M.Sc. Engg. Thesis

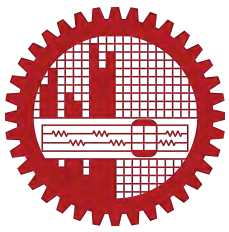
Getting into the Middle of Near Field Communication

by
Sajeda Akter (1014052052 P)

Submitted to

Department of Computer Science & Engineering

(In partial fulfillment of the requirements for the degree of Master of Science in Computer Science & Engineering)



Department of Computer Science & Engineering

Bangladesh University of Engineering & Technology (BUET)

Dhaka 1000

May 05, 2018

Dedicated to my loving parents

Author's Contact

Sajeda Akter

Email: sajeda24@yahoo.com

The thesis titled "Getting into the Middle of Near Field Communication", submitted by Sajeda Akter, Roll No. **1014052052 P**, Session October 2014, to the Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science & Engineering and approved as to its style and contents. Examination held on May 05, 2018.

Board of Examiners

1. _____
Dr. A. B. M. Alim Al Islam — Chairman (Supervisor)
Associate Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

2. _____
Prof. Dr. Md. Mostofa Akbar — Member (Ex-Officio)
Head and Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

3. _____
Prof. Dr. A. K. M. Ashikur Rahman — Member
Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

3. _____
Prof. Dr. Mahmuda Naznin — Member
Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

4. _____
Dr. Nova Ahmed — Member (External)
Associate Professor
Department of Electrical and Computer Engineering
North South University, Dhaka.

Candidate's Declaration

This is hereby declared that the work titled "Getting into the Middle of Near Field Communication", is the outcome of research carried out by me under the supervision of Dr. A. B. M. Alim Al Islam, in the Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology, Dhaka 1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Sajeda Akter

Candidate

Acknowledgment

First of all, I would like to express my heart-felt gratitude to my supervisor, Dr. A. B. M. Alim Al Islam, for his constant supervision of this work. He helped me a lot in shaping, deciding steps of my work, and providing infrastructural supports.

I would also want to thank the honorable members of my thesis committee: Prof. Dr. Md. Mostofa Akbar, Prof. Dr. A. K. M. Ashikur Rahman, Prof. Dr. Mahmuda Naznin, and specially the external member Dr. Nova Ahmed, for their encouragements, insightful comments, and valuable suggestions.

I am grateful to Dr. Sriram Chellappan (Associate Professor, Dept. of Computer Science and Engineering, University of South Florida, USA) and Dr. Anupam Das (Post-doctoral associate, Carnegie Mellon University, USA), who helped me a lot sharing their expert critics and suggestions about this work. They also helped me to give a better shape of this work.

I am also thankful to Novia Nurain (Ph.D. student, CSE, BUET), Tusher Chokroborty (Working at Microsoft Research, India), Taslim Arefin Khan (M.Sc. student, CSE, BUET), and Tarik Reza Toha (M.Sc. student, CSE, BUET). I sought help from them a number of occasions regarding real setup and performance evaluation of this thesis. I am also grateful to all honorable teachers of the department for their comments and suggestions.

Last but not the least, I remain ever grateful to my beloved parents, for their inspirations behind every success of mine.

Abstract

A recent development emanating from the widely used RFID technology is Near Field Communication (NFC). Basically, NFC is a popular short range (<10 cm) wireless communication technology with applications in areas sensitive to security and privacy concerns including contactless payment. Since NFC communications require very close proximity between two communicating devices (for example, a smartcard and a reader), it is generally believed that Man-in-the-Middle (MITM) attacks are practically infeasible here. On the contrary to this general belief, in this research, we successfully establish MITM attack in NFC communications between a passive tag and an active reader. Here, we present physical fundamentals of the attack, our engineering design, and results of our successful implementation. We identify a potential vulnerability in existing contactless payment protocol due to separation between card authentication and transaction authorization phases. Exploiting this vulnerability an attacker is able to conduct transaction interchangeably using original and malicious card. Here, we present practical impacts of the attack from the perspective of how a malicious user can leverage our MITM attack to compromise integrity of contactless payment transactions. We elaborate the complete mechanism of the attack and describe pragmatic attack scenarios to accomplish the practical feasibility of the MITM attack over NFC communication. Through describing different pragmatic attack scenarios, we clarify the beneficiary and loser of this attack. After successfully establishing the attack, we perform rigorous experimental analysis to reveal different aspects of this attack. Finally, we propose a countermeasure to combat the MITM attack based on our experimental analysis. Our proposed countermeasure does not demand any additional hardware to be integrated with the existing system. We evaluate performance of our proposed countermeasure for defending the attack and demonstrate its efficacy in defending the MITM attack.

Acronyms List

AAC = Application Authentication Cryptogram

AES = Advanced Encryption Standard

AFL = Application File Locator

AID = Application Identifier

AIP = Application Interchange Profile

ARPC = Authorization Response Cryptogram

ARQC = Authorization Request Cryptogram

ASCII = American Standard Code for Information Interchange

ATM = Automated Teller Machine

DDA = Dynamic Data Authentication

DDOL = Default Data Object List

EMV = Europay, Mastercard and Visa

fDDA = fast Dynamic Data Authentication

FCI = File Control Information

HF = High Frequency

MAC = Message Authentication Code

MITM = Man-in-the-middle

NFC = Near Field Communication

PAN = Primary Account Number

PIN = Personal Identification Number

PoS = Point of Sale

PPSE = Proximity System Environment

PSE = Payment System Environment

PUF = Physically Unclonable Function

RF = Radio Frequency

RFID = Radio Frequency Identification

SDA = Static Data Authentication

TC = Transaction Certificate

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the past decade, RFID based technologies have been gaining immense popularity with applications in Logistics, Supply Chain Management, Mobility Tracking, Access Control, etc. Within the broad realm of RFIDs, a particular technology is Near Field Communication (NFC). It is a Short range high frequency wireless communication technology which operates at 13.56 MHz frequency and covers less than 10 cm distance. As a finely honed version of High Frequency RFID, near-field communication devices take advantage of the short read range limitations of its radio frequency. To allow communication, since NFC devices must be in close proximity to each other, it has become a popular choice for secure form of data exchange. NFC covers many applications of RFID such as proof of ownership, proof of one's financial assets etc. It also covers some additional applications including contactless payment, access control, and transportation as shown in Figure 1.1. All of these applications are sensitive and demands high level of security.

While existing designs provide a high degree of confidentiality and integrity for NFC communications, one potentially dangerous attack that has not been considered yet in this realm is Man-in-the-Middle attack. Figure 1.2 shows the traditional Man-in-the-Middle attack where a third party gets into the middle of a benign communication and actively communicates with both parties creating an illusion that the benign parties are communicating with each other. In NFC, it is generally believed that MITM attacks are infeasible here because of close proximity between devices and inductive coupling fundamentals, an illustration of which is presented in Figure 1.3. To demonstrate further, let Alice (an active server) and Bob (a passive tag) be two

Figure 1.1: Sensitive application areas of NFC

legitimate entities that are engaging in an NFC communication in close proximity. They can do so, since there exists an RF field generated by Alice. Let Eve be an adversary attempting to launch an MITM attack. In the above scenario, for Eve to launch a successful MITM attack, its RF field needs to be perfectly aligned with that of Alice and Bob (to avoid RF disturbances), which is considered infeasible considering the already close proximity of Alice and Bob (less than 10 cm). Such a situation will likely prevent Eve from becoming a Man-in- the-Middle with NFC communications [1, 2].

1.1 Background on NFC

Being introduced in 2002, NFC already has occupied a large market. Figure 1.4 shows a statistics of worldwide market size of NFC from 2014 to 2024. It depicts current growth rate and future explosion of NFC Usages. Here, the curve is nearly exponentially increasing which implies the rapid growth rate of NFC applications and users.

1.1.1 Underlying Mechanism

Active-Passive communication of NFC works in half-duplex mode. In half-duplex mode, NFC technology enables two electronic devices (one of them typically portable like a smartphone

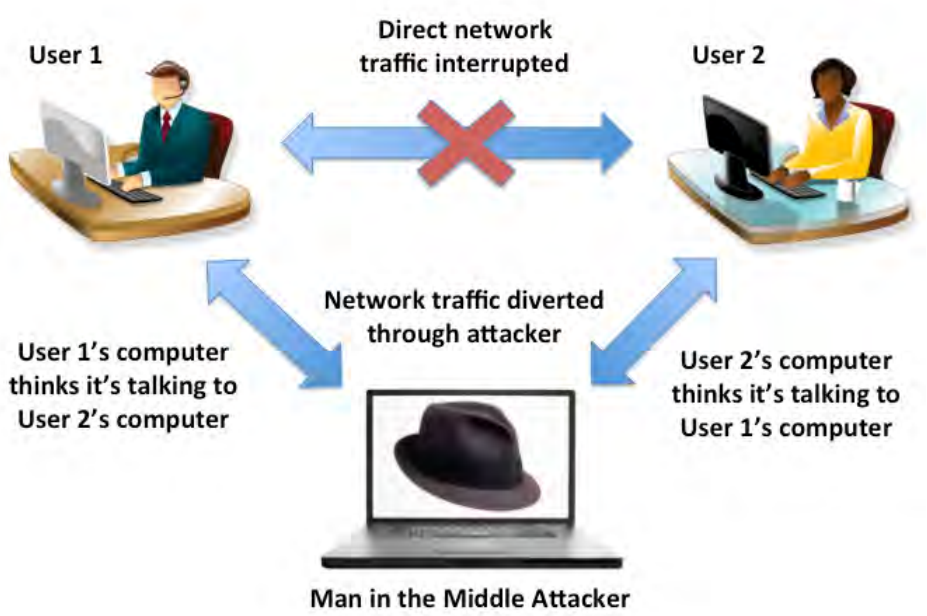


Figure 1.2: Traditional Man-in-the-Middle attack [41]

or a credit card and another one is possibly a reader fixed at a particular place) to establish communication with each other through bringing both devices in very close proximity (within <10 cm of each other). NFC devices can be of two types, namely active and passive, based on whether or not the devices own a power supply. An active device generally possesses a chip connected with a copper-wire coil. When this device is powered on, the coil generates a magnetic field to establish communications. A passive NFC device, on the other hand, does not have its own power supply. When a passive device comes close enough to an active device, due to electromagnetic induction, the coil of the passive device gets powered allowing communication as shown in Figure 1.5.

1.1.2 Applications of NFC Technology

The most critical applications of NFC technology today are in contactless payment systems generally used in smart debitcards or creditcards. These cards are allowed for offline transactions containing all information (for example, credit limit) needed for such transactions. These cards are basically used for payment of goods and services in NFC enabled PoS machines. These cards may also be used to pay for bus and train rides. However, contactless cards are not used in ATM for money withdrawal, as it requires the card to remain in communication

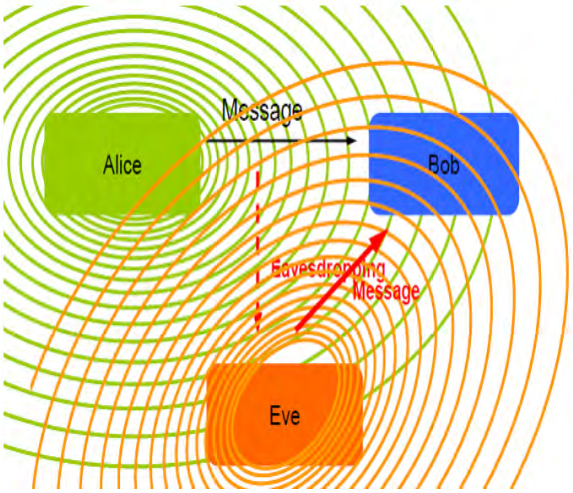


Figure 1.3: Messages will be destroyed due to non-aligned RF fields of Alice and Eve [1]

with the terminal for a period of time for online verification of the transaction. Contactless cards typically follow the Active-Passive model, where the active device is a reader (used by the merchant), while the passive device is the smartcard (presented by the user). They alternatively communicate in half-duplex mode following established protocols. Other applications such as sharing contacts, photos, videos, or files between NFC devices are also there, where both devices (e.g., smartphones) are active and can communicate in full duplex mode. Needless to say, the contactless payment applications (and even others sometimes) are highly security sensitive, with incentives for adversaries to compromise their operations.

1.2 Motivation

As of today, it is generally believed that with NFC technology, since the communications are held in close proximity between devices, the feasibility of unintentional data transfers is low. Nevertheless, to combat attacks, the notion of a Secure Element (essentially a chip) to enable a secure memory and execution environment is integrated within NFC devices. The secure element is a dynamic environment wherein application code and application data can be securely stored and administered, while enabling secure execution of applications. The secure element resides in highly secure crypto chips that also provide functions to encrypt, decrypt, and sign data packets.

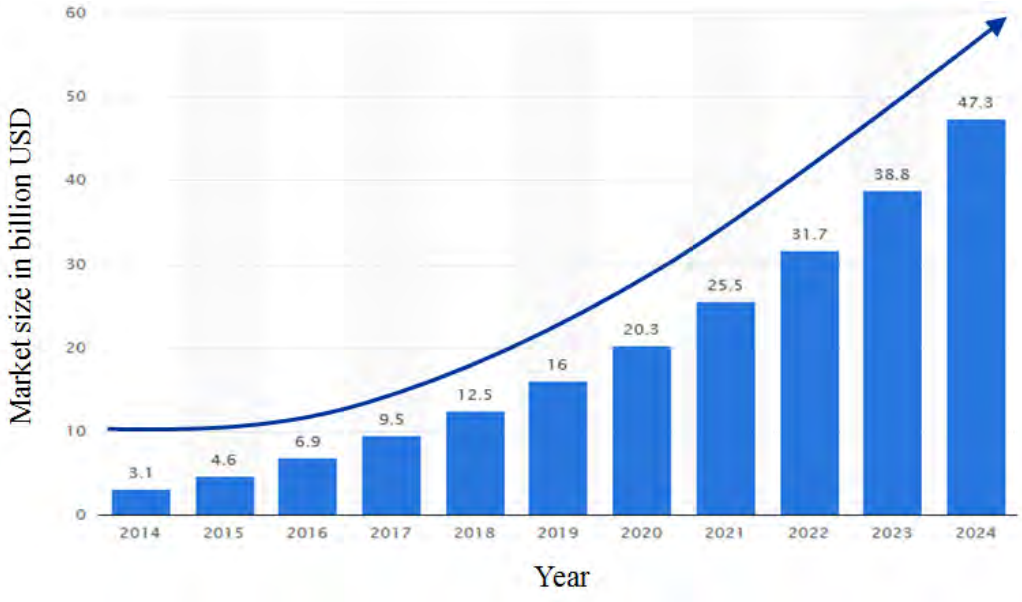


Figure 1.4: Future explosion of NFC usages [8]

Needless to say, existing designs do provide a high degree of confidentiality and integrity for NFC communications considering different attacks (some of them are described in Chapter 2). However, one potentially dangerous attack that has not been considered yet in this realm is Man-in-the-Middle attack. In this thesis, we demonstrate practical feasibility of getting into the middle of NFC communications with malicious intent. Therefore, we are motivated to explore a new vulnerability in this realm.

After getting motivated to demonstrate practical feasibility of MITM attacks over NFC, we develop a hardware prototype using three critical components: NFC shield with antenna, Arduino Uno board containing ATmega328 micro controller, and mifare classic tags. This prototype will act as our attacker module. In this phase, we need to identify a vulnerability in the existing contactless payment protocol in order that our attacker module can read and modify the original message exploiting that vulnerability. Next, a mechanism will be presented demonstrating how our proposed MITM attack can compromise fidelity of a financial transaction executed between two entities using a state-of-the-art protocol for contactless payments.

