Bachelor thesis
Computer Science

Radboud University

Using NFC enabled Android devices to attack RFID systems

Author:
Tiko Huizinga
s4460898

First supervisor/assessor:
prof. dr. Eric Verheul
`eric.verheul@ru.nl`

Second supervisor:
dr. ir. Erik Poll
`e.poll@cs.ru.nl`

Second assessor:
dr. Joeri de Ruiter
`joeri@cs.ru.nl`

August 17, 2018

Abstract

RFID cards are used all over the world for many different applications. This thesis looks at how Android phones can be used to compromise the security of RFID cards. This is interesting because security issues should be known to users of RFID cards in order to make an educated decision about their usage. The reason we look at Android phones specifically is because they are widely available with NFC. In this thesis, we execute three types of attacks with Android phones on RFID cards: skimming, cloning and relay attacks. Of these attacks, the relay attack is investigated in most detail. In earlier research, two Android NFC-enabled devices were used to perform a relay attack on contactless EMV cards, which are bank payment cards[19]. This thesis extends on that earlier research by modifying the Android applications used to perform relay attacks on other RFID cards as well. This thesis describes the specifications that an RFID card must meet for a relay attack with NFC-enabled phones to be possible. We also succeeded in unlocking a rental car and driving away with it using a cloned RFID card. The research is a combination of literature study and by actual experiments with the listed attacks on a selected group of RFID cards like the e-NIK card which is the Dutch ID card, the IRMA card and the cards used for MyWheels.

Contents

Glossary

AID Application Identifier. Identifier to select an application on an ISO/IEC 7816-4 card. 14

ATR Answer to Reset. A message in early stage of communication between a contact smart card and a reader. The message contains information on the cards parameters, manufacturer and more. 28

EMV Europay MasterCard Visa. Payment card system. 5

eNIK Elektronische Nederlandse Identiteitskaart. The Dutch ID card with a chip that communicates via RFID. 31

HCE Host card emulation. Software architecture that permits an NFC device to emulate a contactless smart card using software only. 17

IC Integrated Circuit. A chip that can store data make comutations. 22

IRMA I Reveal My Atributes. Privacy friendly identity platform for authentication and signing. 31

ISO/IEC A joint technical committee between the International Organization for Standardization and the International Electrotechnical Commission, created to developed to develop, maintain, promote and facilitate IT standards. 5

MIME Multipurpose Internet Mail Extensions. Examples are: text/plain, image/png, image/jpeg, audio/mp3. 17

NDEF NFC Data Exchange Format. This is a standardized data format to exchange data between a NFC device and another NFC device or tag. Each NDEF message contains one or more NDEF records. A NDEF record has a header and a payload. 17

NFC Near Field Communication. 5

RFID Radio Frequency Identification. 4

URI Uniform resource identifier. 17

Chapter 1

Introduction

RFID cards are smart cards which can communicate via radio waves on a limited range. RFID cards are used in many different applications such as access to a secured building, authentication at a shared printer, public transportation or payment cards. The market value of RFID systems keeps growing because it is being used more and more[1]. Many contact smart cards are getting RFID support to make them contactless smart cards. Android NFC enabled devices can read from, write to and emulate RFID cards. The devices are widely available and for sale for only a few hundred euros. This raises the question on how these devices might impact the security of systems using RFID cards.

1.1 The research question

The research question for this thesis is:

- How can Android NFC devices be used to compromise the security of systems that use RFID cards?

1.2 Structure of the thesis

This thesis shows four different attacks on RFID cards: eavesdropping, skimming, cloning and relay attacks. For each of these attacks, we explain under what conditions the attack is possible and discuss possible countermeasures.

We limit our research to ISO/IEC 14443 compatible cards. This follows from the research question where we use Android NFC devices and NFC is based on ISO/IEC 14443.

[1] `https://www.idtechex.com/research/reports/rfid-forecasts-players-and-opportunities-2017-2027-000546.asp`

We find that skimming an RFID card is easy when the data on the card is not encrypted and that there are already many different Android applications to do this as discussed in Chapter 2. We also find that eavesdropping RFID communications is possible, with very specific technology. We did not find any mentions in literature of eavesdropping using an Android NFC device. We did no mentionable effort to research if eavesdropping with Android NFC devices could be possible.

In earlier research, a relay attack was performed against contactless EMV cards, which are RFID cards[19][14][13]. This thesis extends that research by modifying the Android applications used in the thesis of van den Breekel[19] to be able to perform relay attacks on more different types of cards. An explanation of the Android applications and an overview of the changes are found in Chapter 4.

Chapter 2, explains eavesdropping, skimming, cloning and relay attacks. This chapter explains the basic ideas of these attacks which are essential for the rest of the thesis. It also gives some practical examples of these attacks.

Chapter 3 contains more background information on RFID. This chapter gives an overview of the different protocols on RFID cards, like ISO/IEC 14443, ISO/IEC 7816 and MIFARE. This chapter also explains the difference between RFID and NFC because these two terms seem to get mixed up in literature.

