic and M. Vukovic, "Analysis of credit card attacks using the NFC technology," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2017 - Proceedings*, 2017, doi: 10.23919/MIPRO.2017.7973615.

[15] EMVCO, "Worldwide EMV® Deployment Statistics - EMVCo." [Online]. Available: https://www.emvco.com/about/deployment-statistics/. [Accessed: 06-May-2020].

[16] UK Finance, "Contactless Transit: Implementation in the UK," vol. 2, pp. 1–42, 2019.

[17] UK Finance, *UK Card Payments:An analysis of trends and card payments in contactless, debit, credit card lending*. UK Finance, 2018.

[18] UK Finance, "FRAUD - THE FACTS 2020 The definitive overview of payment industry fraud," 2020.

[19] UK Finance, "2019: Half Year Fraud Report," 2019. [Online]. Available: https://www.ukfinance.org.uk/system/files/2018-half-year-fraud-update-FINAL.pdf.

[20] S. Parusheva, "Card-not-present fraud - challenges and counteractions.," *Econ. Arch. / Narodnostopan. Arh.*, 2015.

[21] R. Sanders, "From EMV to NFC: the contactless trail?," *Card Technol. Today*, 2008, doi: 10.1016/S0965-2590(08)70077-X.

[22] T. Guardian, “Contactless card limit rises to £30 after surge in transactions,” *The Guardian,* 2015. [Online]. Available: https://www.theguardian.com/business/2015/sep/01/contactless-card-limit-rises-to-30-after-surge-in-transactions. [Accessed: 15-Apr-2020].

[23] British Retail Consort, “Retailers Combat Covid with Contactless,” 2020. [Online]. Available: http://brc.org.uk/news/corporate-affairs/retailers-combat-covid-with-contactless/. [Accessed: 15-Apr-2020].

[24] UK Finance, “Contactless limit in UK to be increased to £45 | UK Finance.” [Online]. Available: https://www.ukfinance.org.uk/press/press-releases/contactless-limit-uk-be-increased-45. [Accessed: 15-Apr-2020].

[25] “Home - EMVCo.” [Online]. Available: https://www.emvco.com/. [Accessed: 15-Jan-2020].

[26] EMVCo, “Book 1: Application Independent ICC to Terminal Interface Requirements,” vol. 4.3, no. November, 2011.

[27] EMVCo, *Book 2: Security and Key Management*, no. November. 2011.

[28] EMVCo, *Book 3: Application Specification, Integrated Circuit Card Specifications for Payment Systems*, no. November. 2011.

[29] EMVCo, *Book 4: Cardholder, Attendant, and Acquirer Interface Requirements*, vol. 4.3, no. November. 2011.

[30] EMVCo, *Book A Architecture and General Requirements*, vol. 2.9, no. March. 2020.

[31] EMVCo, *Book B: Entry Point Specification*, vol. 2.9, no. March. 2020.

[32] EMVCo, *Book D: EMV Contactless Communication Protocol Specification*, vol. 2.6, no. March. 2016.

[33] EMVCO, *Book C-1: Kernel 1 Specification*, vol. 2.6, no. February. 2016.

[34] EMVCO, *Book C-2: Kernel 2 Specification*, vol. 2.9, no. May. 2020.

[35] EMVCO, *Book C-3: Kernel 3 Specification*, no. 2.9. 2020.

[36] EMVCO, *Book C-4: Kernel 4 Specification*, vol. 2.9, no. March. 2020.

[37] EMVCO, *Book C-5: Kernel 5 Specification*, vol. 2.9, no. March. 2020.

[38] EMVCO, *Book C-6: Kernel 6 Specification*, vol. 29, no. March. 2020.

[39] EMVCO, *Book C-7: Kernel 7 Specification*, vol. 2.9, no. March. 2020.

[40] First Data, "EMV: A to Z (Terms and Definitions)," 2013.

[41] O. Ogundele, P. Zavarsky, R. Ruhl, and D. Lindskog, "The implementation of a full EMV smartcard for a point-of-sale transaction and its impact on the PCI DSS," in *Proceedings - 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust and 2012 ASE/IEEE International Conference on Social Computing, SocialCom/PASSAT 2012*, 2012, doi: 10.1109/SocialCom-PASSAT.2012.80.