After successfully establishing the attack, we focus on its defense mechanism. We conduct rigorous experimental studies to reveal different aspects of the MITM attacks in NFC communications. Based on the studies, we devise a defense mechanism for our proposed

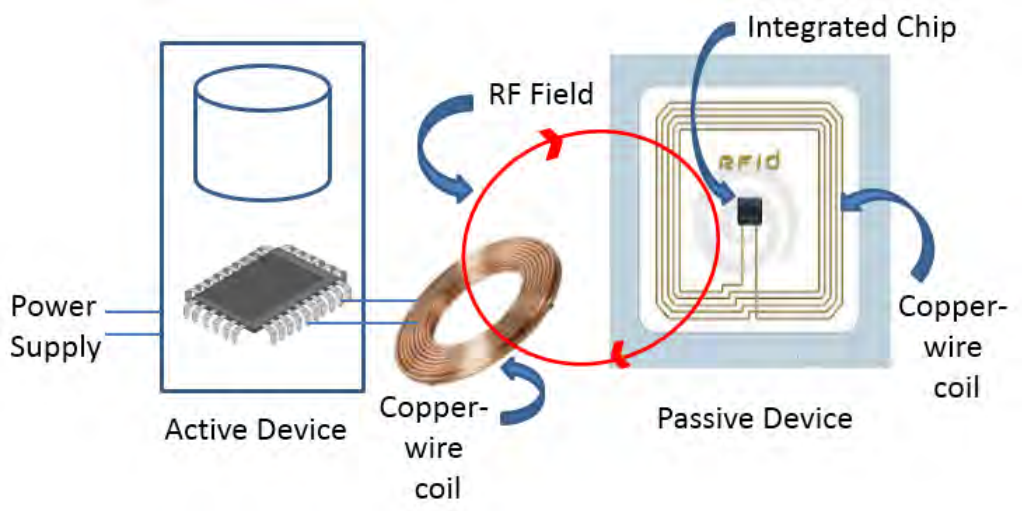


Figure 1.5: Working mechanism of NFC devices

MITM attack. The mechanism is devised in such a way that it does not demand any extra cost for additional hardware either at the user end or at the service provider end. We evaluate performance of our proposed mechanism and compare it against existing alternative countermeasures in terms of accuracy and cost analysis.

1.3 Our Contributions

Based on our work, we make the following set of contributions:

- We establish MITM attack over NFC, that was supposed to be impossible [1], [2] due to close proximity between devices and electromagnetic fundamentals. While still bounded by these physical constraints, we establish the attack between an active NFC device and a passive NFC device in the context of smartcard payments.

- We demonstrate how our proposed MITM attack compromises fidelity of a financial transaction executed between two entities using a state-of-the-art protocol for contactless payments. Here, we show how a card holder can juggle the merchant and bank exploiting the NFC architecture and flaw of contactless payment protocol. Besides, we show how an attacker can bypass all checks creating an illusion that transaction is being performed with a legal card. In such a case, both merchant and bank can be victim here.

- We analyze performances of the existing countermeasures in preventing the proposed attack. However, we could not identify a suitable solution which is able to defend the attack successfully without demanding extra cost for additional hardware.
- We devise a countermeasure for our newly-established attack. Our proposed countermeasure does not demand any additional hardware to be integrated with the NFC devices.
- We present effectiveness of our approach in preventing the attack through real experimentation. According to our experimental analysis, success rate of our proposed mechanism in detecting the attack is 100%.

The rest of the book is organized in the following way. In Chapter 2, we will show the background and related research studies. After that in Chapter 3, we will present the formal attack model and underlying physical fundamentals of our attack model. Subsequently, we will present how our MITM attack can compromise the fidelity of a financial transaction when executed between two entities using a state-of-the-art protocol for contactless payments. To specify the victims of our proposed attack, we will present some probable scenarios in Chapter 4.1. In Chapter 5, we will present real tested experimental results to devise a mechanism for defending against the attack. In Chapter 6, we will present our defense mechanism to guard this attack. After that, we will have a short conclusion including the future possible research directions.

Chapter 2

Related Work

Though near field communication technology is developed for secure data transfer, this technology is being hacked in different modes in recent times. This chapter has been organized listing several attacks that have been performed/demonstrated on such systems in recent years. We now present briefly an overview of important work related to security of NFC devices and their communications, while also highlighting the novelty of our work in this thesis.

2.1 Replay Attack

Common attacks over NFC that have already been disclosed are eavesdropping, Denial-of-Service (DoS), different forms of relay attack and phishing by social engineering. However, the mostly noticed attack in this realm is "Replay Attack" (also called the "Relay and Ghost Attack" or "Mafia Attack") [6], where an adversary reads a tag of a benign user by using a malicious reader device without the concern of its owner. Then, he/ she relays the tag information to a card emulator with which the adversary gets access of a secured place or performs a transaction (see Figure 2.1).

Figure 2.1: Replay attack

2.2 The Differences of Our Proposed MITM Attack from Replay Attacks

Our proposed attack may resemble the well known "Replay Attack", however, there are clear differences. The Replay Attack is one where the reader is typically malicious and it simply relays the contents of a benign tag (e.g., a smartcard) to a malicious entity to enable a fake transaction through compromising the *original card*. As we elaborate below, such attacks can be mitigated using dynamically changing crypto solutions, or location based approaches that attempt to physically tie a card and a reader at a particular location for approving a transaction [6, 10]. On the contrary, our MITM attack enables an attacker module to be physically present in the same environment to collude with the *original card*. It is thus equipped with the ability to read all communications between the reader and the *original card* from start to finish. Solutions proposed to combat Replay Attacks are not effective for our MITM attack proposed in this thesis.

In a research study [17], it is shown that PKES systems are vulnerable to relay attacks. They demonstrate that car can be opened and started even if the key is physically located far from the car. This corresponds to the scenario where the key is e.g., in the owners pocket in the supermarket, and the car is at the supermarket parking. As an immediate countermeasure they propose to shield the key (fob) of the car with a protective metallic element, i.e., creating a Faraday cage around the key. Thus, to prevent the communication between the key and the

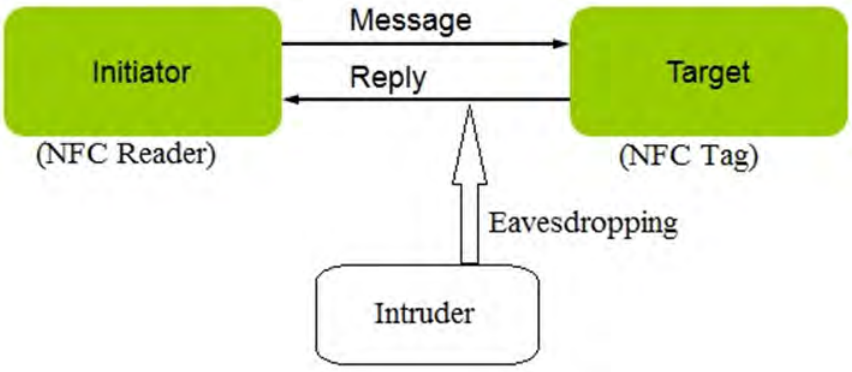


Figure 2.2: Eavesdropping over NFC communication

car, except when the owner wants to unlock the car. Similar countermeasures are proposed to block the possibility of remote reading of RFID tags embedded in e-passports. For providing better usability, they propose to use a button at user end to consciously enable or disable the communication. In our proposed attack, user himself can be malicious (we will illustrate later in Chapter 3) thus, this solution does not work in our case.

According to these countermeasures, initiatives should be taken by the user. However, possibility of the user being an adversary is not explored here. In our work, we describe a scenario where user is malicious and conduct transaction interchangeably using his/ her original and malicious card. Another countermeasure they propose is to disable the active wireless communication abilities of the key by removing the battery from the key that powers the radio signals. In our case, since we are working with passive tag, this solution is not applicable to defend against our attack.

To protect from the relay attack, work in [36] proposes to add a second form of authentication such as a password, a PIN or biometric information. However, they agree that this requirement will definitely eliminate the convenience and advantages of RFID or NFC Communication.

2.3 Other Attacks over NFC and Their Solutions

Near field communication systems have been hacked in different modes in recent times. In this section, we list some attacks that have been experienced to date, on such systems and analyze

their countermeasures to check whether they are applicable in our case.

2.3.1 Crypto Based Solutions

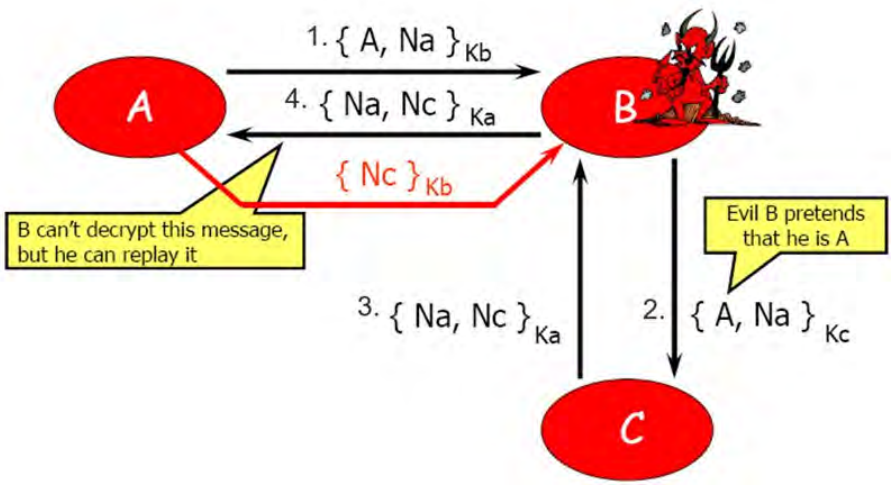
Many attacks over NFC can be mitigated using crypto based solutions as in [1], where it is shown that eavesdropping, data corruption, data modification, and data insertion can be mitigated by establishing a secure channel between the devices with a shared secret key. In most solutions proposed today [1, 2, 3], dynamically changing session keys are recommended to secure the channel between sender and receiver. The work in [3] also shows how a combination of AES encryption [4] and Deffie-Hellman key exchange [5] scheme can be used to prevent data modification, and eavesdropping over NFC. In [1], more innovative approaches are proposed for NFC specific key agreement mechanism. The idea is to synchronize the bits, amplitudes, as well as phases of RF signals randomly generated by two devices. Once they are synchronized, the devices communicate with exactly the same amplitude and phases as secret keys. However, these techniques are also not effective against our MITM attack due to collusion between the *original card* and the malicious *MITM card*.

Many attacks including Man-in-the-middle and Replay Attack can be defended using nonce[1]. Figure 2.3(a) presents a type of MITM attack [18] where evil agent B tricks honest A into revealing C's secret value N_c. i.e., C is convinced that he is talking with A. Working mechanism of this attack is similar to the attack we are working with. However, the defense mechanism of this attack [18] (shown in Figure 2.3(b)) is not applicable in our case. Because, in this case, both parties need to generate nonce which is difficult for low-cost passive tag.

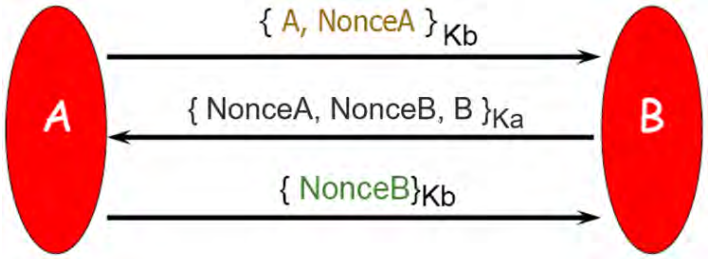
2.3.2 Location Centric Approaches

Another security scheme for NFC communications is called Tap-Tap and Pay [10], where the user of a valid card taps the reader a specific number of times with the card. Then, the accelerometer responses of the card and the reader are sent in real-time to a server along with

[1] A random value within a range, which must be unique for each communication. Therefore, by only checking whether the current value has already been received earlier from this sender, Replay Attacks can easily be detected.



(a) A type of Man-in-the-middle Attack



(b) Fixing the attack using nonce

Figure 2.3: Man-in-the-middle attack and nonce based solution [18]

the time stamps, wherein the server will determine whether they are correlated. If so, then the transaction is approved, and it is rejected otherwise. This technique can mitigate replay attacks. Additionally, a study [6] proposes that a card should be unlocked only when it is in an appropriate (pre-specified) location. These approaches does not work for defending against our attack, since the *original card* and *MITM card* in our attack model are co-located next to each other and colluding as well.

2.3.3 Approaches Leveraging Physical Unclonability of a Tag

This is an interesting approach that leverages unclonability of components of electronic circuitry during fabrication. Briefly, a physically unclonable function (PUF) is a physical entity that is embodied in a physical electronic microstructure that is easy to evaluate, however, hard to predict or clone. In this respect a PUF is the hardware analog of a one-way function [15].

Figure 2.4: Unclonability of components of electronic circuitry

Approaches leveraging PUFs have been used for challenge-response based authentication and also dynamic key generation and sharing in RFID/ NFC based communications. The standard approach is where the more secure and power enabled reader has prior knowledge about the unique properties of the tag that are then challenged and verified at run-time [16]. However, PUF based designs are complex to implement, and furthermore, since being present in between legitimate reader and tag, our attacker module can actively communicate with both ends. Thus, it is capable of relaying challenge-response to both legitimate tag and reader.

To further clarify, consider Figure 2.5. Here, our attacker module consists of a malicious tag T_{mal}, a malicious reader R_{mal}, and a malicious writer W_{mal}. This module receives challenge from the legitimate reader R_{leg} and relays the challenge to the legitimate tag T_{leg}. Then, after receiving response from T_{leg}, it relays back this response to the R_{leg}. It is possible to do so at real-time since our attacker module is co-located and colluding with the legitimate tag. Thus, PUF based solutions fail to prevent our proposed attack.

2.3.4 Approaches Leveraging NDEF Fuzzing

Besides, the study in [29], proposes an approach for security testing of NFC-enabled mobile phone. This approach is concerned with both NFC subsystem and software components that can be controlled through the NFC interface. This approach adopts fuzzing some fields of the NDEF format such as length of fields, type, ID, etc. This approach is able to detect multiple unknown vulnerabilities of NFC-enabled mobile phones through the adoption of NDEF

Figure 2.5: Process of bypassing PUF based mechanism

format. In our proposed MITM attack, the attacker also adopts the conventional NDEF format. Therefore, no abnormality is detected in the NDEF message format and the approach presented in [29] can not guard it.

2.4 The Significance and Novelty of Our Study

Our work in this thesis is important because MITM attacks have simply not been investigated in NFC communications because they are considered unlikely in practice [1, 2, 3]. Furthermore, existing solutions proposed for other security vulnerabilities in the NFC literature (that are highlighted above) are ineffective to prevent our proposed MITM attack. State-of-the-art crypto solutions does not work simply because the *MITM card* is present and listening to all communications (including secret keys) between the *original card* (with which the *MITM card* colludes) and the reader from start to finish. Location based approaches obviously fail because the malicious tag is physically close to the original tag and the reader. PUF based approaches are very challenging since being in the middle, the adversary device is listening all communication (including keys) between the original tag and the reader. The significance and novelty of our work in this thesis is demonstrating the practical feasibility of MITM attacks over NFC, leveraging the attack to compromise contactless payment protocols in manner that existing approaches cannot defend against.

Since existing solutions are not effective in defending our proposed attack, we devise a mechanism to combat the attack. Our mechanism is designed in such a way that it does not demand extra cost for additional hardware to be incorporated with the existing system.

Chapter 3

Proposed Attack Models and Applicability in Contactless Payments

We now present our MITM attack over NFC communications. First, we present the formal attack model. Then, we present the physical form-factor of our proposed design. Subsequently, we present how our MITM attack can compromise the fidelity of a financial transaction when executed between two entities using a state-of-the-art protocol for contactless payments.

3.1 The Formal Attack Model

Our proposed attack model is one where the user/ owner of an NFC-enabled smartcard is malicious. The malicious user (also known as adversary) possesses two smartcards, one called as the *original card*, and the other one called as the *MITM card*. It is important to note that the *MITM card* is one that is a clone of another valid card issued by a bank, however, whose details are exposed (possibly via skimming) by the user. The reader/ server is assumed to be benign. The goal of the adversary is to conduct NFC-enabled communications with the reader using the *original card* and the *MITM card* interchangeably during a single transaction with the motivation to fool the reader (e.g., a merchant).