Chapter 4, shows the possibilities and limitations of Android phones with regard to RFID. It also contains an overview of the most important Android applications used during this research. This chapter also shows how we modified two existing relay applications so they would relay more RFID cards than only EMV cards.

Chapter 5 describes security mechanisms for the Machine Readable Travel Documents (e-passports). The reason this is a separate chapter, is because there are many security mechanisms for the MRTDs, which together cover eavesdropping, skimming and cloning attacks and these mechanisms are described in great detail.

In Chapter 6 we execute skimming, cloning and relay attacks on different cards and explain the results.

Chapter 2

Eavesdropping, skimming, cloning and relay attacks

This chapter explains four attack techniques that can be used by adversaries to compromise the security of a system that uses ISO/IEC 14443 RFID cards: eavesdropping, skimming cloning and relaying. In Chapter 6, we use the skimming technique to gather information on the different RFID cards and use that information to execute relay attacks on these cards. Although there are many different types of RFID cards and these attacks can be executed on all of them, the specific ranges named in this chapter apply to ISO/IEC 14443 proximity cards.

An important security feature of RFID cards is the limited range at which they can operate. This is around a maximum of 10 cm (a more detailed analysis in Chapter 3). Each of these attacks have their own range at which they can operate which will be discussed in this chapter as well.

2.1 Eavesdropping (passive)

The definition of eavesdropping is secretly listening to a private communication between two parties. Eavesdropping on RFID cards is secretly listening to a communication between the card and the reader. This is done using an antenna to capture and decode the data.

2.1.1 Eavesdropping maximum range

In [6], the authors achieved to eavesdrop from a distance of 18 meters on the communication between an RFID card and a reader. This reader was connected to a laptop with an USB cable of about 2m long and this cable amplified the signal which made it possible to eavesdrop from such a long range. When an anti-electromagnetic clip was added to the cable, their eavesdrop setup received no usable signal.

Other research achieved to eavesdrop on a distance of 2-3m[9].

2.1.2 Countermeasures to eavesdropping

The information an attacker gains with this attack is limited to what the card and reader communicate when the attacker is listening. Encrypting the data before communicating would eliminate the impact of this type of attack.

2.2 Skimming (active)

Skimming is reading data from a card without permission. It is a more active type of attack than eavesdropping because here, the attacker himself activates the card to communicate with it. There are many different Android applications that can read data from RFID cards. One of these applications is **NXP TagInfo**. These are the main functionality of the application cited literary from their website[1]:

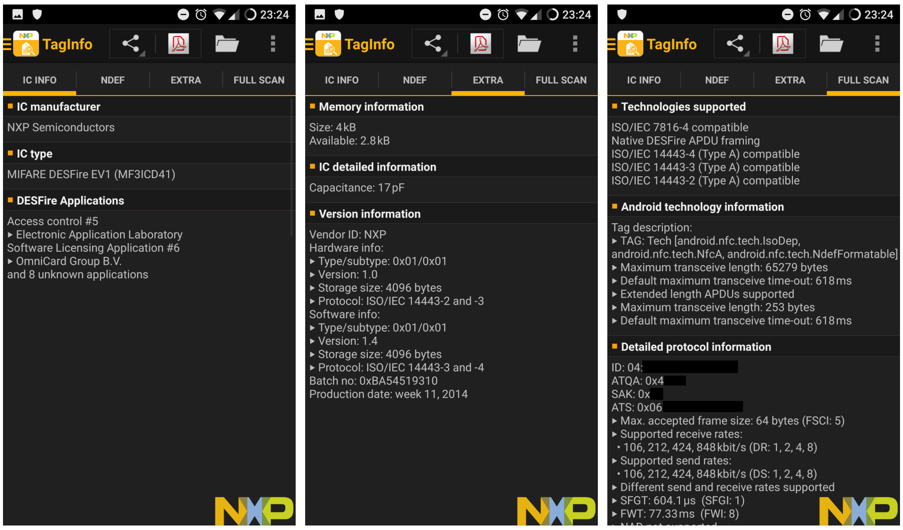
1. Value checker function for a selected range of public transport systems
2. Identify applications contained on cards and tags
3. Identify IC types and IC manufacturer
4. Extract and analyze NFC data sets (NDEF messages)
5. Read out and display the complete tag memory layout

The second and third functionality are used in Chapter 6 to find out what protocols and technologies the card supports. See Figure 2.1 for an example using a RU student card with some identifying information censored.

2.2.1 Maximum range to communicate with a card

In Table 3.1 is stated that the read distance for a HF tag, like an RFID card, is 10-20 cm. With a normal reader, this is true. In more recent research, a device was created with which a card could be activated and read from a distance of 50 cm, way higher than a normal card reader[8]. The device they use for this is large and difficult to handle which makes it harder to use this in a real world skimming scenario.