[42] Cryptomathic, "White Paper EMV Key Management," 2013.

[43] D. Jayasinghe, R. N. Akram, K. Markantonakis, K. Rantos, and K. Mayes, "Enhancing EMV online PIN verification," in *Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015*, 2015, doi: 10.1109/Trustcom.2015.451.

[44] A. Alhothaily, A. Alrawais, X. Cheng, and R. Bie, "A novel verification method for payment card systems," *Pers. Ubiquitous Comput.*, 2015, doi: 10.1007/s00779-015-0881-9.

[45] V. Coskun, B. Ozdenizci, and K. Ok, "The survey on near field communication," *Sensors (Switzerland)*, 2015, doi: 10.3390/s150613348.

[46] C. Olsen, "Getting the most out of EMV with contactless cards," *Card Technol. Today*, 2007, doi: 10.1016/S0965-2590(07)70078-6.

[47] G. Madlmayr, C. Kantner, and T. Grechenig, "Near field communication," in *Secure Smart Embedded Devices, Platforms and Applications*, 2014.

[48] V. Coskun, B. Ozdenizci, and K. Ok, "A survey on near field communication (NFC) technology," *Wireless Personal Communications*. 2013, doi: 10.1007/s11277-012-0935-5.

[49] V. Rajaraman, “Radio frequency identification,” *Resonance*, 2017, doi: 10.1007/s12045-017-0498-6.

[50] X. Leng, “Smart card applications and security,” *Inf. Secur. Tech. Rep.*, 2009, doi: 10.1016/j.istr.2009.06.006.

[51] M. Emms and A. van Moorsel, “Practical Attack on Contactless Payment Cards,” in *HCI2011 Workshop - Heath, Wealth and Identity Theft*, 2011.

[52] M. Mehrnezhad, M. A. Ali, F. Hao, and A. van Moorsel, “NFC payment spy: A privacy attack on contactless payments,” in *International Conference on Research in Security Standardisation*, 2016, vol. 10074 LNCS, pp. 92–111, doi: 10.1007/978-3-319-49100-4_4.

[53] N. Akinyokun and V. Teague, “Security and privacy implications of NFC-enabled contactless payment systems,” in *ACM International Conference Proceeding Series*, 2017, doi: 10.1145/3098954.3103161.

[54] M. Mehrnezhad, F. Hao, and S. F. Shahandashti, “Tap-tap and pay (TTP): Preventing the mafia attack in NFC payment,” *Proc. Second Int. Conf. Secur. Stand. Res.*, vol. 9497, pp. 21–39, 2015, doi: 10.1007/978-3-319-27152-1_2.

[55] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, “Practical relay attack on contactless transactions by using NFC mobile phones,” *Cryptol. Inf. Secur. Ser.*, vol. 8, pp. 21–32, 2012, doi: 10.3233/978-1-61499-143-4-21.

[56] Z. Kfir and A. Wool, “Picking virtual pockets using relay attacks on contactless smartcard systems,” in *Proceedings - First International Conference on Security and Privacy for Emerging Areas in Communications Networks, SecureComm 2005*, 2005, doi: 10.1109/SECURECOMM.2005.32.

[57] J. van den Breekel, “Relaying EMV Contactless Transactions using Off-The-Shelf Android Devices,” in *BlackHat Asia, Singapore*, 2015.

[58] M. Roland, J. Langer, and J. Scharinger, “Applying relay attacks to Google Wallet,” in *2013 5th International Workshop on Near Field Communication, NFC 2013*, 2013, doi: 10.1109/NFC.2013.6482441.

[59] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, “Practical NFC peer-

to-peer relay attack using mobile phones," in *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, 2010, doi: 10.1007/978-3-642-16822-2_4.

[60] P. Li, H. Fang, X. Liu, and B. Yang, "A countermeasure against relay attack in NFC payment," in *ACM International Conference Proceeding Series*, 2017, doi: 10.1145/3018896.3025144.

[61] F. Lipson, "Contactless card fraud warning: Crooks can use them MONTHS after cancellation." [Online]. Available: https://www.moneysavingexpert.com/news/2016/09/card-lost-or-stolen-beware---you-could-be-the-victim-of-contactless-fraud-months-after-youve-cancelled-it/. [Accessed: 12-Jan-2020].