To do so, two things must happen. First, the adversary must first be able to emplace an *MITM card* in between the *original card* and the reader throughout the communication between them, wherein the *MITM card* must be able to read all communication between the *original*

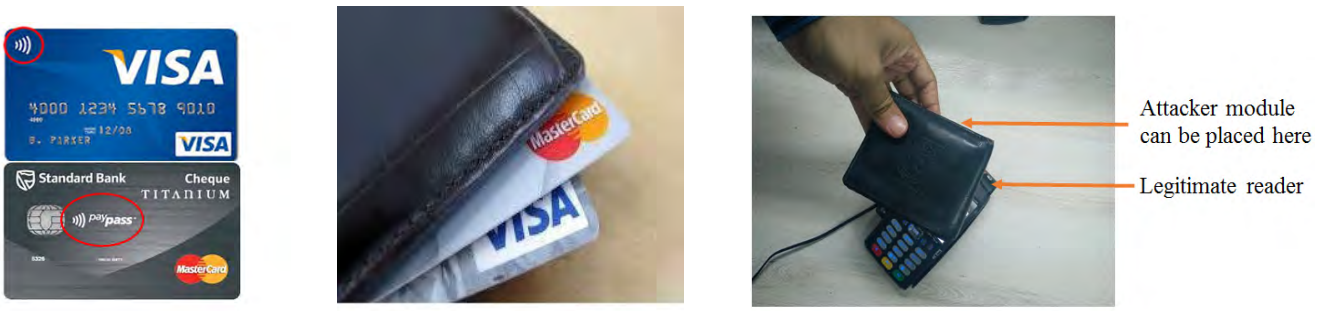
Figure 3.1: Man-in-the-middle (MITM) attack over NFC

card and the reader, while also being able to physically communicate with both parties (see Figure 3.1). This is an engineering challenge. Second, the adversary must be able to exploit a vulnerability in existing contactless payment protocols by intelligently manipulating which smartcard (between the *original card* and *MITM card*) communicates with the reader and when, so that the reader is victimized. This is an algorithmic challenge. In the following, we address both in detail.

3.2 Physical Form-factor of Our Proposed Design

An important consideration here is the physical form-factor of our proposed design. We believe that our proposed MITM attacker module can be easily designed in the form of a regular commercial smartcard with state-of-the-art engineering designs. As such, the entire attacker module can be easily emplaced in a wallet adjacent to another card. This is because, the NFC shield with arduino board in our design presented in Chapter 5 is used for programming only. Once programmed, the NFC shield can be replaced with the microcontroller and the antenna. The dimension of the antenna is 30.48mm $\times$ 27.94mm $\times$ 0.5mm. Thus when integrated with a tag, the resulting dimensions of the MITM module is comparable to typical smartcards, which is 85.6mm $\times$ 53.98mm $\times$ 0.76mm (defined by ISO 7816). Thus, it is possible to accommodate the entire attacker module within 2mm to 3mm width, which makes our MITM attacker practically invisible in a wallet where people use to keep their debit, credit, or loyalty cards. People hardly bring out their RF cards from wallet for doing any transaction. Rather, they hold their wallet in-front of the reader since radio frequency can pass leather, rexine, or cloth [21]. Figure 3.2 presents this typical usages scenario, which we propose to exploit with our MITM attack.

In recent days, debit, credit, or loyalty cards have started evolving to contactless cards using NFC [22]. Such contactless cards utilize a specified protocol for their transaction. To enable



(a) NFC-enabled payment cards from different parties

(b) Multiple cards usually kept in a wallet

(c) Holding wallet in front of a PoS machine while making a payment

Figure 3.2: The feasibility of our MITM attack module being invisible in a Wallet

MITM attack successfully, we must breach the protocol. Next, we present how we breach it through our proposed methodology.

3.3 Details on Contactless Payment Protocol

We are ready to present discussions on how the above attack setup can be practically leveraged by a malicious user to fool a merchant in the domain of contactless payment. Before, we do that, we present in Figure 3.2, an illustration of how the smartcards that employ NFC technologies look like. With our implementation presented below, we can see that it is simple to invisibly emplace the *MITM card* between the *original card* and the reader. How the presence of these two cards creates an attack scenario is presented next.

Contactless payment protocol [23] is based on the traditional contact EMV transaction protocols [24, 25] with few exceptions. Briefly, EMV (Europay, MasterCard, and Visa) is a technical standard for smart payment cards, payment terminals and automated teller machines that accept them. EMV cards are smartcards (also called chip cards or IC cards) that store data on integrated circuits in addition to magnetic stripes (for backward compatibility). Clearly, a critical goal of the protocol is to ensure secure communication between the terminal and the card consuming minimal amount of time.

The current EMV protocol can be split into three phases [26]: 1. Card authentication, 2. Cardholder verification, and 3. Transaction authorization. Contactless transaction skips the second phase since offline Personal Identification Number (PIN) is typically not supported

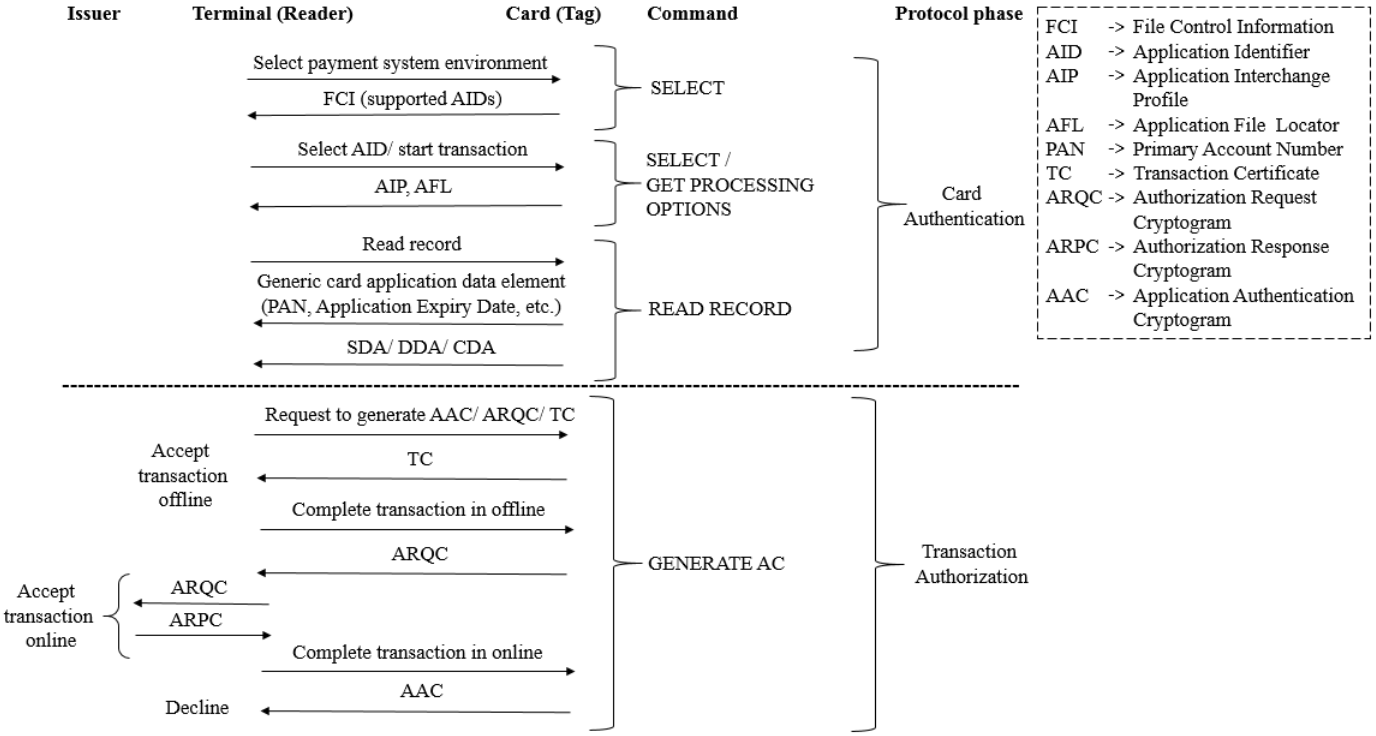


Figure 3.3: Complete mechanism of contactless payment protocol [27, 28]

here due to the security vulnerabilities in terms of eavesdropping to extract the PIN. Besides, it is practically difficult to ask card holders to enter a PIN while holding a card in-front of the terminal [28]. While selecting transaction instances, a PIN could be made mandatory, for the most part it is not, and therefore, in general, only two phases (Card Authentication and Transaction Authorization) are involved in contactless payment system. Figure 3.3 depicts the complete mechanism of the contactless payment protocol. We briefly elaborate the phases next.

3.3.1 Card Authentication

Both terminal (reader) and card (tag) may support multiple sensitive applications such as Payment System Environment (PSE) [27], Proximity System Environment (PPSE) [28], Debit/ Credit card, etc., each of which has different mechanisms to authenticate the card. To do so, the terminal is allowed to select an efficient payment environment using SELECT command. The card responds with the File Control Information (FCI) containing the list of supported applications (AIDs), which indicate whether EMV or Mag-Stripe mode is supported by the

card. Then, the terminal selects an AID (supported by both the card and the terminal) and starts a transaction using GET PROCESSING OPTIONS command. Subsequently, the terminal asks for a generic card application data element using READ RECORD command. With this step, the terminal can validate whether or not the corresponding card is approved for transaction.

The EMV standard defines the following three card authentication mechanisms [54]:

- **SDA (Static Data Authentication):** In SDA, the card provides signed static application data for verification. The signed data is verified by the terminal using public key authentication. This method is used to detect unauthorized cards or tampering. Cards identified as unauthorized in this step are rejected for payment. Otherwise, the terminal performs the next steps.

- **DDA (Dynamic Data Authentication):** DDA cards generate asymmetric cryptogram using a public/ private key pair along with a signature of the public key to prove its authenticity. DDA then involves a challenge-response mechanism, where the card proves its authenticity by signing a challenge chosen by the terminal using a private asymmetric key.

  However, in DDA, transaction authorization process is not tied with the authentication process, i.e., the terminal can authenticate a card but cannot verify that the subsequent transaction is actually carried out by the authenticated card.

- **CDA (Combined Data Authentication):** CDA repairs the deficiency of DDA. In CDA, the card digitally signs both card data and transaction data. Thus, CDA not only authenticates the card, but also authenticates the subsequent transaction.

Now, in contactless payments, SDA was typically adopted at the early stage till the year 2009 [20]. In this case, only cloning of a card was enough to make the system vulnerable and the risk of simple Replay attack remained. Later, after the year 2009, many contactless card providers mandate to use Dynamic Data Authentication (DDA) in contactless payment transactions [42]. Here, dynamic (vary in each transaction) data elements are used for the purpose of card authentication. Such systems overcome the vulnerability of Replay attack that can be enabled through cloning (as it was for SDA). However, the adoption of DDA exposes

vulnerability to our proposed MITM attack. Nonetheless, in case a more advanced version of DDA such as fDDA or CDA would be used in this regard, our proposed MITM attack would not be possible any more. This happens as such advanced authentication methods enable verification of transaction data in addition to card data. However, these advanced authentication methods are yet to be deployed in many geographical areas and by many brands (such as UnionPay [42]) in mass scale leaving numerous DDA-enabled cards already in operation leaving them vulnerable to our MITM attack.

3.3.2 Transaction Authorization

If a card is validated, then the terminal asks the card to generate a cryptographic MAC in addition to transaction related details such as amount, date, currency, country, etc., as parameters with the GENERATE AC command. Here, the terminal may request the card to generate TC, ARQC, or AAC (explained in Figure 3.3), which are essentially digital signatures of the financial transaction, generated via secret card keys (Card master key) and session keys [43]. Here, the card responds with TC if it allows offline transaction, returns ARQC if it forces the transaction to be online or returns AAC if it rejects the transaction.

Typically, ARQC is preferred by the terminal since any fraud can be detected at run-time using it. Once the ARQC is received from the card, the terminal forwards the ARQC to the issuer bank of the card to get an approval or rejection for the transaction. In this phase, issuer may check the credit limit and status of the card (whether it is stolen or lost marked). Issuer may also have Fraud Management System where different risk-based rules or machine learning algorithms can be used before approving the transaction. If everything seems to be perfect, issuer of the card approves the transaction by sending an ARPC to the terminal [44]. These steps also ensure two aspects:

- The financial message (amount, currency, date, etc.) is originated from the source that it claims to be from, and
- Content of the message is not altered.

Generating ARQC involves the following steps [56]:

- **Card and Session Key Derivation** To generate ARQC for a particular transaction, two keys are required: the first key is called Card Key and the second key is called Session Key. The Card Key is unique to the card and the Session Key is unique to the transaction.

- **Data Preparation** In this step, input data is prepared for ARQC generation in parallel to the key derivation. Which EMV tags are concatenated to prepare this input data depends on different EMV schemes (such as M/Chip and Visa).

- **ARQC Generation** Finally, once the Session Key and Input Data are ready, ARQC is generated by encrypting the Input Data using the Session Key. Thus, ARQC should be unique for each transaction.

A critical fact to observe here is that the check for validity of a card to be processed by a particular terminal happens only in the Card Authentication Phase by performing SDA or DDA. Once the terminal decides that a card is validated, then in the next phase of Transaction Authorization, the issuer bank of the card will only validate if the card whose details are supplied by the terminal was indeed issued by the bank (along with financial details and card status to verify its integrity). In this phase, no checks are performed if the card is actually authentic for transaction in the particular terminal. The absence of redundancy in checking simplifies the overall protocol, and speeds up transactions, which is vital for contactless payment. However, this gain also presents a vulnerability, which our proposed MITM attack attempts to exploit as presented laater in this chapter.

3.4 Commonly Followed Card Acceptance Guidelines

Usually, banks provide terminals (readers) to merchants, and they allow all their issuing cards with the right BINs[1] to be processed in their terminals for free. These transactions are called *on-us* transaction. However, when a bank is willing to accept cards issued by other banks (Union Pay, Visa, MasterCard, American Express, etc.) in their terminal, then the bank that supplies the terminal is called acquirer. The acquirer bank should have an agreement with the

[1]Bank Identification Number (BIN) refers to first four to six digits of a card that indicates a specific card type of a specific Bank.

bank that issued the card. Different charge or commission may be fixed for different card types during these agreements. These transactions are called *off-us* transaction.

Figure 3.4 illustrates the flow of transaction for Visa cards (magnetic stripe and contact/ contactless chip cards) transaction [14]. Following parties are involved in an *off-us* transaction:

- **Cardholder** is an authorized user of a payment card.
- **Merchant** is an authorized business entity to accept cards for the payment of goods and services.
- **Acquirer** is a financial institution (i.e., a bank) who accepts and processes the cards on behalf of a merchant.
- **Card Issuer** is a financial institution (i.e., a bank) who issues the cards and contacts with its cardholder for billing and payment of transaction.
- **Payment Card Association** is a publicly traded corporation that mediates between issuer and acquirer.

In a contactless payment, first of all, cardholder holds a card in front of a reader (terminal) at merchant site. Here, as discussed above, card authentication is performed in offline which is nothing but checking a digital signature. Transaction authorization can also be performed in offline or card/ reader can force the transaction to be performed in online. In offline transaction, terminal itself is capable of verifying the transaction without communicating with the issuer. Since these transactions are more vulnerable, banks are less likely to allow offline transaction. In online authorization process, acquirer electronically sends the authorization request to the respective card association (Visa, MasterCard, UnionPay, American Express, etc.) who then routes the request to the issuer of the card. Issuer sends back a positive/ negative response to the acquirer through the respective card association. Acquirer forwards the response to the merchant. Merchant completes the transaction according to the received response and sends a message to the terminal.