[1] `https://www.MIFARE.net/en/products/tools/nfc-taginfo-app/`



IC manufacturer

NXP Semiconductors

IC type

MIFARE DESFire EV1 (MF3ICD41)

DESFire Applications

Access control #5
▸ Electronic Application Laboratory
Software Licensing Application #6
▸ OmniCard Group B.V.
and 8 unknown applications

Memory information

Size: 4 kB
Available: 2.8 kB

IC detailed information

Capacitance: 17 pF

Version information

Vendor ID: NXP
Hardware info:
▸ Type/subtype: 0x01/0x01
▸ Version: 1.0
▸ Storage size: 4096 bytes
▸ Protocol: ISO/IEC 14443-2 and -3
Software info:
▸ Type/subtype: 0x01/0x01
▸ Version: 1.4
▸ Storage size: 4096 bytes
▸ Protocol: ISO/IEC 14443-3 and -4
Batch no: 0xBA54519310
Production date: week 11, 2014

Technologies supported

ISO/IEC 7816-4 compatible
Native DESFire APDU framing
ISO/IEC 14443-4 (Type A) compatible
ISO/IEC 14443-3 (Type A) compatible
ISO/IEC 14443-2 (Type A) compatible

Android technology information

Tag description:
▸ TAG: Tech [android.nfc.tech.IsoDep, android.nfc.tech.NfcA, android.nfc.tech.NdefFormatable]
▸ Maximum transceive length: 65279 bytes
▸ Default maximum transceive time-out: 618 ms
▸ Extended length APDUs supported
▸ Maximum transceive length: 253 bytes
▸ Default maximum transceive time-out: 618 ms

Detailed protocol information

ID: 04:[redacted]
ATQA: 0x4[redacted]
SAK: 0x[redacted]
ATS: 0x06[redacted]
▸ Max. accepted frame size: 64 bytes (FSCI: 5)
▸ Supported receive rates:
• 106, 212, 424, 848 kbit/s (DR: 1, 2, 4, 8)
▸ Supported send rates:
• 106, 212, 424, 848 kbit/s (DS: 1, 2, 4, 8)
▸ Different send and receive rates supported
▸ SFGT: 604.1 µs (SFGI: 1)
▸ FWT: 77.33 ms (FWI: 8)

Figure 2.1: NXP TagInfo reading the RU student card

2.2.2 Risks of skimming attacks

One of the risks of a skimming attack is the loss of privacy. For example a user Alice, can have her EMV card in her pocket. An attacker Eve comes up to Alice and holds her phone next to the pocket of Alice. After skimming the card Eve could know the account number of Alice and/or her transaction history. With the knowledge of the bank account number, Eve could make an unauthorized direct debit. With the transaction history, Eve could see if Alice recently bought something expensive or withdrew money. This could be used as a preparation for a mugging.

Another risk of skimming is that it can be used as a preparation for a cloning attack. An attacker skims the data and puts the data on another card or device to clone the original card.

2.2.3 Countermeasures to skimming

An easy way for a user to prevent a skimming attack is to put the RFID card in a metal case which forms a Faraday cage around the card and will prevent HF radio signals from interfering with the card.

For a card issuer, a countermeasure to skimming is an Machine Readable Zone on the card. This can be used as a key which is needed to start communication with the card. The card reader must have visual contact with the card to read the zone if it wants to communicate with the card. This prevents a skimming attack from someones wallet or pocket where the attacker .

Another way to prevent skimming, is to require a pin code that only the user of the card knows to be able to read a card.

Further, a card can have a list of public keys of trusted card readers. To read a card, the reader has to authenticate itself to the card by signing a challenge sent by the card. The disadvantage is that it is hard to revoke certificates because for this, direct access to the card is needed.

Finally, the less information is stored in the card, the less information can leak during a skimming attack.

2.3 Cloning

Cloning of a card is a step further than skimming. Where skimming is actively reading data from a card, cloning is putting that data on another card or device to mimic the original card. An attacker does not always have to clone all the information present on the card. This depends on what he wants to use the cloned card for. In some cases, cloning the UID of a card could be enough to fool the system. Section 6.2.6 contains a successful attack on a car renting service in which a cloned UID was used to unlock a car and drive away.

2.3.1 Ways to receive data to clone

There are different ways to receive the data to clone to another card or device. This makes it that The most obvious is to skim the RFID card. Another way is to eavesdrop on a card communicating with a reader and to extract the data from the responses of the card. A third way is when data is stored in a database and the attacker has access to this database.

2.3.2 Countermeasures to cloning

To prevent cloning, a card can have a public and private key-pair stored on the card. The private key must be in secure memory which can not be read by a card reader. For a reader to gain access to the data, it sends a challenge to the card. The card signs this message with its private key and sends it back to the reader, which can verify the signature with the public key of the card. Implementations of the protocols that prevent cloning of Machine Readable Travel Documents are active authentication and chip authentication which are described in Chapter 5. As the private key is stored in secure memory and cannot be read by a card reader, cloning of such a card is only possible with hardware based attacks on the card, which are significantly harder.