[62] "How fraudsters can use your contactless credit and debit cards AFTER you've cancelled them with the bank – The Sun." [Online]. Available: https://www.thesun.co.uk/living/1749352/fraudsters-used-my-card-seven-months-after-it-was-stolen-contactless-card-flaw-that-could-put-thousands-at-risk/. [Accessed: 12-Jan-2020].

[63] M. Roland, A. Usenix, and W. Washington, "Cloning Credit Cards : A combined pre-play and downgrade attack on EMV Contactless," vol. 2013, no. August, 2013.

[64] M. Bond, O. Choudary, S. J. Murdoch, S. Skorobogatov, and R. Anderson, "Chip and skim: Cloning EMV cards with the pre-play attack," in *Proceedings - IEEE Symposium on Security and Privacy*, 2014, doi: 10.1109/SP.2014.11.

[65] M. Emms, B. Arief, N. Little, and A. Van Moorsel, "Risks of offline verify PIN on contactless cards," *Int. Conf. Financ. Cryptogr. Data Secur.*, vol. 7859 LNCS, pp. 313–321, 2013, doi: 10.1007/978-3-642-39884-1_26.

[66] M. Emms, B. Arief, L. Freitas, J. Hannon, and A. Van Moorsel, "Harvesting high value foreign currency transactions from EMV contactless credit cards without the PIN," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 716–726, 2014, doi: 10.1145/2660267.2660312.

[67] L. Hong, H. C. Yong, and Q. H. Zhang, "The survey of RFID attacks and

defenses," in *2012 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2012*, 2012, doi: 10.1109/WiCOM.2012.6478720.

[68] J. W. Yum, B. C. Yoo, K. Y. Park, and J. H. Jang, "Smart card with an integrated electrical switch for secure operation," in *International Workshop on Antenna Technology: Small Antennas, Innovative Structures and Materials*, 2010, doi: 10.1109/IWAT.2010.5464687.

[69] N. El Madhoun, E. Bertin, and G. Pujolle, "An overview of the EMV protocol and its security vulnerabilities," *2018 4th Int. Conf. Mob. Secur. Serv. MOBISECSERV 2018*, vol. 2018-Febru, pp. 1–5, 2018, doi: 10.1109/MOBISECSERV.2018.8311444.

[70] H. Vats, R. Ruhl, and S. Aghili, "Fingerprint security for protecting EMV payment cards," in *2015 10th International Conference for Internet Technology and Secured Transactions, ICITST 2015*, 2016, doi: 10.1109/ICITST.2015.7412065.

[71] C. H. Kim, G. Avoine, F. Koeune, F. X. Standaert, and O. Pereira, "The Swiss-Knife RFID distance bounding protocol," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, doi: 10.1007/978-3-642-00730-9_7.

[72] S. Drimer and S. J. Murdoch, "Keep your enemies close: Distance bounding against smartcard relay attacks," in *16th USENIX Security Symposium*, 2007.

[73] G. P. Hancke and M. G. Kuhn, "An RFID distance bounding protocol," *Proc. - First Int. Conf. Secur. Priv. Emerg. Areas Commun. Networks, Secur. 2005*, vol. 2005, pp. 67–73, 2005, doi: 10.1109/SECURECOMM.2005.56.

[74] C. H. Kim and G. Avoine, "RFID distance bounding protocol with mixed challenges to prevent relay attacks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, doi: 10.1007/978-3-642-10433-6_9.

[75] S. Guizani, "Implementation of an RFID relay attack countermeasure," in *IWCMC 2015 - 11th International Wireless Communications and Mobile*

Computing Conference, 2015, doi: 10.1109/IWCMC.2015.7289273.

[76] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji, “Detecting relay attacks with timing-based protocols,” in *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, ASIACCS ’07*, 2007, doi: 10.1145/1229285.1229314.

[77] T. Chothia, F. D. Garcia, J. De Ruiter, J. Van Den Breekel, and M. Thompson, “Relay cost bounding for contactless EMV payments,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, doi: 10.1007/978-3-662-47854-7_11.