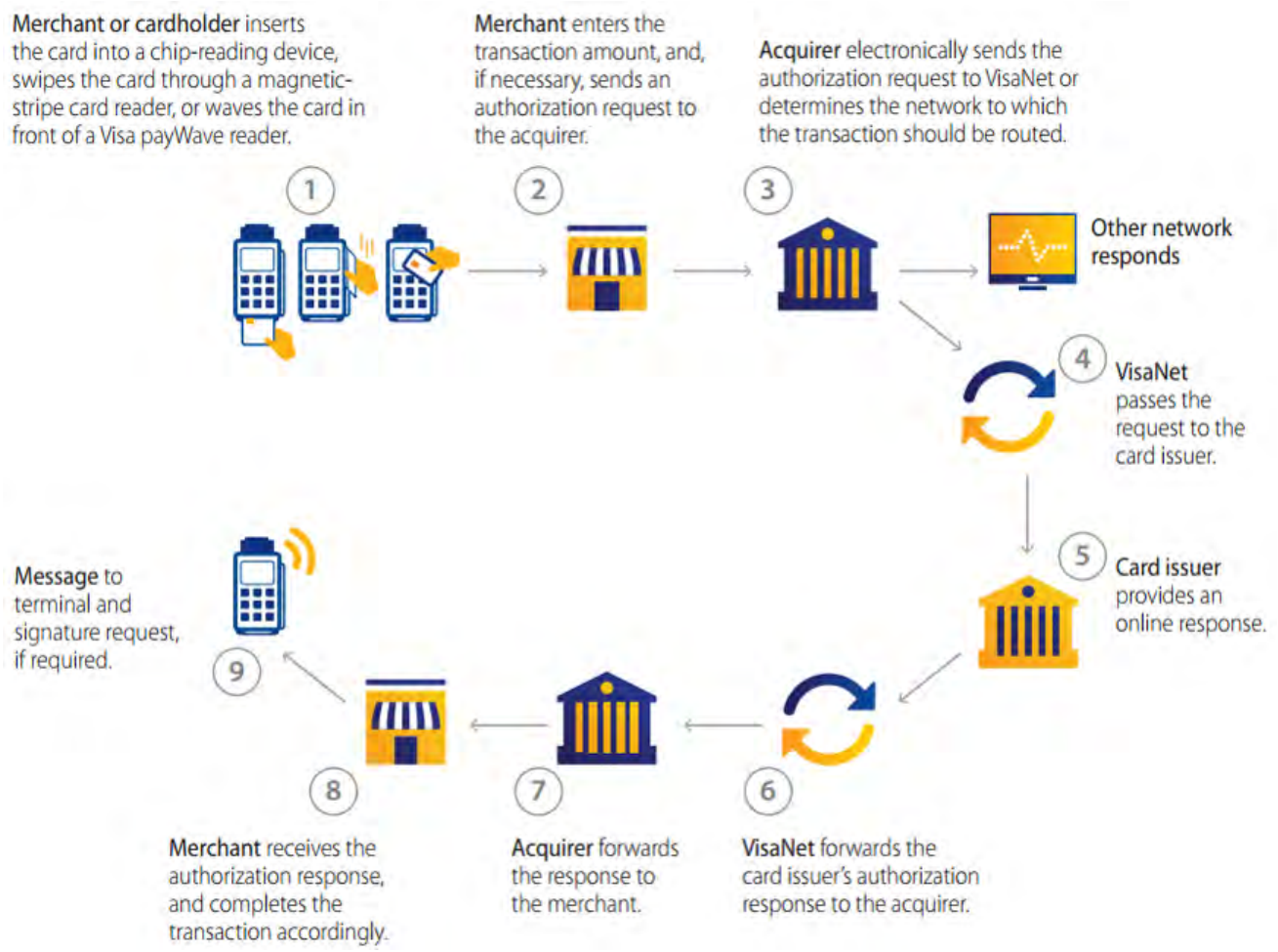


Figure 3.4: Online authorization process for credit or debit transactions [14]

3.5 Clearing and Settlement Process

Normally, an offline settlement is performed between merchant and acquirer, and then acquirer and issuer after several days of a transaction. This time may vary according to their agreement. During the clearing and settlement of a transaction, merchant submits the transaction information to the acquirer. Acquirer credits the merchants account and submits the transaction information to the issuer via respective card association claiming transaction amount from them. Card issuer checks the transaction and clears payment to the acquirer. Figure 3.5 shows the flow of settlement and clearing process for Visa credit cards.

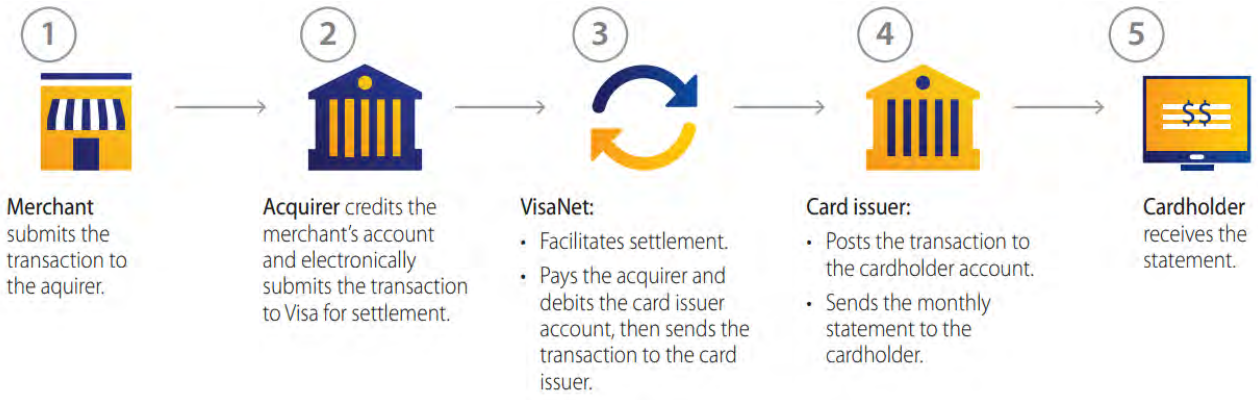


Figure 3.5: Process of clearing and settlement of a transaction [14]

3.6 Attack Model over the Payment Protocol

We now present details on how the presence of the MITM card can compromise the above protocol. Figure 3.6 depicts a way how the MITM attack can be incorporated with contactless payment protocol. Here, let a malicious user owns two cards. One of them will be accepted by the terminal (i.e., the *original card*). The other card is the *MITM card*, and is the clone of another valid card or a stolen card, which is legally issued by a bank. Note that the *MITM card* has been engineered by the malicious user using our designs presented in this thesis, and via skimming details of the valid card [7]. Note also that the *MITM card* is one that is not authorized for use at a particular terminal. In this context, we present a potential attack scenario.

First, the terminal initiates communication with the user. This is through the *MITM card* because it is an MITM between the *original card* and the terminal. Here, the payment environment is selected by the *MITM card*. When the terminal asks for generic card application data, the *MITM card* simply relays the request to the *original card*, receives a response, and relays it to the terminal. Since this data comes from the *original card*, the Card Authentication phase is successful using the right keys. Since the attacker just relays the messages, he/ she does not need to uncover any messages.

Once the Card Authentication phase gets completed, the *MITM card* does not need to communicate with the *original card* anymore. In this phase, since *MITM card* directly communicates with the terminal, it usually responds with TC to perform an offline transaction. If

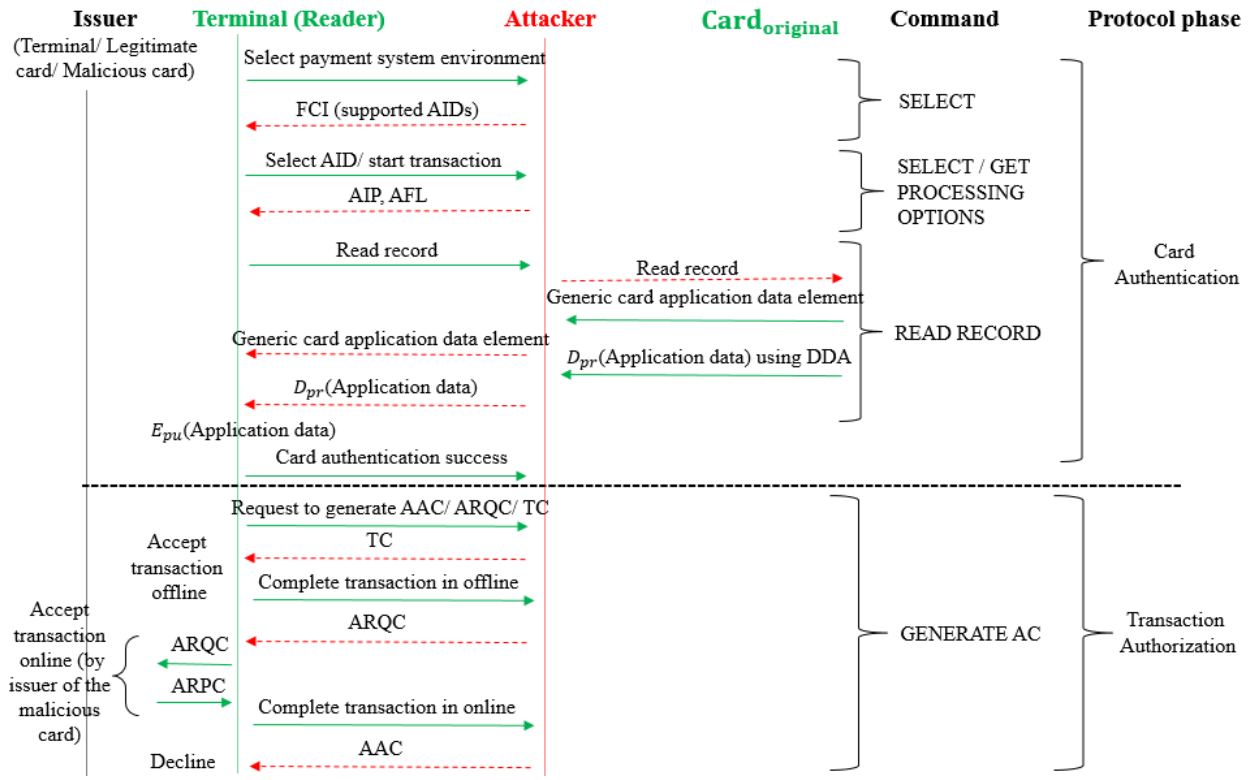


Figure 3.6: Attacker in between card and terminal performs card authentication using *original card* information, then transaction authorization using a fake card

the terminal supports offline transactions, then the attacker gets succeeded at this point. If the terminal does not support offline transactions or it decides to go for an online transaction, the attacker needs to respond with its ARQC to the terminal. The terminal then sends this cryptogram to the corresponding payment card association (e.g., UnionPay, Visa, MasterCard, American Express, etc.), which usually does not apply any verification, however, rather simply sends the cryptogram to the bank that issued the card for verifying the requested transaction details. Since the requesting card appears to be a card issued by a bank (as for being a clone of a valid card), the terminal is expected to receive a successful ARPC from the issuer bank (see Figure 3.6) for a successful transaction, which is then executed.

Note that, here, possibility of the attacker getting succeeded even after the online verification remains, as the verifying entity in this case is the issuer of the card cloned to the malicious card (can be different from the acquirer bank who provide the terminal) and the card information presented in Card Authentication phase is not used in this verification. Thus, the segregation of Card Authentication phase and Transaction Authorization phase in terms

of checking card data remains the source of vulnerability posed by our proposed MITM attack. This is applicable specifically owing to the conventional usage of DDA [42] as shown in Figure 3.6. The vulnerability can expose different stakeholders of the transaction as the ultimate victim under different cases. We analyze the cases in the next chapter.

Chapter 4

Attack-Victim Analysis

In this chapter, we present some probable scenarios to specify the potential victims of our proposed attack. Before discussing the scenarios, we point out our assumptions and considerations that lead in road to enabling our proposed attack. Note that our assumptions and considerations are based on the usage of DDA (not any of SDA, fDDA, or CDA) as already mentioned in the last chapter.

4.1 Assumptions and Considerations behind Enabling Our Proposed Attack

We adopt a set of assumptions and considerations to perform our attack-victim analysis. It is worth mentioning that our adopted assumptions and considerations are not the only cases where our proposed attack could be enabled. Even then, we present the assumptions and considerations to more vividly portray a few cases where our proposed attack exhibits substantial implications. The assumptions and considerations are as follows:

- An attacker can use a stolen/ lost card as the malicious card for performing **offline transactions**. Such usage of stolen/ lost cards can practically happen as already pointed in several happenings [47]. Such happenings could be overcome if the card status is checked during the Card Authentication phase. However, our proposed attack can go beyond it even the checking is incorporated in the Card Authentication phase, as our attack uses a valid card in this phase.

- An attacker can use Debit/ Credit cards interchangeably using our proposed attack in a terminal, which supports only a certain types (either Debit or Credit) of cards. Such restrictions of using a certain types of cards from a specific terminal can often be found in practice as per the contract between a merchant and its bank [48, 51, 52]. This scenario to be exploited by an attacker is applicable even for **online transactions**, as card data is generally checked only in the Card Authentication phase [46, 49, 50]. Besides, to the best of our knowledge, the card data (whether it is Debit or Credit) is not checked during the Transaction Authorization phase, as we are yet to find any publicly available content that mandates for it.

- Blocked cards (or not-supported cards) can be used as the malicious card for **online transactions**. We present this consideration from our field experience, as we have found that such blocking is often done by acquirer banks suspecting some cards as malicious (or for other reasons). In Appendix B, we include an email conversation with a bank of Canada, where it is told that they accept internationally issued VISA cards in most of the cases, which eventually means that they can reject it in some cases. Usually, the checks pertinent to the blocked cards is performed in the Card Authentication phase keeping the vulnerability exploited by our proposed attack. Note that such events of blocking some cards and checking them during the Card Authentication phase often retained confidential by the banks, as banks generally prefer to keep their security related events confidential [50].

Note that, in our study, we consider *off-us* transactions, where the acquirer bank and issuer bank are different. The reason behind such consideration is the fact that, for *off-us* transactions, offline settlements are performed between merchants and acquirers, and then acquirers and issuers after several days of the transactions. These happen irrespective of whether the transactions are initiated by debit or credit cards, which generally follow the same transaction authorization process as presented in the earlier section (Fig. 3.3). Accordingly, in case the transaction authorization process can be started after passing the Card Authentication phase, a transaction is expected to be completed as considered in our second consideration mentioned above.

Another worth mentioning aspect in our consideration is that owners of the valid card and

malicious card can vary in different scenarios. In case of the first and second consideration, the malicious card can be owned either by the attacker or by another innocent person (whose card is cloned or stolen). Here, for the first consideration, the activity of cloning or stealing could have already been reported in the bank. This must not have been done for the second consideration. Besides, for the third consideration, the malicious card is preferred to be owned by an innocent person (other than the attacker). This preference is based on the fact that the account pertinent to the malicious card could be charged in this case (will be elaborated in the next section).

On the other hand, in all the considerations, the valid card could be owned by the attacker. However, such owning involves the risk of being traced later as the valid card's information gets entered during the Card Authentication phase. If the attacker wants to overcome this risk, s/he can use yet another cloned or stolen card (yet to be reported) as the valid card. Here, the corresponding PIN needs to be known in case the PIN of the second card is needed to be entered for cardholder verification. Such PIN entering is *not* always mandatory for contactless payments [28, 53]. Besides, in real scenario, there exists a lot of PoS devices, which are unable to process PIN. As a result, a successful transaction may be performed without asking for PIN (as an evidence of this claim, in Appendix C, we include an email from an issuer bank to the acquirer bank mentioning a POS machine that is not PIN enable and experienced a fraud transaction). Thus, there retains the possibility of using the cloned or stolen card as the valid card even without knowing the PIN. It may be argued that what would be the point of using a second cloned or stolen card only in Card Authentication phase when they can be directly used in malicious transactions. Such an argument will be valid if the intended limit of malicious transaction is within the permitted limit of the second card. However, if limit of the intended malicious transaction crosses limit of the second card, the first malicious card can become handy permitting availability of its higher limit of transaction. Here, the first card cannot be used for the whole transaction as for not being accepted by the terminal in the Card Authentication phase as per our consideration.

Now, based on the above assumptions and considerations, we will discuss different cases focusing on potential victims of our attack model.

4.2 Potential Victims of Our Attack in Different Cases

We have already discussed earlier that a user himself can be malicious. Thus, a user can possess a stolen card with which he wants to make a malicious transaction. For conducting such fraud transactions, malicious users usually prefer PoS terminals that allow transaction authorization to be processed in offline. This happens as, in these cases, possibility of getting the attempted malicious transactions successful remains high. Here, if the acquirer bank or the transactions authorizing terminal remains ignorant about the stolen card, a successful transaction will be conducted with this card. However, if the acquirer bank is notified about the stolen card by issuer of the card, then the transactions authorizing terminal will reject this card in the Card Authentication phase. Here comes the requirement of our MITM attacker module. Now, consider the following scenario based on the first assumption.

Scenario #1:

i Attacker has a stolen card, which is already lost marked by both the issuer and the terminals that are capable to process transaction in offline.

ii Attacker wants to conduct offline transaction using this lost card. Such usage of stolen/ lost cards can practically happen as already pointed in several happenings [47]. Such happenings are usually overcome by checking the card status during the Card Authentication phase.

iii To pass this check, attacker uses a valid card (his own card or another lost card having low balance or limit) in Card Authentication phase, and the lost card in Transaction Authorization phase (as shown in Figure 3.6).

iv Thus, transaction is successfully performed interchangeably using a valid and a lost card.

v Now, during settlement, acquirer bank will clear payment with the merchant. However, when the acquirer bank will go for settlement with the issuer bank of the stolen card, it may deny to pay for this transaction (since it already notified about this stolen card).

vi If such happens, the acquirer bank will be the victim. In case the acquirer bank charges the merchant back for such happening, then the merchant will become the victim.

This whole scenario pertaining to our first consideration (presented in the earlier section) is depicted in Figure 4.1(a).