2.4 Relay attack

Imagine you do not know the rules of chess but you want to play at least a draw to a chess grandmaster. You can challenge two chess grandmasters to a game. In game A you play black, in game B you play white. When your opponent in game A makes his first move, you play the same move in game B. Then you wait for your opponent in game B to make his move and you play that same move in game A. If you continue like this until the games end, both games will end in the same result. You either drew in both the games or you won one game and lost the other. From this point on you will be able to say you drew or won from a chess grandmaster without even knowing the rules. This idea was described as the chess grandmaster problem[3] and it is a type of relay attack.

In a relay attack there are three parties. Two of them are the victims, let's call them Alice and Bob. The third party is the attacker, called Eve. When Eve wants to do a relay attack on Alice and Bob, Eve initiates the communication to both parties and all the communication goes via Eve. Both Alice and Bob think they are talking to a worthy partner. For Alice, Eve seems like a trustworthy girl because Eve responds just like her friend Bob. For Bob the same thing is true but the other way around.

2.4.1 Relay attack with RFID cards

To execute a relay attack on an RFID card, two devices are needed:

- Mole device that communicates with the RFID card and pretends it is a legitimate card reader
- Relay device that can communicates with the RFID card reader and pretends it is a legitimate card

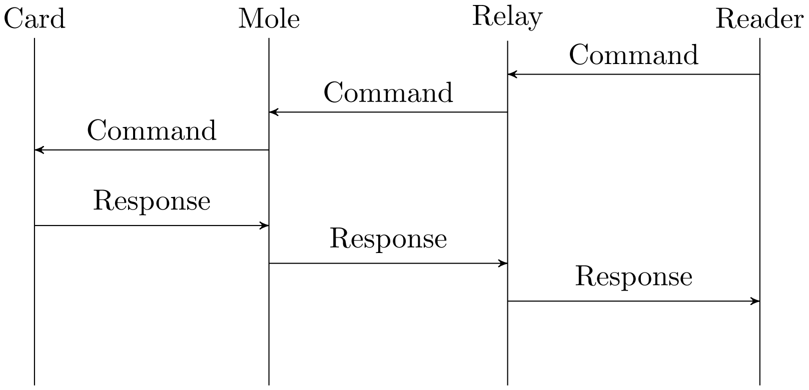


Figure 2.2: Message sequence diagram for a relay attack on RFID cards

An NFC enabled Android phone can do both these things as Chapter 4 describes. Previous research succeeded in relaying contactless EMV (bank) cards using two Android NFC-enabled devices[19], meaning that they were able to pay at a payment terminal using the EMV card through the phones. To achieve this, they used two Android applications, one for each device. These applications were specifically designed and programmed to work for EMV cards. In Section 4.4.5, these applications are modified to make them able to relay more different cards.

2.4.2 Range of a relay attack

The range of this relay attack is global because the two Android NFC-enabled devices can communicate with each other over the Internet. This is a problem because one security features of an RFID card is that it can only communicate when held close to the reader. The relay attack breaks this.

2.4.3 Risks of relay attacks

For card users, a relay attack forms a big risk when the card is used for authorization. An attacker Eve can authorize as user Alice without Alice knowing. This section describes three scenarios. The first two scenarios poses a direct risk to the user of the card. The third scenario poses a direct risk to the card issuer. Note that when the direct risk is for the user, this always forms an indirect risk for the issuer because of bad publicity or even legal responsibility.

Scenario: EMV relay pickpocketing

Alice stands in the line at the supermarket. Eve holds her phone next to Alice's pocket. Eve's accomplice Chuck is in the next line and is about to pay. He uses his phone to pay contactless and via the relay Chuck pays with Alice's card.

Scenario: EMV relay deception

In a parking garage with two payment terminals, Eve tapes his phone on top of the payment terminal and disguises it so it looks like a normal payment terminal. When Alice wants to pay for her parking ticket she goes to the payment terminal that Eve prepared. She tries to buy her ticket. Her card is being relayed to some other payment terminal where Eve buys something else and Alice does not get her parking ticket. Alice goes to the other payment terminal because this one does not work and she gets a parking ticket and leaves, not knowing that Eve stole her money.

Scenario: OV card

If a relay attack on the OV chip card were possible, the following scenario shows how this can be abused. Eve wants to make a long train journey from Nijmegen to Amsterdam without paying. She checks in in Nijmegen, travels to Amsterdam but instead of checking out, she calls her friend Chuck who is still in Nijmegen. Chuck holds his phone to the checkout terminal in Nijmegen and Eve holds her OV card to her phone. She is now checked out in Nijmegen and did not have to pay for her journey. Meanwhile, if she were checked by the conductor, she would pass because she checked in in Nijmegen.

2.4.4 Countermeasures to relay attacks

This section gives an overview of countermeasures to relay attacks. Section 2.4.4 and Section 2.4.4 make use of the fact that a relay is an extra step in the communication that delays the communication. When it is possible to identify this delay, it is possible to stop communication.