[78] JavaCardOS, “JC30M48CR Java Card | JavaCardOS Store.” [Online]. Available: https://www.javacardos.com/store/javacard-jc30m48cr.php. [Accessed: 20-Jan-2020].

[79] PayPal, “PayPal Here Card Reader - PayPal Here - UK.” [Online]. Available: https://uk.paypal-here.com/paypal-here-card-reader/. [Accessed: 20-Jan-2020].

[80] Advanced Card Systems: Card & Reader Technologies, *ACR 122U : USB NFC Reader*, vol. 2.3. 2017.

[81] I. Attali, D. Caromel, C. Courbis, L. Henrio, and H. Nilsson, “An integrated development environment for Java Card,” *Comput. Networks*, 2001, doi: 10.1016/S1389-1286(01)00162-1.

[82] EMVCo, *EMV Card Personalization Specification*. 2007.

[83] D. Molvig, “The Mobile Wallet,” *Credit Union Mag.*, 2012.

[84] EMVCo, *Contactless Mobile Payment: Application Activation User Interface Overview, Usage Guidelines, and PPSE Requirements*. 2010.

[85] D. Jayasinghe, K. Markantonakis, R. N. Akram, and K. Mayes, “Enhancing EMV tokenisation with dynamic transaction tokens,” *Int. Work. Radio Freq. Identif. Secur. Priv. Issues*, 2017, doi: 10.1007/978-3-319-62024-4_8.

[86] R. D. Hof, “Apple pay,” *Technol. Rev.*, 2015.

[87] EMVCo, *Payment Tokenisation Specification Technical Framework*, no. September. 2017.

[88] T. T. Scoping SIG and P. S. S. Council, *Information Supplement : PCI DSS Virtualization Guidelines*, no. August. 2011.

[89] D. Jayasinghe, K. Markantonakis, I. Gurulian, R. N. Akram, and K. Mayes, "Extending EMV tokenised payments to offline-environments," in *Proceedings - 15th IEEE International Conference on Trust*, 2016, doi: 10.1109/TrustCom.2016.0095.

[90] EMVCo, *White Paper on Payment Account Reference (PAR)*, no. September. 2019.

[91] A. S. Jawale and J. S. Park, "A security analysis on apple pay," in *Proceedings - 2016 European Intelligence and Security Informatics Conference, EISIC 2016*, 2017, doi: 10.1109/EISIC.2016.041.

[92] EMVCO, "2018: a Year in Review," 2018. .

[93] P. Urien, "Cloud of secure elements: An infrastructure for the trust of mobile NFC services," in *International Conference on Wireless and Mobile Computing, Networking and Communications*, 2014, doi: 10.1109/WiMOB.2014.6962173.

[94] N. El Madhoun and G. Pujolle, "Security enhancements in EMV protocol for NFC mobile payment," in *Proceedings - 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 10th IEEE International Conference on Big Data Science and Engineering and 14th IEEE International Symposium on Parallel and Distributed Proce*, 2016, doi: 10.1109/TrustCom.2016.0289.

[95] M. H. Yang, "Security enhanced emv-based mobile payment protocol," *Sci. World J.*, vol. 2014, 2014, doi: 10.1155/2014/864571.

[96] N. El Madhoun, F. Guenane, and G. Pujolle, "A cloud-based secure authentication protocol for contactless-NFC payment," *2015 IEEE 4th Int. Conf. Cloud Networking, CloudNet 2015*, pp. 328–330, 2015, doi:

10.1109/CloudNet.2015.7335332.

[97] U. B. Ceipidor, C. M. Medaglia, A. Marino, S. Sposato, and A. Moroni, “A protocol for mutual authentication between NFC phones and POS terminals for secure,” pp. 115–120, 2012.

[98] J. H. and A. van M. Martin Emms, Budi Arief, “POS Terminal Authentication Protocol to Protect EMV Contactless,” *Tech. Rep. Ser. Newcastle Univ.*, vol. No. CS-TR-, no. 2, pp. 2–9, 2013, doi: 10.1145/335527.335528.