Next, in our second consideration, a terminal accepts only a certain types of cards (can be either Debit or Credit). Thus, a user can not generally perform transactions with other types of cards in this terminal except the acceptable ones. For example, if a terminal accepts only Debit cards, a user can not use a Credit card in the terminal. Here, Debit cards are differentiated from Credit cards using different ranges of Bank Identification Number (BIN). Terminal usually performs such checks on cards data in the Card Authentication phase [49, 50]. If this phase can be passed using the permitted types of cards (Debit cards in this case), transaction authorization can be performed even in online using other types of cards (can be a Credit card in this case). Consider the following scenario based on the second assumption.

Scenario #2:

i Attacker has a credit card (his own card or a cloned card). He wants to make a transaction using this card.

ii However, the terminal with which he is trying to make a transaction accepts only debit cards. Such restrictions are pragmatic. For example, [51] refers an agreement form between a merchant and Woodforest National Bank, a bank of United States, where an option of adopting only debit cards remains.

iii In case of debit only transactions, debit cards are differentiated from credit cards using specific BIN range. Such card data are usually checked in the Card Authentication phase [46, 49, 50].

iv To pass this phase, at first, an attacker uses a debit card (his own card or a cloned card) for card authentication, and then conduct transaction using the credit card. As the agreement was confined between the merchant and acquirer bank, issuer of the credit card will not apply any restriction for this specific terminal or card type. Thus, transaction authorization can be performed even in online in this case.

v During settlement, an issuer bank generally pays to the acquirer. However, in a later stage, owner of the credit card can claim back the money. Here, the owner can place a strong argument mentioning that the terminal performing the transaction does not support the type of card on which he is being charged. As the allowed types of cards for a merchant

is generally known to public, denying this fact is very difficult for both the acquirer and merchant.

vi The issuer may pay back the owner retracting the money from the acquirer. If it happens, then either the acquirer or the merchant will be the victim based on whether the acquirer retains the money to the merchant or it also retracts the money back from the merchant respectively.

Figure 4.1(b) depicts the whole scenario.

Finally, from our field experience, we have found that acquirer banks often block several cards (specially through blocking Bank Identification Numbers or BINs, which may be of local or foreign banks) suspecting them as sources of malicious transactions. List of such blocked cards is usually checked in the Card Authentication phase. In this scenario, a successful transaction can be performed interchangeably using an unblocked card first during the Card Authentication phase, and then a blocked card during the Transaction Authorization phase following our attack model. Following scenario is based on our third assumption.

Scenario #3:

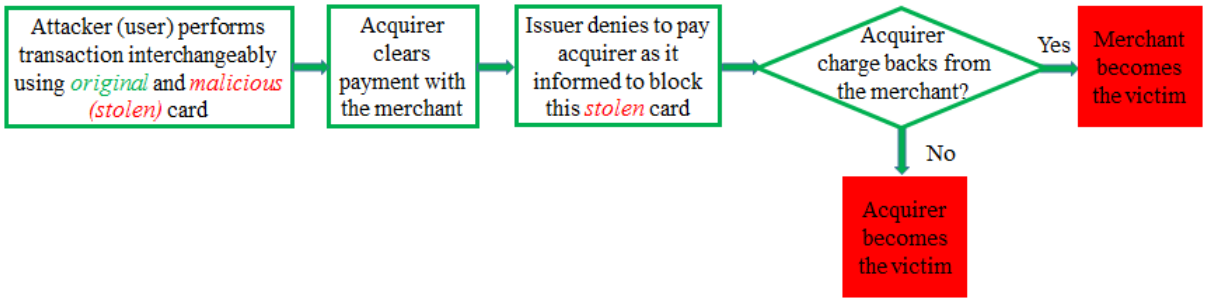
i Attacker has a card, which is blocked by the terminal suspecting it as sources of malicious transactions. Using this card, the attacker wants to make a transaction. List of such blocked cards is usually checked in the Card Authentication phase.

ii Attacker performs successful transaction interchangeably using an unblocked card first during the Card Authentication phase, and then a blocked card during the Transaction Authorization phase.

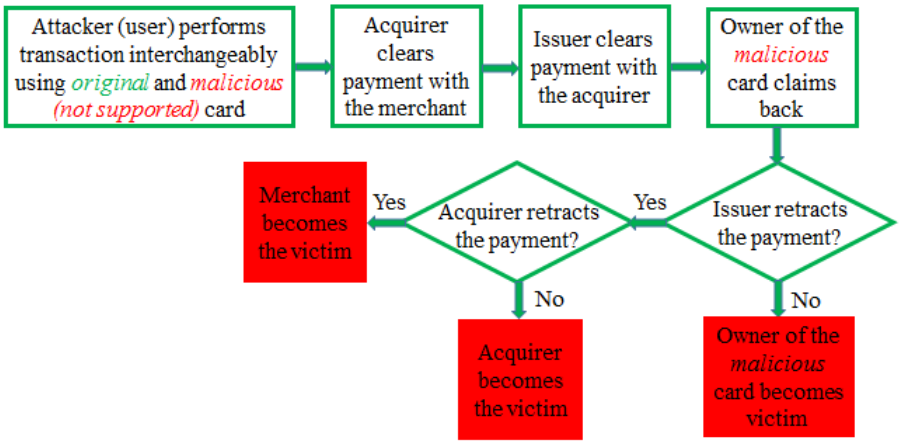
iii If this happens, the acquirer bank can claim money from the issuer bank for this transaction as the policy of such blocking is completely internal to the acquirer bank.

iv Afterwards, the acquirer can either retract the money back from the merchant or retain the transaction as it is based on the terms and conditions between acquirer and merchant along with frequency of happening such cases.

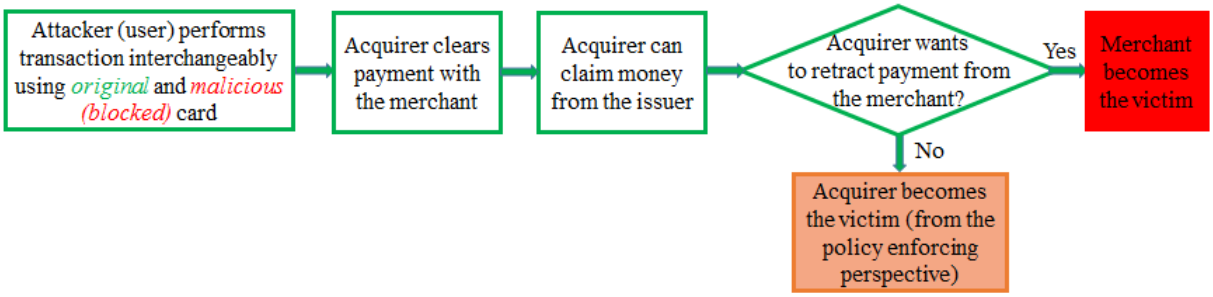
v If the retraction happens, the merchant will be the victim. In case the transaction remains



(a) Scenario when our attack enables a transaction with a stolen card (only in offline transactions)



(b) Scenario when our attack enables a transaction with a non-permitted type of card (applicable in both online and offline transactions)



(c) Scenario when our attack enables a transaction with an internally blocked card (applicable in both online and offline transactions)

Figure 4.1: Possible victims of MITM attack over contactless payment

as it is, the acquirer may not incur any financial loss, however, must fail in its policy enforcement becoming a victim from the policy enforcement perspective.

Figure 4.1(c) depicts the above scenario.

Table 4.1: Considerations and potential victims of our attack

Scenario	Attacker owns	Purpose	Victim
1	One lost card and one valid card	Performing transaction with the lost card	Acquirer bank or merchant
2	One debit and one credit card	Performing transaction with the credit card in a terminal, which accepts only debit cards	Acquirer bank or merchant or owner of the credit card
3	One blocked card and one unblocked card	Performing transaction with the blocked card	Acquirer bank or merchant

Note that, in several cases, if fraud transactions repeatedly occur with a specific merchant, corresponding merchant or acquirer bank may be subject to incremental chargebacks, settlement delay, termination of agreement, audit and imposition of fines, etc., [47]. Thus, the extent of being victimized by a merchant or an acquirer bank can be much severe than what we present above.

Table 4.1 summarizes the above scenarios explicitly mentioning our assumptions/ considerations and victims for each cases.

4.3 Clarifying Discussions on the Attack

Our MITM attack possesses different aspects that are practical to be implemented to the best of our knowledge. For example, the hardware required for enabling the attack (we will present in testbed settings later) is feasible for NFC communications in the current form factor. Besides, with simple sniffing and/or skimming techniques, a card can be perfectly cloned [45, 54, 55], which is capable of generating TC and ARQC similar to a valid card.

Note that the feasibility or appeal of our proposed attacker module may look of no use in case the original card uses SDA, as simple cloning and Replay attack should suffice here. While this is doable and will eliminate the need for our proposed separate MITM module, we believe our proposed module will offer more flexibility to an attacker even in doing so. This happens as, in case of adopting simple cloning, an attacker will need to have separate cloned cards for each of his legitimate cards. On the other hand, if he would use our proposed module, only a single attacker module will suffice to emulate each of his different legitimate cards. Nonetheless, our

proposed attacker module becomes a compulsion for MITM attack when the legitimated card is equipped with the capability of DDA, which is now pervasive in many parts of the world [42]. This attack is possible due to the design flaw in DDA, which solely authenticates the card rather than verifying that the subsequent transaction is actually carried out by that card or not [54].

Moreover, the scenarios we described in this chapter is only a few possible case studies of our proposed attack mechanism. There can be many more such cases that may be revealed in future. For example, the user can be benign and a third party can incorporate the attacker module within his wallet without his concern. Besides, it is possible that our MITM module can be in proximity to the terminal as a component designed to sniff cryptogram details of benign cards that could be used later for generating fake/ malicious transactions. In this manner, a terminal does not need to be tampered within, however, the station containing the terminal can still be forced in malicious transactions. We do not elaborate on this aspects in more detail, however, this is practical.

Another considerable aspect is that if terminals employ PUF based detection approaches (as presented in Chapter 2), the valid card must be present and not tampered with for a successful PUF based validation. Here, simple cloning of legitimate card's information will not suffice unless the cloned card can emulate the PUF in a similar way the legitimate card does. Additionally, it is worth mentioning that usage of our proposed MITM module does not remove the option of discarding it in case the attacker wants to use his legitimate card. This only needs to substantially dislocate the attacker module from the legitimated card while performing a transaction using the legitimate card. Furthermore, the close physical proximity and collusion between the *MITM card* and the *original card* mean that existing protocols proposed in the literature to defend against other attacks in NFC communications (presented earlier in Chapter 2) are not geared for defending against our MITM attack.

We also believe that with the wide popularity of NFC based applications in smart transportation cards, smart tolls, passport based entry systems, inventory tracking (e.g., medicines), new attacks are possible when adversaries leverage our designs presented in this research to launch MITM attacks with new modalities, and investigating these is part of our future study.

Finally, in certain cases, there may be much lower limits on transaction amounts that are

allowed to be conducted via contactless payments in order to provide faster financial services. While our attack is still feasible in such scenarios, this issue opens up a new spectrum of the cost-effectiveness to an attacker in engineering attacks for financial gains. This is yet another issue that could be potentially investigated from both an attack and defense perspective.

Chapter 5

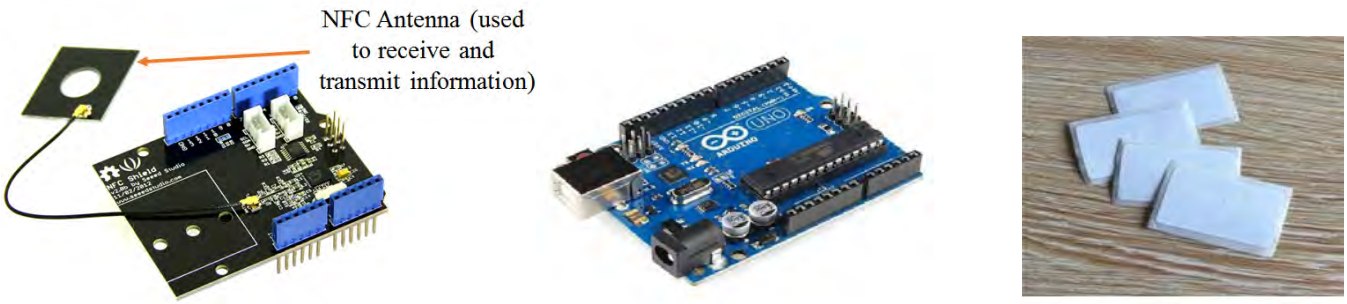
Real Deployment of Our MITM Attack

In this chapter, we present the experimental evolution of our MITM attack over NFC communications and findings of our experiments. First, we present the physical fundamentals of our MITM attack. Then, we present rigorous experimental analysis over the attack to reveal different aspects of the attack so that it can be used as a key metric to detect the attack.

5.1 The Physical Fundamentals

There are three critical components (shown in Figure 5.1) in our design of the MITM attack module. The first is NFC shield with antenna (Figure 5.1(a)) to transmit and receive information. In our setup, we use three NFC shields (v2.1) [11] as active devices, whose maximum effective communication range is 5 cm over a frequency of 13.56 MHz. Second is Arduino Uno boards [12] containing ATmega328 microcontroller (Figure 5.1(b)), which is used to make the shields programmable. The last component is the passive card. For this, we use MiFare Classic 1K cards (Figure 5.1(c)). Figure 5.2(a) shows the schematic view of our MITM attack where the *MITM card* (that is embedded with a reader, and a writer) resides between the *original card* and the reader. Figure 5.2(b) shows the detailed implementation setup.

To make the NFC shield operational, we stack the NFC shield on an Arduino development board and connect the board to a computer using a USB cable. The NFC shield can act as a reader or a writer depending on the instructions enabled in it. In both cases, when an NFC-



(a) NFC shield (b) Arduino Uno board (c) MiFare Classic 1K tag

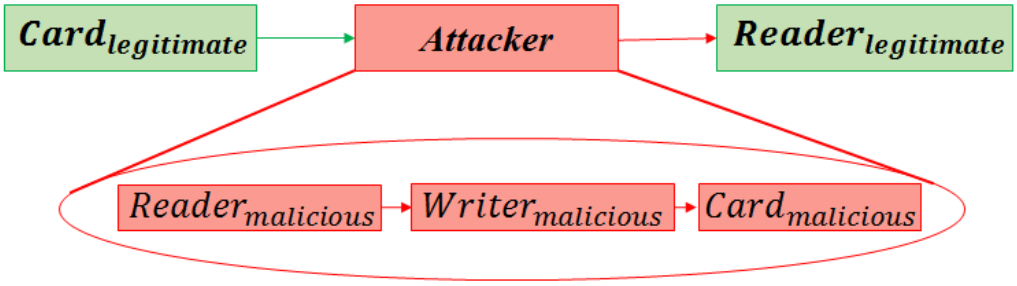
Figure 5.1: Device Specification

enabled card is held in-front of the antenna of a NFC shield, it can detect and communicate with the card.

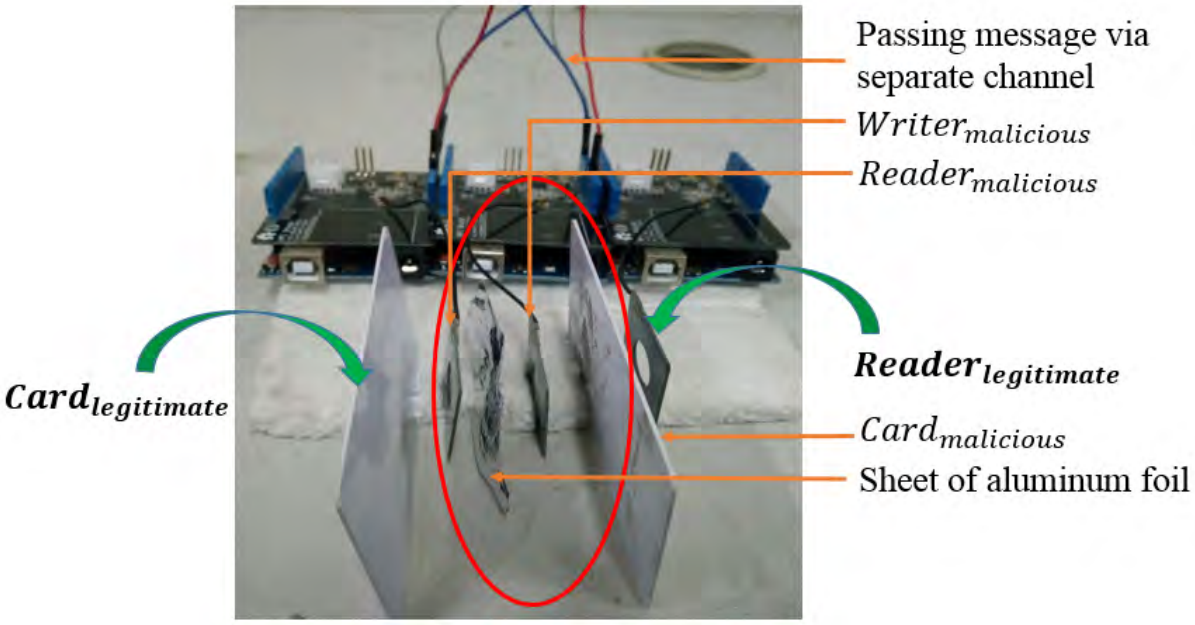
5.1.1 Settings of Real Deployment

In our experiment, the *MITM card* is placed in between the $Reader_{original}$ and $Card_{original}$. A sheet of aluminium foil is used to isolate $Reader_{malicious}$ and $Writer_{malicious}$ to avoid collision between their radio signals. They are connected via a separate channel (for example, wire in our case) to pass information. Here, three active and two passive devices are placed in passive-active-active-passive-active manner where the devices act as card-reader-writer-card-reader mode.