Timing based on speed of light

The first countermeasure based on timing uses the fact that RFID communication travels with the speed of light. If you know exactly how long it takes for the card to process the message and send the response, it is possible to use this time frame to identify a relay attack because it delays the communication between the card and the reader[10]. Unfortunately in real world applications, it turns out that it often is not possible to predict how fast a card will response because this depends on the circumstances like the power of the reader device, the build quality of the card and noise or obstruction of the radio waves.

Timing countermeasure against relay with 'basic' hardware

To counter relay attacks with NFC enabled Android phones, researchers developed a distance bounding protocol[1]. They found that relaying dynamic messages (messages for which the card has to calculate something before responding) does not always result in larger times between request and response but relaying static messages (messages the card knows the answer to without calculations) does impact the timing significantly. The problem is that the attacker can also knows the responds to the static messages. As a reader, to verify that you are communicating directly to a trusted card, you would want the card to authenticate faster than the delay a relay device would add. So they wanted the combination of authentication (dynamic) but with the timing of static messages. This is possible by first asking the card to compute the authentication and later asking for this result.

The researchers designed this protocol specifically for contactless EMV cards but the principal is applicable to smart card protocols in general.

Physical protection

Just like with skimming, a user can protect their cards from being accessed without their knowledge by putting the card in a metal case. Note that this protection does not work when the relay is being initiated by this user because they trust the relay device. Section 2.4.3

Chapter 3

RFID and NFC

The terms RFID and NFC are being used interchangeably in literature. They are, however, not the same. Section 3.1 describes what RFID is. Section 3.2 contains information and buildup of the ISO/IEC 14443 standard for RFID cards. Section 3.3 describes the ISO/IEC 7816 which is the standard designed for contact smart cards. Section 3.4 describes the different types of MIFARE cards and their relation to the ISO/IEC standards.

3.1 RFID

RFID stands for **R**adio **F**requency **Id**entification. It is the technique to identify objects using radio frequency waves and is described in ISO/IEC 18000. There are RFID transponders (*tags*) which can be wirelessly queried by RFID transceivers (*readers*) using radio waves. The tag will respond with some identifying information and optionally extra data. A tag contains an antenna and a memory chip to store data. It could also contain a power source like a battery to extend the maximum range of the contactless communication. Without this power source, the reader powers the tag via electromagnetic induction. This is called a passive tag.

RFID is used on different frequency ranges as can be seen in Table 3.1 below[20]. In this thesis we will focus on High-Frequency tags, more specifically tags compatible with ISO/IEC 14443, as these are the only ones that are interoperable with NFC. These are the tags that are used in contactless proximity cards. As you can see, the passive read distance according to Weis is about 10-20 cm. Recent research achieved to activate and read an ISO/IEC 14443 card from a distance of 50 cm[8].

3.2 ISO/IEC 14443

ISO/IEC 14443 is a standard for RFID cards. ISO/IEC 14443 consists of four parts. I will give a short description of what is specified in these parts.

Frequency Range	**Frequency**	**Passive Read Distance**
Low Frequency (LF)	120-140 KHz	10-50 cm
High Frequency (HF)	13.56 MHz MHz	10-20 cm
Ultra-High Frequency (UHF)	868-928 MHz	3 meters
Microwave	2.45 & 5.8 GHz	3 meters
Ultra-Wide Band (UWB)	3.1-10.6 GHz	10 meters

Table 3.1: Common RFID frequencies and their maximum read distances[20][8]

This is needed to understand the relation between this standard, ISO/IEC 7816 and NFC.

3.2.1 ISO/IEC 14443-1: Physical characteristics

Part 1 is about the physical characteristics like the size of the card which is not important in the scope of this paper.

3.2.2 ISO/IEC 14443-2: Radio frequency power and signal interface

Part 2 specifies the power and field strength needed for a connection. It makes a distinction between Type A and Type B cards in modulation and bit-codings. The result is that any reader that is able to read type A tags, can only read type B tags if it has the appropriate software/drivers. In terms of hardware, there is no difference between type A and B tags or readers.

3.2.3 ISO/IEC 14443-3: Initialization and anti-collision

Part 3 specifies how protocol initialization and anti-collision should be handled. Again, this is different for Type A and B cards.

3.2.4 ISO/IEC 14443-4: Transmission protocol

This part describes data exchange between the reader and the card. This is the same for type A and B cards.

3.3 ISO/IEC 7816-4

ISO/IEC 7816 is the standard for smart cards. It was designed for smart cards with contact chips. Part 4 of this standard defines the ‘language’ in which a card and a card reader communicate. For sending and receiving data, ISO/IEC 7816-4 defines APDU messages. The reader sends a command APDU and the card responds with a response APDU.

At the start of communication between a card and the reader, the reader asks the card if it supports one or more certain applications using the Application ID (AID). If the card supports this application, it will communicate further using this applications protocol.

Concactless RFID cards can use ISO/IEC 7816-4 on top of ISO/IEC 14443-4 to act like a contact card. This way, all the existing applications for contact cards did not have to be rewritten for contactless cards.