[99] M. H. Liu, Y. Xin, Y. X. Yang, and X. X. Niu, “Security mechanism research of EMV2000,” in *Proceedings - 2007 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT Workshops 2007*, 2007, doi: 10.1109/WIIATW.2007.4427595.

[100] N. Floissac and Y. L’Hyver, “From AES-128 to AES-192 and AES-256, how to adapt differential fault analysis attacks on key expansion,” in *Proceedings - 2011 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2011*, 2011, doi: 10.1109/FDTC.2011.15.

[101] EMVCO, “Contact EMV Global Adoption,” *Emvco*, 2018. [Online]. Available: https://www.emvco.com/wp-content/uploads/2020/03/EMVCo_WorldMap-20200306.pdf. [Accessed: 07-Apr-2020].

[102] A. Drymonitis and A. Drymonitis, “Introduction to Arduino,” in *Digital Electronics for Musicians*, 2015.

[103] Arduino, “Software Arduino (IDE),” *Arduino*, 2017. .

[104] A. Kataria and M. D. Singh, “A Review of Data Classification Using K-Nearest Neighbour Algorithm,” *Int. J. Emerg. Technol. Adv. Eng.*, 2013.

[105] P. Senin, “Dynamic Time Warping Algorithm Review,” *Science (80-.).*, 2008, doi: 10.1109/IEMBS.2007.4353810.

[106] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, “Euclidean Distance Matrices: Essential theory, algorithms, and applications,” *IEEE Signal Process. Mag.*, 2015, doi: 10.1109/MSP.2015.2398954.

[107] Office for National Statistics (UK), “UK: mobile phone ownership 1996-2018 | Statista,” *Statista*, 2019. [Online]. Available: https://www.statista.com/statistics/289167/mobile-phone-penetration-in-the-uk/. [Accessed: 04-Mar-2020].

[108] ScientiaMobile, “2018 Q4 Mobile Overview Report (MOVR),” *ScientiaMobile*, 2019. [Online]. Available: https://www.scientiamobile.com/2018-q4-download-complete-movr/. [Accessed: 03-Mar-2020].

[109] K. Owusu Kwateng, K. A. Osei Atiemo, and C. Appiah, “Acceptance and use of mobile banking: an application of UTAUT2,” *J. Enterp. Inf. Manag.*, 2019, doi: 10.1108/JEIM-03-2018-0055.

[110] J. Garrett, “Mobile Banking security,” *Credit Union Mag.*, vol. 77, no. 11, pp. 24–28, 2011.

[111] B. P. Technologies, “EMV Issuer scripts and how do they work.” [Online]. Available: https://www.eftlab.com/emv-issuer-scripts-and-how-do-they-work/. [Accessed: 17-Mar-2020].

Appendix A

Appendix A breaks down both the APDU commands and responses and shows the meaning of each tag as following:

1. PPSE APDU Command & Response Decoding

POS → 00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 00

CLA: 00

INS: A4

P1: 04

P2: 00

LC: 0E

Data: 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 (Proximity Payment System Environment)

LE: 00

Card → 6F 3A 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 28 BF 0C 25 61 23 4F 07 A0 00 00 00 03 10 10 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 0A 08 00 01 05 01 00 00 00 00 90 00

6F: File Control Information (FCI) Template

3A: Length of FCI (58 bytes)

84: Dedicated File (DF) Name

0E: Length of DF (14 bytes)

Data: 32 50 41 59 2E 53 59 53 2E 44 44 46 30 31

A5: File Control Information (FCI) Proprietary Template

28: Length of FCI Proprietary (40 bytes)

BF0C: File Control Information (FCI) Issuer Discretionary Data

25: Length of File Control Information (FCI) Issuer Discretionary Data (37 bytes)

61: Application Template

23: Length of Application Template (35 bytes)

4F: Application Identifier (AID)

07: Length of AID (7 bytes)

Data: A0 00 00 00 03 10 10

50: Application Label

0A: Length of Application Label (10 bytes)

Data: 56 69 73 61 20 44 65 62 69 74 (Visa Debit)

87: Application Priority Indicator

01: Length of Application Priority Indicator (1 byte)

Data: 03

9F0A: Application Selection Registered Proprietary Data

08: Length of Application Selection Registered Proprietary Data (8 bytes)