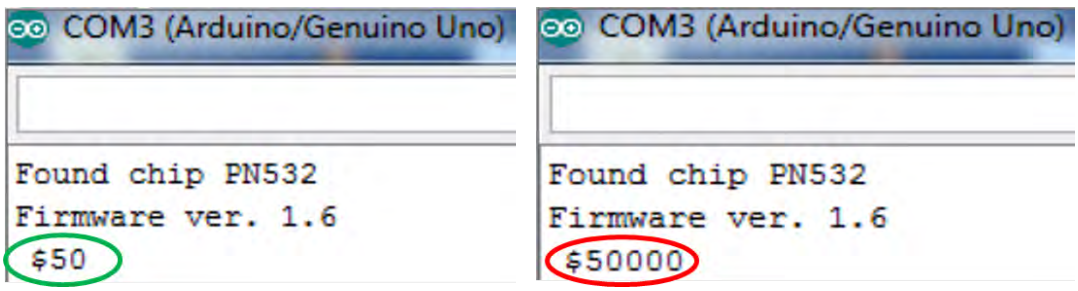
We now refer back to Figure 5.2(a) to illustrate how the MITM attack works in our setup. In the absence of the MITM attack, two-way communication is normal between the *original card* and the reader. Under attack, $Reader_{malicious}$, $Writer_{malicious}$, and $Card_{malicious}$ combinedly act as the MITM attacker. Here, $Reader_{malicious}$ reads any message from $Card_{original}$, modifies it (if needed), and sends the modified message to the $Writer_{malicious}$. Then, $Writer_{malicious}$ writes the information in $Card_{malicious}$. Once writing has been completed, $Writer_{malicious}$ needs to release the channel so that $Reader_{original}$ gets the channel free and can read $Card_{malicious}$. Therefore, when the original reader $Reader_{original}$ wants to read $Card_{original}$, it actually reads the attacker's card $Card_{malicious}$ which may contain a modified message. Here, since attacker is in the middle of the original reader and original card, he/ she can decide when and which message will be passed to the original reader. Such messages can be anything from payment



(a) High-level view

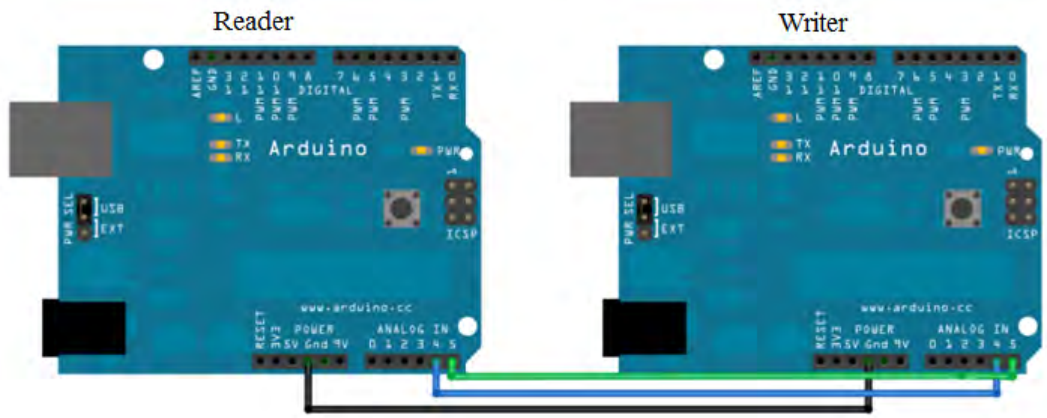


(b) Real setup

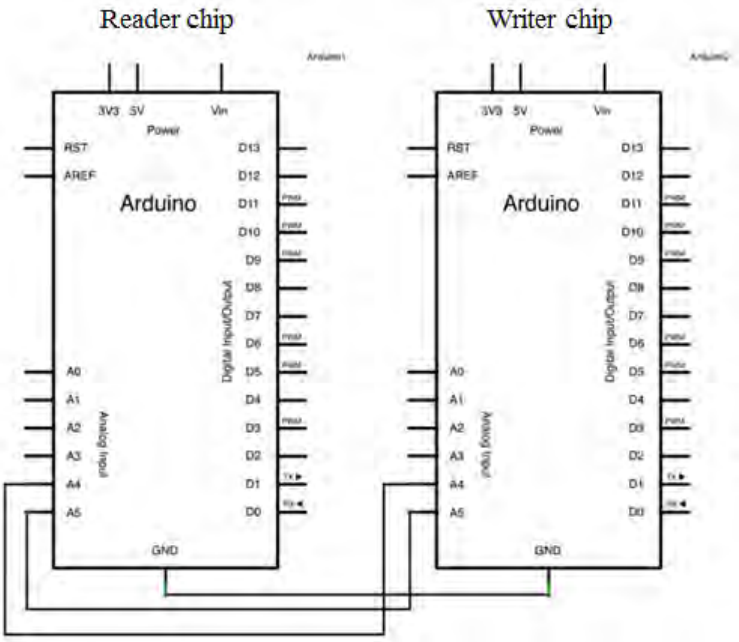


(c) Screenshots of a legitimate transmission and a transmission under demo attack

Figure 5.2: Experimental setup and demonstration of MITM



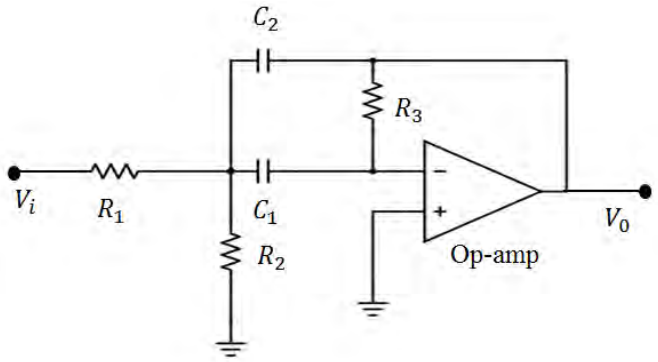
(a) Connection between reader and writer



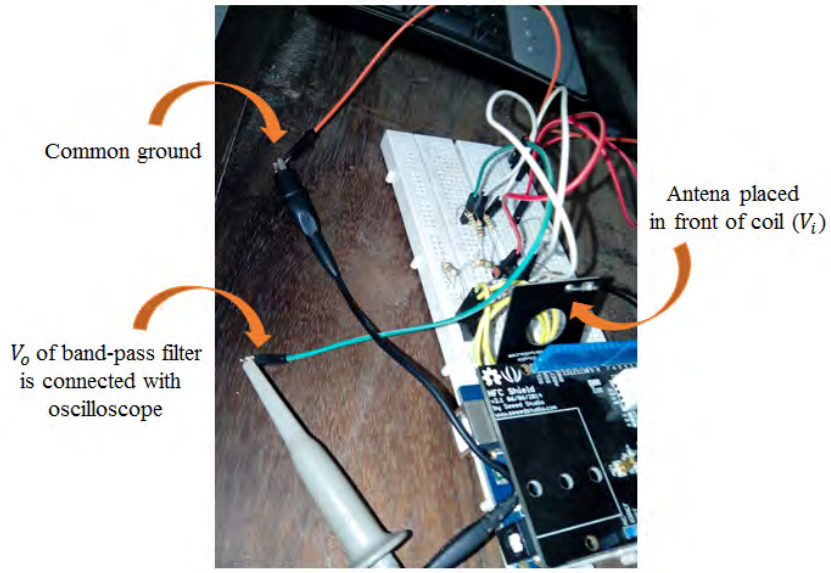
(b) Schematic view

Figure 5.3: Circuit diagram of our attacker module

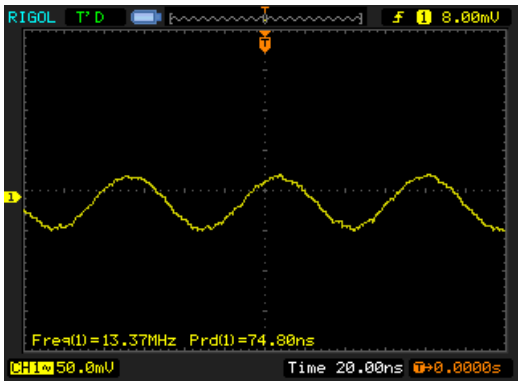
details, challenge-responses, personal details etc. Note that as long as the attacker keeps the channel busy, $Reader_{original}$ cannot detect the presence of any card. Therefore, the attacker can control the channel smartly to ensure that $Reader_{original}$ cannot figure out any channel switch during communication. To clarify, Figure 5.2(c) shows a screen-shot of demo attack. Here, for simplicity, we just concatenate two zeros with the original message under transmission through the attacker module. Therefore, the original reader receives the message "50000" (red marked) when the original tag sends message "500" (green marked). The left side of Figure 5.2(c) shows a legitimate transmission, whereas the right side shows a transmission under attack.



(a) Circuit diagram of band-pass filter



(b) Device setup



(c) Capturing signals using oscilloscope

Figure 5.4: Testbed hardware setup for measuring signal amplitude

5.1.2 Circuit Diagram of Our Attacker Module

As discussed above, our attacker module consists of three components: a reader, a writer and a tag. For this purpose, two Arduino Uno boards are programmed to communicate with each

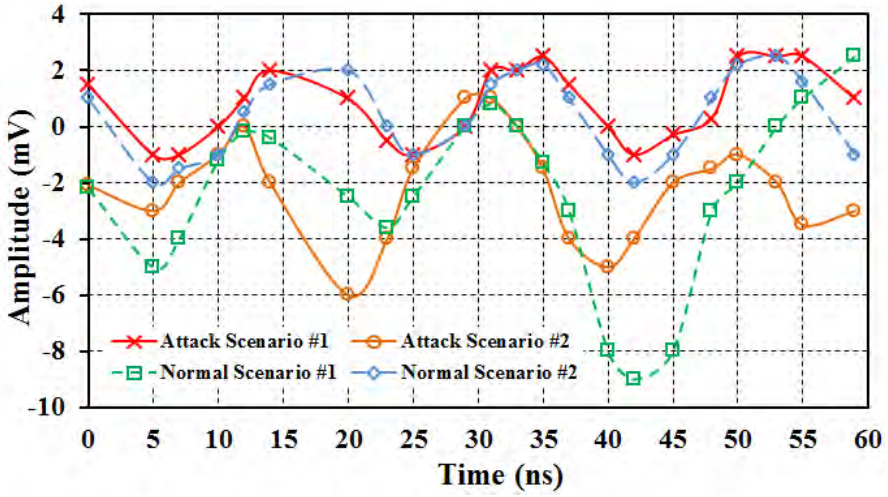


Figure 5.5: Changes in signal amplitude

other in a Master Reader/Slave Sender configuration via the I2C synchronous serial protocol. Several functions of Arduino's Wire Library are used to accomplish this. Arduino 1, the Master Reader, is programmed to request, and then read, 3 bytes of data sent from the uniquely addressed Slave Sender Arduino. Once that message is received, it can then be viewed in the Arduino Software (IDE) serial monitor window.

The I2C protocol involves using two lines to send and receive data: a serial clock pin (SCL) that the Arduino or Genuino Master board pulses at a regular interval, and a serial data pin (SDA) over which data is sent between the two devices. Pin 4 (the data, or SDA, pin) and pin 5 (the clock, or SCL, pin) on the master board are connected to their counterparts on the slave board. Both boards share a common ground. In order to enable serial communication, the master board is connected to a computer via USB. Figure 5.3 shows the circuit diagram of our attack model.

5.2 Findings of Real Deployment

Recall from Chapter 2, where we show why existing defense strategies cannot be applied for our MITM attack. We now report experimental results on the physical nature of NFC communications with and without an MITM attack to identify insights for successful defense.

5.2.1 Variation of Signal Amplitude

First off, we try to check whether there exists any fluctuation in signal level in presence of the attacker module. To do so, we design a band-pass filter that is able to capture signals of 13.56 MHz (approximately) frequency. To display the signals we use RIGOL DS1052E oscilloscope. Figure 5.4(a) depicts the circuit diagram of band-pass filter, Figure 5.4(b) depicts the snapshot of our device setup, and Figure 5.4(c) depicts the snapshot of the oscilloscope's output during measuring signals.

From this experiment, we find a significant change in signal strength (amplitude value) in presence of an MITM attacker which provides insights of a mechanism to detect the attack. However, this change is not due to MITM attack alone, significant change is noticed also for other reasons for example, varying proximity between card and reader, molecular absorption etc. Figure 5.5 shows the change in signal amplitude over time for two separate instances of normal and attack scenarios. The green and blue curves indicate signal amplitude for two normal scenarios and red and orange curves indicate signal amplitude for two attack scenarios. Here, the red curve vastly differs from the green curve which provides insights of a mechanism to detect the attack. However, unfortunately, even for two normal scenarios (green and blue curves) without MITM attack, the signal amplitudes are vastly different. With a view to identifying unknown patterns (if there exists any) from amplitude data to separate attack scenarios from normal scenarios we applied several machine learning algorithms using Weka tool. For this purpose, we take data for six different tags. For each tag, we take data five times for normal scenarios and five times for attack scenarios. Thus, total sixty times data is taken where thirty times for normal scenarios and thirty times for attack scenarios. From these signal data, total 1860 coordinates are retrieved to input in machine learning algorithms in order to classify normal and attack scenarios. However, we did not find good accuracy here. Table 5.1 shows accuracy rate of different algorithms in classifying two scenarios (normal scenarios and attack scenarios). Here, best accuracy is only 59% which precludes signal amplitude as a reliable marker to detect MITM attack.