3.4 MIFARE

MIFARE is a trademark for contactless smart cards. The trademark is owned by NXP-semiconductors, co-inventor of NFC. The different MIFARE protocols are: MIFARE DESfire, MIFARE Ultralight, MIFARE Classic and MIFARE Plus. The Classic and Ultralight protocols operate on top of the ISO/IEC 14443-3 protocol. The MIFARE Plus protocol can operate on both ISO/IEC 14443-3&4 protocols. The DESfire card operates on top of the ISO/IEC 14443-4 protocol and uses optional ISO/IEC 7816-4 commands like AID[17][16].

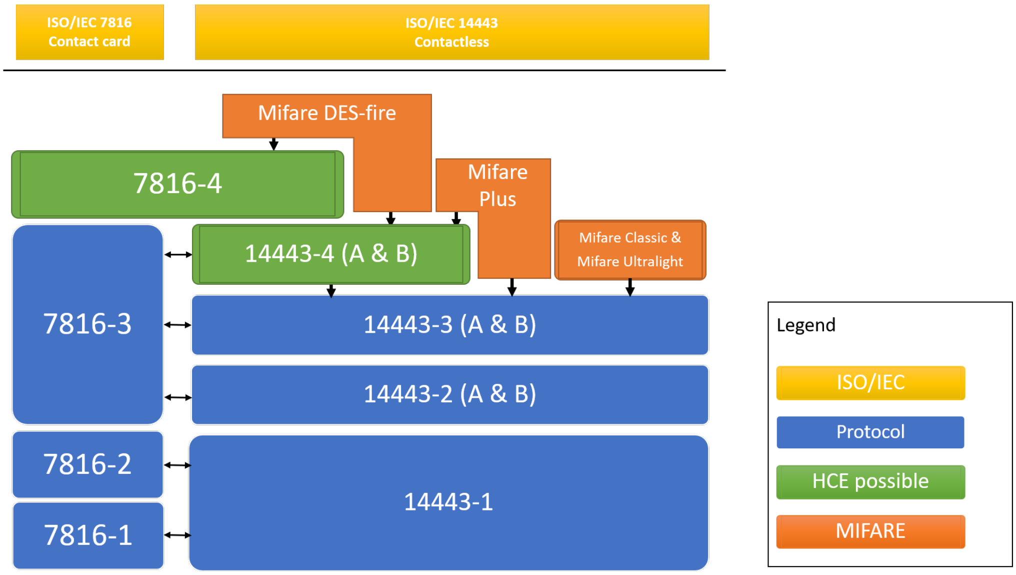


Figure 3.1: Relation between smart card protocols

3.5 NFC

NFC stands for Near Field Communication. It is a wireless interface between devices, similar to Bluetooth, infrared or Wi-Fi. NFC is implemented in most modern smartphones[2]. An NFC device has an antenna and a controller with a power source. It can be used in active mode and in passive mode. In active mode, it communicates with another NFC device which is also in active mode to transmit data. In passive mode, it communicates with an RFID tag or with an RFID reader. NFC is based on ISO 14443 (RFID cards) and it communicates over the same frequency [18]. Because of this, an NFC-enabled device can communicate with an RFID tag. In Chapter 4, we will discuss how NFC works in Android devices.

Chapter 4

NFC in Android

Android NFC-enabled devices support three modes of operation: Reader/Writer mode, P2P mode and Host Card Emulation (HCE). The P2P mode allows two NFC devices to communicate which is not relevant in the scope of this thesis because we look at how NFC can impact the security of systems using RFID cards. The other two modes will be explained with reference to the NFC API guide for Android developers[1] in this chapter.

4.1 Basics

When you hold an NFC tag against an NFC-enabled Android phone, the "Tag Dispatch System" will analyze the tag, parse it and start an activity that is waiting for the tag. For the Tag Dispatch System, there are two types of tags. If this tag is an NDEF tag, the system will try to map a MIME or an URI to the NDEF message based on its header. If this is successful, the system will create the intent ACTION_NDEF_DISCOVERED. If unsuccessful, it will create the intent ACTION_TECH_DISCOVERED. The system sends this intent to an application that filters for this intent. If more than one application filters for this intent, the Activity Chooser popup comes up on the screen so the user can decide which application should handle the intent. In Figure 4.1, we can see how the system handles an NFC tag.