Data: 00 01 05 01 00 00 00 00

9000: SW1 SW2 (Successful)

2. Select AID APDU Command & Response Decoding

POS→ 00 A4 04 00 07 A0 00 00 00 03 10 10 00

CLA: 00

INS: A4

P1: 04

P2:	00

LC:	07

Data:	A0 00 00 00 03 10 10 (Application Identifier (AID) For Visa Debit)

LE:	00

Card →	6F 52 84 07 A0 00 00 00 03 10 10 A5 47 50 0A 56 69 73 61 20 44 65 62 69 74 87 01 03 9F 38 18 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 BF 0C 1A 9F 5A 05 31 08 26 08 26 9F 0A 08 00 01 05 01 00 00 00 00 BF 63 04 DF 20 01 80 90 00

6F:	File Control Information (FCI) Template

52:	Length of File Control Information (FCI) Template (82 bytes)

84:	Dedicated File (DF) Name

07:	Length of Dedicated File (DF) Name (7 bytes)

Data:	A0 00 00 00 03 10 10

A5:	File Control Information (FCI) Proprietary Template

47:	Length of File Control Information (FCI) Proprietary Template (71 bytes)

50:	Application Label

0A:	Length of Application Label (10 bytes)

Data:	56 69 73 61 20 44 65 62 69 74 (Visa Debit)

87:	Application Priority Indicator

01:	Length of Application Priority Indicator (1 byte)

Data:	03

9F38:	Processing Options Data Object List (PDOL)

18: Length of PDOL (24 bytes)

Data: 9F 66 04 9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04

9F66: Terminal Transaction Qualifiers

04: Length of Terminal Transaction Qualifiers

9F02: Transaction Amount

06: Length of Transaction Amount (6 bytes)

9F03: Other Transaction Amount (Cash Back)

06: Length of Other Transaction Amount (6 bytes)

9F1A: Terminal Country Code

02: Length of Terminal Country Code (2 bytes)

95: Terminal Verification Results

05: Length of Terminal Verification Results (5 bytes)

5F2A: Transaction Currency Code

02: Length of Transaction Currency Code (2 bytes)

9A: Transaction Date

03: Length of Transaction Date (YYMMDD)

9C: Transaction Type

01: Length of Transaction Type (1 byte)

9F37: Unpredictable Number

04: Length of Unpredictable Number

BF0C: File Control Information (FCI) Issuer Discretionary Data

1A: Length of File Control Information (FCI) Issuer Discretionary Data (26 bytes)

9F5A: Application Program Identifier

05: Length of Application Program Identifier (5 bytes)

Data: 31 08 26 08 26

9F0A: Application Selection Registered Proprietary Data

08: Length of Application Selection Registered Proprietary Data

Data: 00 01 05 01 00 00 00 00

BF63: Unknown tag

04: Length of Unknown tag

Data: DF 20 01 80

9000: SW1 SW2 (Successful)

3. GPO APDU Command & Response Decoding

POS→ 80 A8 00 00 23 83 21 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

CLA: 80

INS: A8

P1: 00

P2: 00

LC: 23

Data: 83 21 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

LE: 00

Card→ 77 4D 82 02 20 00 94 04 10 02 05 00 57 13 47 51 39 05 86 29 91 07
D2 21 02 21 99 20 00 00 00 00 0F 5F 34 01 00 9F 10 07 06 01 0A 03

```
A0 00 00 9F 26 08 F4 E1 96 C3 0D 93 CD 81 9F 27 01 80 9F 36 02 00
5E 9F 6C 02 2E 00 9F 6E 04 20 70 00 00 90 00
```

77: Response Message Template Format 2

4D: Length of Response Message Template Format 2

82: Application Interchange Profile (AIP)

02: Length of Application Interchange Profile (AIP) (2 bytes)

Data: 2000

94: Application File Locator (AFL)

04: Length of Application File Locator (AFL) (4 bytes)

Data: 10 02 05 00

57: Track 2 Equivalent Data

13: Length of Track 2 Equivalent Data (19 bytes)

Data: 47 51 39 05 86 29 91 07 D2 21 02 21 99 20 00 00 00 00 0F

5F34: Application Primary Account Number (PAN) Sequence Number

01: Length of Application Primary Account Number (PAN) Sequence Number (1 byte)