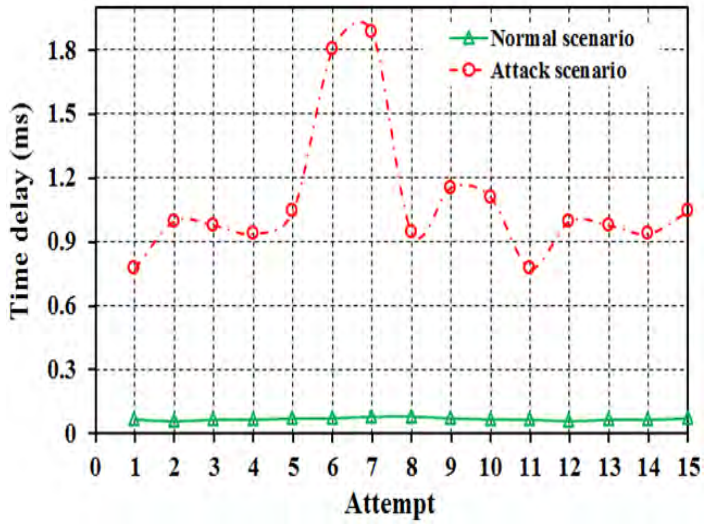
Table 5.1: Success rate of different machine learning algorithms in differentiating normal scenarios and attack scenarios using signal amplitude

Method	**Success Rate (%)**	**Method**	**Success Rate (%)**
BayesNet	54	NaiveBayes	54
NaiveBayesMultinominalTest	50	NaiveBayesUpdatable	55
Logistic	55	SGD	57
SGDTest	50	SimpleLogistic	55
VotedPerception	55	SMO	58
IBK	53	Kstar	53
LWL	53	AdaBoostM1	51
AttributeSelectedClassifier	59	Bagging	55
ClassificationViaRegistration	56	CVParameterSelection	50
FilteredClassifier	59	IterativeClassifierOptimizer	52
LogitBoost	53	MulticlassClassifier	55
MulticlassClassifierUpdatable	57	MultiScheme	50
RandomCommittee	55	RandomizableFilteredClassifier	55
RandomSubspace	59	Stacking	50
Vote	50	WeightedInstancesHandlerWrapper	50
DecisionTable	59	Jrip	55
OneR	59	PART	57
ZeroR	50	DecisionStump	50
HoeffdingTree	57	J48	59
LMT	54	RandomForest	55
RandomTree	55	REPTree	58

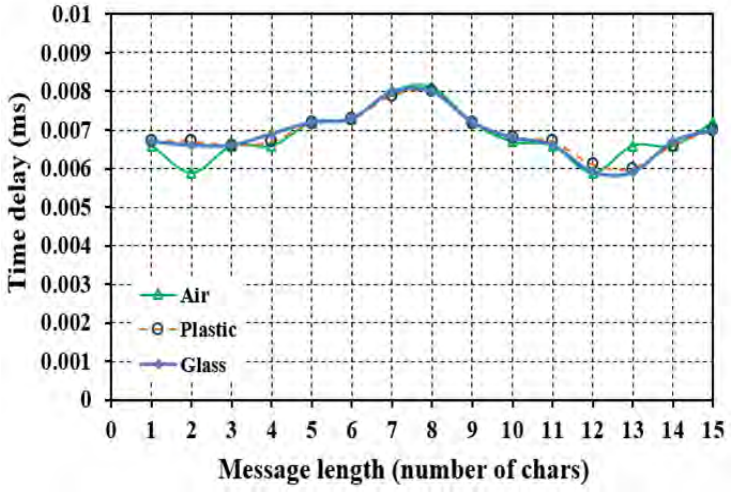
5.2.2 Variation of Time Delay

Now, in our attack scenario, since message passes through some extra devices before reaching the benign one, and also there occurs a message modification, there exists a reasonable time delay. In our experiments, a reader normally takes only 59 to 81 milliseconds (ms) to read a card without an MITM, whereas it takes 777 to 1863 ms (around 20 times more) in presence of an MITM. Please note that, this time may vary depending on several things for example, micro-controller capacity, internal wire delay and also on the time the writer takes to release the channel.

Figure 5.6(a) shows time variation between normal and attack scenarios obtained from our experiment. Here, the green curve indicates required time for normal scenarios and the red curve indicates required time for attack scenarios. Significant gap between two carves depicts



(a) Time variation between normal and attack scenario

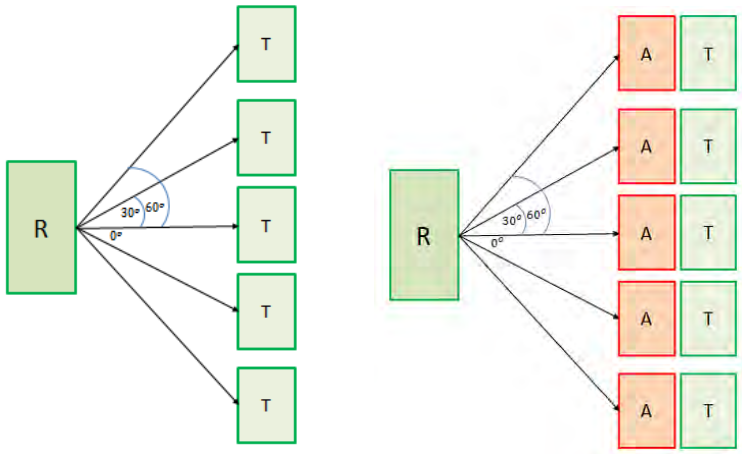


(b) Time delay for different communication medium between card and reader

Figure 5.6: Time delay in different scenarios

non-negligible increase in delay in presence of an MITM which provides a mechanism to detect the MITM attack.

Note that, the delay should also vary depending on message length and medium of communication as per intuition. Therefore, to check whether the delay actually varies with the message length, we measure time delay for different message lengths in our experiments. However, since the maximum length of payload gets fixed while being in NFC communications, and since the *nfc.read()* command reads the whole card at a time, the variation in message length



(a) Normal communication (b) Attacker with tag

Figure 5.7: Reading card in different angles

Settings	Angles between Card and Reader (°)	Success Ratio (%)	Delay			
			Min	Max	Avg.	Stdev
Legitimate	0	100	66	81	71	7
	30	100	66	108	79	17
	60	67	67	109	85	15
Attack	0	100	777	1883	1199	440
	30	100	946	2108	1488	581
	60	56	2251	4527	3383	1299

Table 5.2: Percentages of improvement using our protocol with respect to different alternative protocols

does not exhibit any significant effect in delay. This also validates increased delay as the best marker for detecting MITM attack. It is also important to note that in our experiments, the increase in delay is found to be independent of the medium of communication between the card and the reader (i.e., air, plastic, glass, etc.) as shown in Figure 5.6(b), further validating the impact of leveraging increase in delay to detect MITM attack.

To further clarify, we take data at different angles between card and reader for both normal and attack scenarios. Table 5.2 presents results obtained from the setup presented in Figure 5.7. As presented in Figure 5.7, the reader is at slightly different angles compared to the original card with and without the MITM attacker. Here, at 0°angle, success rate is 100% and reading time varies from 66 ms to 81 ms in normal scenarios. However, in attack scenarios,

success rate is still 100% but reading time varies from 777 ms to 1883 ms which is more than ten times than that of the normal scenarios. At 30°angle, reading time is almost same for normal scenarios but delay increases in attack scenarios. Beyond an angle of 60°between the reader and the tag, success rate of communication goes down, which also provides a marker for detecting non-aligned MITM attackers. Therefore, according to our experiment, the increase in delay is consistent and non-negligible between the normal scenario and the attack scenario which validates the time delay to be used as a key metric to detect the MITM attack.

Chapter 6

Proposed Defense Mechanism

From our experiment, we find that there exists a good amount of delay when MITM occurs. This delay is independent of message length and communication medium. Therefore, it can be a valid metric to prevent the attack. According to our mechanism, central authority (possibly, issuer of the card) should fix the maximum time T_{max} required to read the specific tag and a threshold θ. A reader calculates the reading time $T_{current}$ when it reads a tag. If the difference between $T_{current}$ and T_{max} is greater than the threshold θ, reader will suspect some abnormality and deny the transaction.

Now, T_{max} and θ can have different value for different applications. In our experiment, maximum reading time is 81 ms when reader and card are perfectly aligned. However, it takes 27 ms to 28 ms more if card is in slightly different angle compared to the reader. After a complete analysis on time delay, we determine $T_{max} = 81$ ms and $\theta = 30$ ms. With this threshold, we successfully detect the attack.

Table 6.1: Comparison among existing countermeasures

Countermeasures	**Require Extra Hardware**	**Can Solve Proposed MITM Attack?**
Secure channel with a shared secret key [1, 2, 3]	No	No
Location aware and safer cards [6]	Yes	No
Tap-Tap & Pay [10]	Yes	No
PUF-based Authentication [15]	Yes	No
Our proposed mechanism	No	Yes

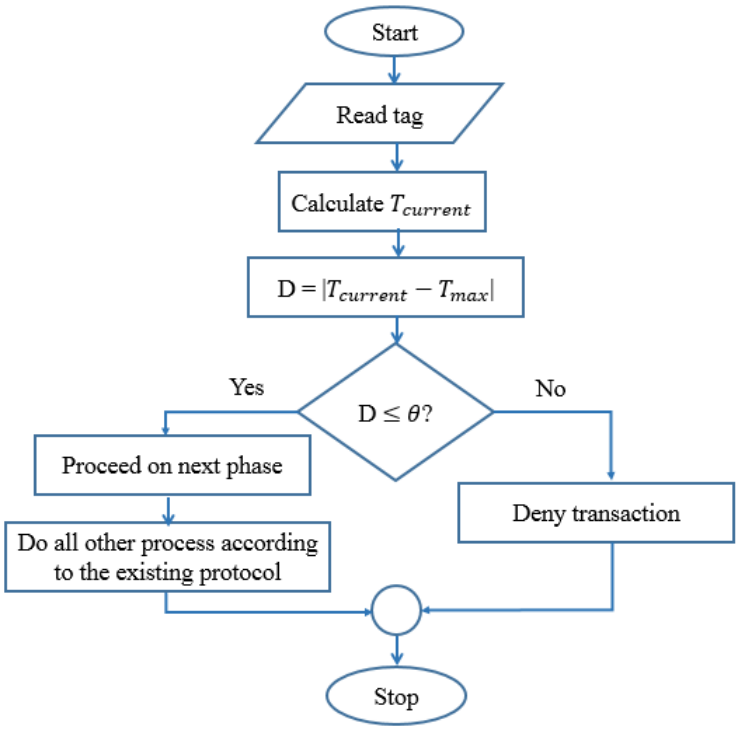


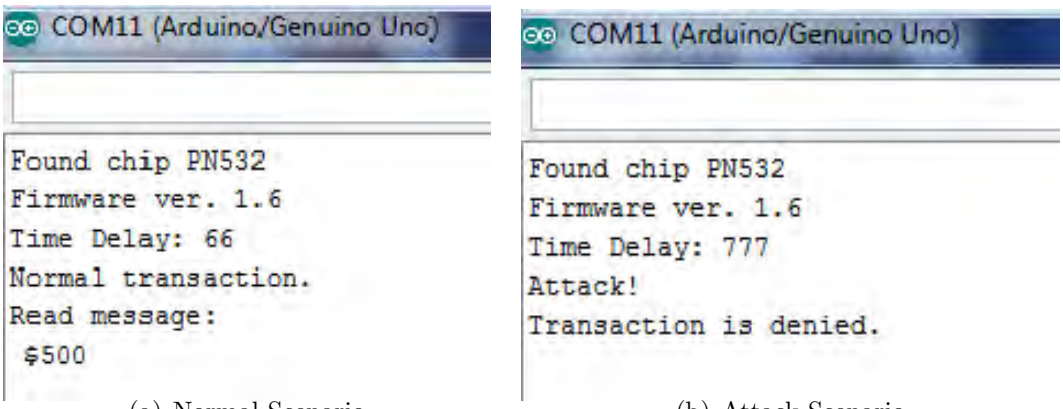
Figure 6.1: Proposed Defense Mechanism

Figure 6.1 depicts the flow of actions of our proposed mechanism in detail. Here, $T_{current}$ is calculated by the reader during reading a tag. Then, difference D between $T_{current}$ and T_{max} is calculated and compared with the the threshold θ that is considered to be set previously by the central authority. If D exceeds the threshold θ, transaction is denied. Otherwise, it will proceed to the next step allowing communication.

Algorithm 1 Detection of attack exploiting *time delay*

```
1: x ← start time
2: if tag is present then
3:     read the tag
4:     y ← current time
5:     d ← y − x.
6:     if d > θ then
7:         alert "Attack!"
8:         break
9:     process tag data
```

Figure 6.2 shows two screen shots where first one depicts the normal scenario and second one depicts the attack scenario. In Figure 6.2(a), reading time is 66 ms. Thus, considering it as a normal transaction, message *500$* is displayed. On the other hand, in Figure 6.2(b), reading



(a) Normal Scenario (b) Attack Scenario

Figure 6.2: Detection of attack exploiting *time delay*

time is 777 ms, which is grater than the threshold value. Thus, considering it as a malicious one, our system declines this transaction.

Using our mechanism, we successfully detect the MITM attacks in our experiment. In detecting the attack, success rate of our mechanism is 100%. Moreover, it is fully algorithmic. Thus, it does not demand extra cost for additional hardware either at user end or at service provider end. Table 6.1 shows a comparison of our proposed defense mechanism with existing alternative countermeasures.

It should be mentioned that use of time delay is not new in the field of security. Delay is being used for both benign and malicious purpose. In timing attack, flow of a system can be guessed by injecting query and observing time delays in the response [57]. On the other hand, register-to-register path delay can be used as a Trojan detection method [58]. Inter-packet delay can also be used to identify the source of an attack where intruders use intermediate "Stepping Stones" to conceal their identity [59]. Thus, use of time delay as a key metric to detect the attack is rational.

Note that a check of the card number in transaction authorization phase or another dynamic authentication during the transaction authorization phase may be other alternative solutions to the MITM attack. Advanced authentication mechanism such as fDDA and CDA adopt such countermeasures. In case of having any of these advanced authentication mechanisms in operation, the possibility of MITM attack will not remain in practice. However, as pointed

earlier, numerous cards all over the world are already in operation [42] with DDA, not having either fDDA or CDA. Thus, these cards will remain vulnerable under the MITM attack.

Chapter 7

Conclusion and Future Work

In this thesis, we demonstrate for the first time, the practical feasibility of MITM attack over NFC communication that was supposed to be infeasible, due to close proximity between devices and electromagnetic fundamentals. We present practical attack scenarios in the realm of contactless payments to clarify our attack model. We also present a defense mechanism exploiting time delay against our newly-established MITM attack. A comparative analysis is performed among existing countermeasures and our proposed solution to show its effectiveness.

We are currently investigating approaches to reduce the form factor of MITM over NFC communications. In this thesis, we used three extra devices for attacker module. In future, we will try to establish the attack in more convenient way by reducing the number of attacker devices so that the attacker can be thinned further and getting into the middle of NFC devices could be much easier. Besides, at present, our attack works in active-passive mode. In future, we will analyze whether the attack is possible in peer-to-peer communication mode too.

Finally, in this study, we show that existing security mechanisms used for NFC communications fail to prevent our attack. Accordingly, we present a defense mechanism based on the experimental results on the physical nature of NFC communications with and without an MITM attack. We plan to conduct more rigorous theoretical and experimental studies through a combination of algorithmic and hardware technologies to devise more effective countermeasure based on a firmware to combat MITM attacks.

Appendices

Appendix A

Source Code

In this section, we present the source code of original reader and source codes we use to develop our attacker module. As discussed above, one reader and one writer are used in our module. For this purpose, two Arduino Uno boards are programmed to communicate with each other in a Master Reader/Slave Sender configuration via the I2C synchronous serial protocol. Several functions of Arduino's Wire Library are used to accomplish this. Here, the malicious writer works as the Master, which is programmed to request, and then receive data sent from the uniquely addressed Slave sender. In our experiment, Master requests 3 bytes of data from the Slave device #8 (Source Code for Malicious Reader). Here, malicious reader works as the Slave, which is programmed to read the original tag and send the tag data to the malicious writer through wire.

Source Code for Original Reader

```
#include <SPI.h>
#include <PN532_SPI.h>
#include <PN532.h>
#include <NfcAdapter.h>
#include <Time.h>

PN532_SPI pn532spi(SPI, 10);
NfcAdapter nfc = NfcAdapter(pn532spi);
char data[20];

void setup(void) {
    Serial.begin(9600);
    delay(500);
    Serial.println("NDEF Reader");
    delay(20000);
    nfc.begin();
}

void loop(void) {
    Serial.println("\nScan a NFC tag\n");

    if (nfc.tagPresent())
    {
        NfcTag tag = nfc.read();         // read the tag
        //tag.print();
        NdefMessage message = tag.getNdefMessage();
        NdefRecord record = message.getRecord(0);
        int payloadLength = record.getPayloadLength();
        byte payload[payloadLength];
        record.getPayload(payload);

        for(int i=1; i<payloadLength; i++)
        {
            data[i-1] = (char)(payload[i]);
            Serial.print(data[i-1]);
        }
        Serial.print("\n");
        Serial.print("Time: ");
        Serial.println(millis()); // print reading time
    }
}
```

Source Code for Malicious Writer (Master)

```
#include <Wire.h>
#include <SPI.h>
#include <PN532_SPI.h>
#include <PN532.h>
#include <NfcAdapter.h>
#include <Adafruit_PN532.h>
#include <avr/wdt.h>

PN532_SPI pn532spi(SPI, 10);
NfcAdapter nfc = NfcAdapter(pn532spi);
char data[20];

#define PN532_IRQ   (2)
#define PN532_RESET (3)

void setup() {
  Wire.begin(); // join i2c bus (address optional for master)
  Serial.begin(9600); // start serial for output
  nfc.begin();
  wdt_enable(WDTO_60MS); // set interval threshold
}

void loop() {
  if(nfc.tagPresent())
  {
      int index = 0;
      Wire.requestFrom(8, 3); // request 3 bytes from slave device #8
      while (Wire.available()) {
        char c = Wire.read(); // receive a byte as character
        data[index++] = c; // store the character
      }
      data[index++] = '0'; // modifying the message (concatenate zero)
      data[index++] = '0';
      data[index] = '\0';

      if(data[0] != '0'){       // message is received from slave device
        NdefMessage message = NdefMessage();
        message.addUriRecord(data);//convert character array into NDEF message format
        bool success = nfc.write(message); // write NDEF message into the tag
        if(success){
            Serial.println(data); // print the message written into the tag
        }
      }
      else{     // no message is received from slave device
        bool success = nfc.write(""); // write null value
      }
  }
}
```

Source Code for Malicious Reader (Slave)

```
#include <Wire.h>
#include <SPI.h>
#include <PN532_SPI.h>
#include <PN532.h>
#include <NfcAdapter.h>

PN532_SPI pn532spi(SPI, 10);
NfcAdapter nfc = NfcAdapter(pn532spi);
char data[20];

void setup() {
  Serial.begin(9600);
  Wire.begin(8); // join i2c bus with address #8
  nfc.begin();
}

void loop() {
  if (nfc.tagPresent())
  {
        NfcTag tag = nfc.read(); // read the tag

        NdefMessage message = tag.getNdefMessage();
        NdefRecord record = message.getRecord(0);
        int payloadLength = record.getPayloadLength();
        byte payload[payloadLength];
        record.getPayload(payload);
        int i;
        for(i=1; i<payloadLength; i++)
        {
            data[i-1] = (char)(payload[i]);
        }
         data[i-1] = '\0';

         Serial.print(data); // print payload value
         Serial.print("\n");
         Wire.onRequest(requestEvent); // register event
    }
}

// function that executes whenever tag is present and data is requested by master
// this function is registered as an event, see loop()
void requestEvent() {
  if (nfc.tagPresent())
        Wire.write(data); // respond with message of 3 bytes as expected by master
  else
        Wire.write("000");
}
```

Appendix B

Acquirer Bank can Deny an Internationally Issued Card

In this section, we include an anonymized email conversation with a bank of Canada, which indicates that they accept internationally issued VISA cards in most cases. It eventually means that they do not accept all international cards in all cases. Acquirer banks can deny to accept a card according to their own policy.