4.2 Host Card Emulation

Host Card Emulation gives NFC-enabled Android devices the possibility to 'act' like an RFID card that supports ISO/IEC 7816-4. It is available on Android 4.4 and higher. This functionality is used, for example, to do contactless

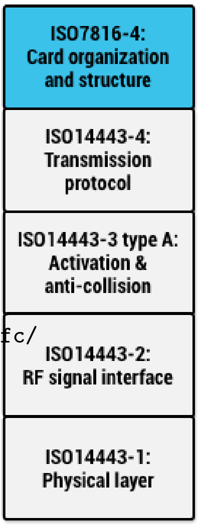


Figure 4.2: HCE protocol stack (Source:developer.android.com)

[1] `https://developer.android.com/guide/topics/connectivity/nfc/`

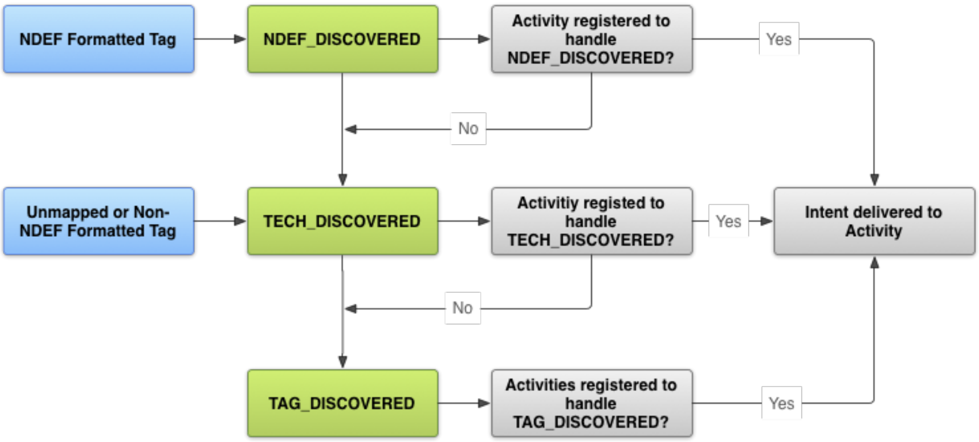

NDEF Formatted Tag
NDEF_DISCOVERED
Activity registered to handle NDEF_DISCOVERED?
Yes
No
Unmapped or Non-NDEF Formatted Tag
TECH_DISCOVERED
Activitiy registed to handle TECH_DISCOVERED?
Yes
Intent delivered to Activity
No
TAG_DISCOVERED
Activities registered to handle TAG_DISCOVERED?
Yes

Figure 4.1: Tag dispatch system (Source: `https://developer.android.com/guide/topics/connectivity/nfc/nfc.html`)

payments using Android payment applications by the corresponding bank like ING[2], ABN Amro[3] and most other Dutch banks.

It is possible to emulate cards based on the ISO 14443-4 protocol and cards running the ISO 7816-4 layer on top of the 14443-4 protocol. It also states: “Support for Nfc-B (ISO/IEC 14443-4 Type B) technology is optional”[5]. This could mean that some Android versions also support HCE for type B cards.

4.2.1 AID routing

When a card is held to a card reader, the first message from the reader to the card is: ”Do you support application AID?”. When holding an Android NFC device to the card reader, the reader will ask the same question. The Android OS then looks if there is any HCE application installed that supports this AID and routs this and the next messages during this communication to this application. This process is called AID routing.

[2] `https://play.google.com/store/apps/details?id=com.ing.mobilepayments&hl=nl`

[3] `https://play.google.com/store/apps/details?id=com.abnamro.nl.mobile.wallet&hl=nl`

4.3 Modify AID routing

The Android OS does not support a way to send all APDU messages to a certain application, no matter what AID is requested. If we want to create a relay application that relays many different RFID cards without having to specify the AID in the source code of the application, we need some way to modify the standard AID routing.

To achieve this, we need a modification to the the Android OS. The Xposed Framework is a framework for rooted Android devices. It makes it possible to install modules that modify the core of the Android OS on the device. It gives the possibility to make system level changes without having to install a new custom ROM. In this thesis we will use it to modify the way Android handles AID routing.

The Xposed module *Modify AID Routing*[4] makes it possible to hard code a 'Magic AID' in to an NFC HCE application. This acts as a wild card. No matter what AID the card reader is asking for, the APDU will always be routed to the application with the magic AID. This module works only with Android 4.4 because the implementation of AID routing in later versions of Android changed. Specifically, this module modifies the method 'resolveAidPrefix' in "src/com/android/nfc/cardemulation/RegisteredAidCache.java"[5].

4.4 Existing Android applications

This section gives an overview of the main Android NFC applications we used for this thesis. The Relay Reader and Relay Emulator are the relay applications that are being used in Chapter 6.

4.4.1 Credit Card Reader

Credit card reader is an application to read contactless EMV cards. The application is made by Julien Millau[6]. In this thesis version 4.3.6 of this application is used. Since June 2018, Only the Pro version of this application is still available on the Google Play store[7]. The application has the following features:

1. Show bank account number

2. Show expiration date

[4] https://github.com/johnzweng/XposedModifyAidRouting

[5] https://android.googlesource.com/platform/packages/apps/Nfc/+/android-4.4.4_r2.0.1/src/com/android/nfc/cardemulation/RegisteredAidCache.java

[6] Github page of Julien Millau: https://github.com/devnied

[7] https://play.google.com/store/apps/details?id=nfc.credit.card.reader.pro2

3. Show how many pin tries are left

4. Show transaction history when available

5. Show APDU communication log

4.4.2 NXP TagInfo

Earlier shown in Section 2.2. For completeness here the main functionalities of this application again:

1. Value checker function for a selected range of public transport systems

2. Identify applications contained on cards and tags

3. Identify IC types and IC manufacturer

4. Extract and analyze NFC data sets (NDEF messages)

5. Read out and display the complete tag memory layout

In this thesis we use version 4.23 of this application[8].