Data: 00

9F10: Issuer Application Data

07: Length of Issuer Application Data (7 bytes)

Data: 06 01 0A 03 A0 00 00

9F26 Application Cryptogram

F4E196C30D93CD81

9F27 Cryptogram Information Data

80

9F36 Application Transaction Counter (ATC)

005E

9F6C Unknown tag

2E00

9F6E Unknown tag

20700000

90 00: SW1 SW2 (Successful)

4. Read Record APDU Command & Response Decoding

POS→ 00 B2 02 14 00

CLA:	00	
INS:	B2	
P1:	02	Record Number
P2:	14	Short File Identifier (SFI)
LC:	00	
Data:	None	
LE:	00	

Card→ 70 2D 8F 01 08 9F 32 01 03 92 24 7D 45 14 0E E7 12 73 98 AC 86 AD 77 4C 14 FE D1 71 09 B5 F8 E3 2E C1 8A BE ED A0 0A 45 64 2D D4 E1 63 FE 0D 90 00

70:	EMV proprietary Template
2D:	Length of EMV proprietary Template (45 bytes)
8F:	Certification Authority Public Key Index
01:	Length of Certification Authority Public Key Index (1 byte)

08: Certification Authority Public Key Index value

9F32: Issuer Public Key Exponent

01: Length of Issuer Public Key Exponent (1 byte)

03: Issuer Public Key Exponent value

92: Issuer Public Key Remainder

24: Length of Issuer Public Key Remainder (36 bytes)

Data: 7D 45 14 0E E7 12 73 98 AC 86 AD 77 4C 14 FE D1 71 09 B5
F8 E3 2E C1 8A BE ED A0 0A 45 64 2D D4 E1 63 FE 0D

90 00: SW1 SW2 (Successful)

POS→ 00 B2 03 14 00

CLA: 00

INS: B2

P1: 03 Record Number

P2: 14 Short File Identifier (SFI)

LC: 00

Data: None

LE: 00

Card→ 70 81 B3 90 81 B0 6B 77 26 81 B5 EC F6 0B 14 25 0B 71 39 70 2E
92 85 0B AC 81 FC 5D C1 79 53 B5 EF 6E 7B 70 01 B0 3A 46 C2 28
FF B9 C2 28 5C 05 68 04 EC EE BF A0 96 B0 BC C9 55 85 4C 63 93
4A 86 5A 3A D5 44 A7 23 6F 45 2D AB DF DC D7 68 FE F1 B5 DF
C8 BD 29 46 65 A3 B6 E4 C6 4D F4 E2 2C 6B 2D AD A0 38 B8 7E
A6 A0 70 83 04 4C 1F 07 2A 62 3C 41 A8 70 F1 A7 86 FB 9C 99 9E
41 7E 23 8C 3B CC 55 BF DA 86 F9 C3 9B CE 06 DE 7D 85 C5 00
1E 22 08 17 A7 BB 6A E7 66 1D D5 6B 86 E5 A5 82 7A AC C0 98
0E 4D 77 33 B5 D3 AA 7E 70 61 48 A4 B7 27 BD 95 22 AC

90: Issuer Public Key Certificate

81: Length of Issuer Public Key Certificate (129 bytes)

Data: B0 6B 77 26 81 B5 EC F6 0B 14 25 0B 71 39 70 2E 92 85 0B
AC 81 FC 5D C1 79 53 B5 EF 6E 7B 70 01 B0 3A 46 C2 28
FF B9 C2 28 5C 05 68 04 EC EE BF A0 96 B0 BC C9 55 85
4C 63 93 4A 86 5A 3A D5 44 A7 23 6F 45 2D AB DF DC D7
68 FE F1 B5 DF C8 BD 29 46 65 A3 B6 E4 C6 4D F4 E2 2C
6B 2D AD A0 38 B8 7E A6 A0 70 83 04 4C 1F 07 2A 62 3C
41 A8 70 F1 A7 86 FB 9C 99 9E 41 7E 23 8C 3B CC 55 BF
DA 86 F9 C3 9B CE 06 DE 7D 85 C5 00 1E 22 08 17 A7 BB
6A E7 66 1D D5 6B 86 E5 A5 82 7A AC C0 98 0E 4D 77 33
B5 D3 AA 7E 70 61 48 A4 B7 27 BD 95 22 AC