Subject: RE: A query regarding using VISA card[20183145041678248 2/20183145041678248 2]

From:customer.support@xxx.com

To:YYY@YAHOO.COM

Date: Tuesday, March 20, 2018, 5:39:58 PM GMT+6

Good Morning YYY QUERY MAKER,

Thank you for taking the time to write to XXX BANK.

Great question, **in most cases** Visa cards issued internationally can be used abroad and with our POS machines with the attachment of an additional external charge which you will have to contact your issuing bank to further inquire on.

To give you an example, the Visa cards we issue with our bank can be used internationally with the following additional charges:

A 2.5% fee will apply when you use your XXX BANK Credit Card or Visa Debit Card for purchases which result in the conversion of Canadian dollars to a foreign currency. And, for foreign currency withdrawals performed at ATMs outside of Canada, the exchange rate includes an amount equal to 2.5% of the converted amount. You can find more about this in XXX BANK's "About Our Accounts and Related Services" document:

http://www.xxxcanadatrust.com/document/PDF/accounts/513796-20171030.pdf

Regarding the rate used, your credit card and debit card transactions will be converted to Canadian dollars based on the foreign exchange rate charged to XXX BANK Canada Trust in effect at the time that the transaction is processed. Foreign exchange rates fluctuate from one second to the next throughout the business day. The reason for this is that they are based on real time foreign exchange market rates. As a result it is difficult to tell you what rate you will receive. Please follow this link to a foreign exchange calculator on our website that you may find helpful:

http://www.xxxcanadatrust.com/foreignExchange.form?lang=en

Again, for the rates charged by your bank, it is best to contact them for additional assistance.

Please let us know if we can help you book a branch appointment, if so please reply with the best location and time.

Alternatively, please feel free to contact us at 1-866-222-**** (24/7) collect 416-983-****.

Should you need to contact us from overseas at any time and don't wish to use airtime, a great way to connect with us is to use a Voice over IP service such as Skype or Google Hangouts. A representative will be more than happy to assist using whichever Voice over IP service you choose. Voice over IP services are calls placed over the Internet (WiFi) without the use of mobile airtime.

To make a call with these services, simply open one of these services, login, select the "Call" option and dial our number at 866-222-**** (EasyLine). Please ensure the country code is set to "Canada + 1" and that will include the "1" in the number for you.

I trust this provides clarity. Please be sure to write back if you have any additional questions or if I can offer any further assistance. Enjoy the rest of your day!

Take care and all the best!

Nissreen I Digital Communications I XXX Group
XXX Canada Trust 1-866-222-****
TDD (Telephone Device for the Hearing Impaired) 1-800-361-**** (toll free)
Follow XXX_Canada on Twitter
Become a XXX Fan on Facebook

Subscribe to XXX BANK's YouTube Channel
Download the XXX BANK Mobile App Now

Disclaimer;

XXX BANK Canada Trust endeavors to provide accurate and up-to-date information relating to its products and services. However, please note that rates, fees and information are subject to change. Remember that email sent over the Internet is generally unencrypted. We recommend that you use caution when forwarding free-format email messages to us and that you do not include confidential information (such as account numbers or other personal data) in those messages, as they are not encrypted. Please note that if you have disclosed any account numbers or personal information in your email, we have blocked the information to protect your privacy.

Dear Concern,

I am planning to travel Canada soon.

I was wondering whether my VISA credit card, issued from a bank in a country other than Canada, can be used in Canada. For example, can the card be used in the stores that use POS machines acquired from you?

I will highly appreciate your response.

--
Regards,
YYY QUERY MAKER

Appendix C

Accepting Transaction Without PIN

In this section, we include an anonymized email from an issuer bank to the acquirer bank to verify a fraud transaction, which has been possible due to the PoS device being unable to process PIN. Thus, a successful transaction has been performed without providing PIN.

From: AAA,Card [mailto:aaa.card@xyzbank.com]
Sent: 15 April, 2018 7:02 PM
To: ZR <z.r@wxybank.com>; npsb@wxybank.com ; md.s@wxybank.com
Cc: HOC <as.card@xyzbank.com >; H Card <h.card@xyzbank.com >; card@xyzbank.com ; M Card <m.card@xyzbank.com >
Subject: Requesting to verify a fraud transaction

Dear Sir

Greetings from XYZ Bank Card Division!

Please be informed that one of card holder unconsciously lost his card and a transaction has been made using that lost card at your merchant before block the card. Since the POS device was not PIN enable hence the transaction become successful without providing PIN. In this circumstance, our card holder make a general diary about the fraud transaction. Below we have listed the details of the transaction for your ready reference. Please check the log and suggest us how we can resolve the issue.

Card Number: 599999****123456
RRN: **8094*****864**
Terminal ID: **59****19**
Terminal Address: HHH Shop
Amount: **9440.00**
Transaction Date: **April 04, 2018**

Thanking you
AAA
Card Division
XYZ Bank Ltd.

Bibliography

[1] E. Haselsteiner and K. Breitfuss, "Security in Near Field Communication (NFC)", Workshop on RFID Security, 2006.

[2] A. Suraperwata and I. Pratiwi, "Solutions to Near Field Communication (NFC) Vulnerabilities Against Interception Type Attacks", CISAK, 2013.

[3] S. Kavya, K. Pavitra, S. Rahman, M. Vahini, and N Harini,"Vulnerability Analysis And Security System For NFC-Enabled Mobile Phones", International Journal of Scientific and Technology Research Volume 3, Issue 6, 2014.

[4] J. Daemen, V. Rijmen, "The Design of Rijndael: AES - The Advanced Encryption Standard", Springer Science & Business Media, Mar 9, 2013.

[5] A. Mahalanobis, "Deffie-Hellman Key Exchange Protocol, Its Generalization and Nilpotent Groups", PhD thesis, Florida Atlantic University, August 2005.

[6] Di Ma, Anudath, N. Saxena, and T. Xiang, "Location-Aware and Safer Cards: Enhancing RFID Security and Privacy via Location Sensing", IEEE Transactions on Dependable and Secure Computing (Volume: 10, Issue: 2, March-April 2013).

[7] W. Nel, A. Burger, "Proving Cybercriminal's Possession of Stolen Credit Card Details on Compromised POS Devices", 5th International Conference on Management Leadership and Governance, 2017.

[8] "NFC market size worldwide 2014-2024", https://www.statista.com/statistics/691585/global-nfc-market-size/, Last accessed on January 09, 2018.

[9] NFC Forum. Available at: http://www.nfc-forum.org, Last accessed on January 19, 2017.

[10] M. Mehrnezhad, F. Hao, and F. Shahandashti, "Tap-tap and pay (TTP): preventing the mafia attack in NFC payment. International Conference on Research in Security Standardisation, Springer, Cham, 2015.

[11] "NFC Shield V2.0", http://www.seeedstudio.com/depot/NFC-Shield-V20-p-1370.html, Last accessed on January 19, 2017.

[12] "Arduino UNO", https://www.arduino.cc/en/Main/ArduinoBoardUno, Last accessed on January 19, 2017.

[13] "Arduino Mega 2560", http://www.arduino.cc/en/Main/ArduinoBoard Mega2560, Last accessed on January 19, 2017.

[14] "Card Acceptance Guidelines for Visa Merchants", Visa, 2015.

[15] L. Bolotnyy and G. Robins, "Physically Unclonable Function -Based Security and Privacy in RFID Systems", PerCom'07, 2007.

[16] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan, "Lightweight Secure Search Protocols for Low-cost RFID Systems", ICDCS'09, June 2009.

[17] A. Francillon, B. Danev , S. Capkun, "Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars", In Proceedings of The 18th Annual Network And Distributed System Security Symposium. The Internet Society, 2011.

[18] J. Brun-Nouvion, H. Hossayni, "Security models", 1st Semester 2010/2011.

[19] G. Shu-qin, WU Wu-chen, H. Li-gang, and Z. Wang, "Anti-collision algorithms for Multi-Tag RFID", Radio Frequency Identification Fundamentals and Applications Bringing Research to practice, February 01, 2010.

[20] M/Chip, Acquirer Implementation Requirements, "MasterCard PayPass".

[21] "Hacker's Demo Shows How Easily Credit Cards Can Be Read Through Clothes And Wallets", https://www.forbes.com/sites/andygreenberg/2012/01/30/hackers-demo-shows-how-easily-credit-cards-can-be-read-through-clothes-and-wallets/#248d41bf78a6, Last accessed on April 3, 2017.

[22] "Contactless EMV Payments: Benefits for Consumers, Merchants and Issuers", http://www.emv-connection.com/downloads/2016/06/Contactless-2-0-WP-FINAL-June-2016.pdf, Last accessed on April 3, 2017.

[23] "Contactless Specifications for Payment Systems", Book B, Version 2.6, July 2016.

[24] M. Bakhoff, "EMV (Chip-and-PIN) Protocol", December 15, 2014.

[25] D. Ruiter, Joeri, and E. Poll. "Formal analysis of the EMV protocol suite." Joint Workshop on Theory of Security and Applications, Springer Berlin Heidelberg, 2011.

[26] J. Murdoch, S. Drimer, R. Anderson, and M. Bond, "Chip and PIN is Broken", IEEE Symposium on Security and Privacy, May 2010.

[27] "Visa Integrated Circuit Card Specification", Version 1.5, May 2009.

[28] Technical Specification, "PayPass M/Chip", Version 1.3, September 2005.

[29] C. Mulliner, "Vulnerability Analysis and Attacks on NFC-enabled Mobile Phones", International Conference on Availability, Reliability and Security, 2009.

[30] V. Coskun, F. Soylemezgiller, B. Ozdenizei, and K. Ok, "Development and Performance Analysis of Multifunctional City Smart Card System", International journal of Computer, Electrical, Automation, Control and Information Engineering, 2014.

[31] W. Park, D. H. Kim, and D. Lee, "Vulnerability of Rechargeable RFID Tag Card Based on NFC", International Journal of Control and Automation, 2015.

[32] http://www.statista.com/statistics/251306/nfc-payment-transaction-value-in-the-united-kingdom, Last accessed on April 20, 2017.

[33] P. Cortese, F. Gemmiti, B. Palazzi, M. Pizzonia, and M. Rimondini, "Efficient and Practical Authentication of PUF-Based RFID Tags in Supply Chains", 2010 IEEE International Conference on RFID-Technology and Applications (RFID-TA), June 2010.

[34] L. Dongsheng, Z. Xuecheng, Li Yongsheng, and Li Xiaohuang, "Anti-collision algorithm for RFID systems", Journal of Huazhong University of Science and Technology, 2006-09.

[35] G. Madlmayr and J. Langer, "NFC Devices: Security and Privacy", Third International Conference on Availability, Reliability and Security, 2008.

[36] A. Mitrokotsa, R. Rieback and S. Tanenbaum, Classifying RFID attacks and defenses", Springer Science & Business Media, July 29, 2009.

[37] A. Asaduzzaman, S. Mozumder, S. Selinas, and Muhammad F. Mridha, A security-aware Near Field Communication architecture", Networking, Systems and Security, 2017.

[38] S. Ghosh, A. Majumder, J. Goswami, A. Kumar, S. P. Mohanty, and B. K. Bhattacharyya, Swing-Pay: One Card Meets All User Payment and Identity Needs: A Digital Card Module using NFC and Biometric Authentication for Peer-to-Peer Payment", IEEE Consumer Electronics Magazine, 2017.

[39] Y. Ma, NFC Communications-based Mutual Authentication Scheme for the Internet of Things", International Journal of Network Security, 2017.

[40] J. Ling, Y.Wang, W. Chen, An Improved Privacy Protection Security Protocol Based on NFC", International Journal of Network Security, 2017.

[41] "Introduction to Security", http://web.cs.ucla.edu/classes/winter13/cs111/scribe/17b, Last accessed on January 23, 2018.

[42] "Strengthening Card Authentication: a migration to DDA", Smart Payment Association, July, 2015.

[43] https://www.atmmarketplace.com/videos/arqc-and-arpc-generation-and-validation/, Last accessed on February 21, 2018.

[44] "Transaction Authorization Process", http://blog.unibulmerchantservices.com/transaction-authorization-process/, Last accessed on February 21, 2018.

[45] "Chip and Skim: cloning EMV cards with the pre-play attack", Security and Privacy (SP), 2014.

[46] T. Aura, "Payment systems", https://mycourses.aalto.fi/pluginfile.php/556240/mod_resource/content/2/10-security-payment.pdf, Last accessed on February 28, 2018.

[47] "Contactless payments mean card fraud now happens after cancellation", https://www.theguardian.com/money/2015/dec/19/contactless-payments-card-fraud-after-cancellation-bank-account, Last accessed on February 28, 2018.

[48] "Bank Agreement", https://www.braintreepayments.com/en-sg/legal/bank-agreement?referrer=https%3A%2F%2Fmail.yahoo.com%2F, Last accessed on February 28, 2018.

[49] "Credit Card Authentication", https://www.investopedia.com/terms/c/credit-card-authentication.asp, Last accessed on February 28, 2018.

[50] A. Kurt, Effectiveness of Cyber Security Regulations in the US Financial Sector: A Case Study", Carnegie Mellon University, 2015.

[51] Woodforest National Bank, "Merchant Payment Card Application/Agreement", www.bpsworldwide.com/files/standard_application.pdf, Last accessed on February 28, 2018.

[52] "Merchant Services Application and Agreement", https://ucollect.biz/files/ucollect-ach-cc.pdf, Last accessed on February 28, 2018.

[53] 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", Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, 2014.

[54] "EMV in a nutshell", http://www.cs.ru.nl/ erikpoll/papers/EMVtechreport.pdf, Last accessed on March 27, 2018.

[55] "Contactless Payment Cards: Vulnerabilities, Attacks, and Solutions", https://cybercamp.es/cybercamp2015/sites/default/files/contenidos/material/slides_cybercamp-15.pdf, Last accessed on March 27, 2018.

[56] "Chip and PIN", http://chipnpin.blogspot.com/2012/01/what-is-arqc.html, Last accessed on April 01, 2018.

[57] W. G. Halfond, J. Viegas, and A. Orso, A Classification of SQL Injection Attacks and Countermeasures", Proceedings of the IEEE International Symposium on Secure Software Engineering, 2006.

[58] M. Tehranipoor and F. Koushanfar, A Survey of Hardware Trojan Taxonomy and Detection", IEEE Design Test of Computers, 2010.

[59] X. Wang, D. S. Reeves, and S. F. Wu, Inter-Packet Delay Based Correlation for Tracing Encrypted Connections through Stepping Stones", Springer Berlin Heidelberg, 2002.