POS→ 00 B2 03 14 00

CLA: 00

INS: B2

P1: 04 Record Number

P2: 14 Short File Identifier (SFI)

LC: 00

Data: None

LE: 00

Card→ 70 1A 5A 08 47 51 39 05 86 29 91 07 5F 24 03 22 10 31 5F 28 02 08
26 9F 07 02 00 00

70: EMV Proprietary Template

1A: Length of EMV Proprietary Template (26 bytes)

5A: Application Primary Account Number (PAN)

08: Length of Application Primary Account Number (PAN) (8 bytes)

Data: 47 51 39 05 86 29 91 07

5F 24: Application Expiration Date

03: Length of Application Expiration Date (3 bytes)

Data: 22 10 31 (YYMMDD)

5F 28: Issuer Country Code

02: Length of Issuer Country Code (2 bytes)

Data: 08 26 (UK)

9F 07: Application Usage Control

02: Length of Application Usage Control (2 bytes)

Data: 00 00

POS→ 00 B2 03 14 00

CLA: 00

INS: B2

P1: 05 Record Number

P2: 14 Short File Identifier (SFI)

LC: 00

Data: None

LE: 00

Card→ 70 81 C2 9F 46 81 B0 95 68 39 3B BF 80 F4 52 91 50 BA 5E 2F A9 CC 0E 62 C0 00 EF CB 62 8A B5 50 FB D5 6E CC 5A 59 3A 92 60 93 97 D1 D7 2D A3 5B E7 02 4F FF B7 F7 CA BF 1E 05 C9 67 E0 22 E0 35 8E 67 DB 4F 5B 88 23 AE AF 0D 6F C1 E4 41 0C 52 B4 33 8F

```
C7 53 3F 21 97 28 D2 86 98 C3 EA DD 64 39 87 DC 5C AF 0E 6E E8
21 3C 7C 82 EC 63 0F 2A 18 D1 B4 9F 89 05 C7 AC 75 23 9A A7 A8
E6 A1 AF 1B 5B B0 94 31 88 24 7C ED E9 36 25 53 70 5F EC CE 0F
B0 D2 8B 76 B7 65 DC 8A 02 4C 9E BB 70 AE 62 3A 55 DC 9F 1E
7B BB F1 2F 45 2B CE F8 AA 5E AA 30 65 D5 46 38 B4 9F 47 01 03
9F 69 07 01 00 00 00 00 00 00
```

9F 46: Integrated Circuit Card (ICC) Public Key Certificate

81: Length of Integrated Circuit Card (ICC) Public Key Certificate (129 bytes)

Data: 95 68 39 3B BF 80 F4 52 91 50 BA 5E 2F A9 CC 0E 62 C0 00 EF CB 62 8A B5 50 FB D5 6E CC 5A 59 3A 92 60 93 97 D1 D7 2D A3 5B E7 02 4F FF B7 F7 CA BF 1E 05 C9 67 E0 22 E0 35 8E 67 DB 4F 5B 88 23 AE AF 0D 6F C1 E4 41 0C 52 B4 33 8F C7 53 3F 21 97 28 D2 86 98 C3 EA DD 64 39 87 DC 5C AF 0E 6E E8 21 3C 7C 82 EC 63 0F 2A 18 D1 B4 9F 89 05 C7 AC 75 23 9A A7 A8 E6 A1 AF 1B 5B B0 94 31 88 24 7C ED E9 36 25 53 70 5F EC CE 0F B0 D2 8B 76 B7 65 DC 8A 02 4C 9E BB 70 AE 62 3A 55 DC 9F 1E 7B BB F1 2F 45 2B CE F8 AA 5E AA 30 65 D5 46 38 B4

9F 47: Integrated Circuit Card (ICC) Public Key Exponent

01: Length of Integrated Circuit Card (ICC) Public Key Exponent (1 byte)

Data: 03

9F 69: Card Authentication Related Data

07: Length of Card Authentication Related Data (7 bytes)

Data: 01 00 00 00 00 00 00