4.4.3 NFC-proxy

At DEFCON 20, one of the presentations was about relaying contactless credit cards[14]. They created NFC-proxy as relay application for EMV credit cards. This was the first application that we tried to use for the relay attack in this thesis. Unfortunately, nothing seemed to happen when trying the relay attack on an EMV card. We did not pursue why it did not work.

4.4.4 NFC Card Emulator Pro

NFC Card Emulator Pro[9] is an application that emulates RFID cards by cloning their UID. There are two ways to add a card UID in this application. The first one is by just holding the device near an RFID card and it will read out the UID and save it. The other way is to enter the UID manually. This application can save multiple cards with a button next to each card to simulate that card. To simulate the cards, the Android device must be rooted because the application has to edit the UID in the NFC configuration file on the phone. There are systems that only use the UID of a card as identifier. This application makes those

[8] NXP TagInfo PlayStore page: `https://play.google.com/store/apps/details?id=com.nxp.taginfolite&hl=nl`

[9] `https://play.google.com/store/apps/details?id=com.yuanwofei.cardemulator.pro&hl=en_US`

4.4.5 Relay Reader & Relay Emulator

The Relay Reader and Relay Emulator are the applications used in the masters thesis of Jordi van den Breekel[19] improved by Joeri de Ruiter. These applications are not publicly available but stored in a private git repository which I received from Joeri de Ruiter.

The Relay Reader is the application that should be held against the card. It will also start a server on a chosen port for the Relay Emulator to connect to.

The Relay Emulator is the application that emulates a contactless EMV card. It must connect to the Relay Reader's server and then be held against the payment terminal. This will relay the communication between the card and the payment terminal. All APDU's are being logged.

Modifications

Magic AID We added the following line in the apduservice.xml file of the Relay Emulator. This is the Magic AID which acts as a wild card so all the APDUs from a card reader go to the Relay Emulator.

```
[language=Java]<aid-filter android:name="F04E66E75C02D8" />
```

AID-group category We registered the application as a payment application so no other payment application would pop up when holding the device to a payment terminal.

```
[language=Java]<aid-group android:category="payment">
```

APDU message length In the communication between the Relay Reader application and the Relay Writer application, the length of the APDU message in bytes is specified by only one byte. This means that the length can not be longer than 256 bytes. The applications use this variable to set the buffer size for the receiving application.

When executing the relay on the IRMA card, one of the messages was 257 bytes long. This resulted in a byte overflow on the length variable which resulted in the length variable being set to 1. We modified the applications such that if the length of the incoming APDU is 1, set the buffer to 257 to receive 257 bytes.

A command APDU is an APDU from the reader to the card. This APDU has a header of 4 bytes and is thus never 1 or 2 bytes small. The maximum length of a command APDU is 258 bytes.

The maximum length of a response APDU from the card to th reader is 65538 bytes, which is $2^{16} + 2$ so the bug still exists. To fix the bug we would need to have three bytes defining the length of an APDU.

Chapter 5

Security measures in Machine Readable Travel Documents

This chapter describes the security measures that are used fore Machine Readable Travel Documents (e-passports). The reason that these measures are described so extensively in a seperate chapter, is because they are well documented and secure protocols to prevent skimming, eavesdropping cloning and ensure data integrity. Section 5.1 first quickly describes some important features of the data structure for MRTDs. Section 5.2 then describes the different security measures and protocols designed to prevent skimming, eavesdropping and cloning and to authenticate the data integrity on the IC.

5.1 Data structure of MRTDs

This section contains the essential parts of the data structures of MRTDs. The information from this section comes from ICAO 9303 part 10[11]

5.1.1 Data Groups

There are sixteen data groups on the IC of an MRTD. This section shows the most interesting ones.

DG1: Machine Readable Zone (Required) This data group contains the full MRZ as it is on the passport as well. This MRZ contains among others:

- Document code
- Name

- Nationality
- Sex
- Etc..

DG2: Face (Required) An image of the face of the holder of the MRTD.

DG3: Finger print(s) (Optional) The finger prints of the owner of the MRTD. This data group is only accessible after Extended Access Control is completed.

DG4: Iris(es) (Optional) The iris(es) of the owner of the MRTD. This data group is only accessible after Extended Access Control is completed.

DG14: Security options Contains security options for additional security mechanisms.

DG15: Active Authentication public key info Contains the public key for Active Authentication described in Section 5.2.3.

5.1.2 Document Security Object

The Document Security Object contains the hash of each of the data groups on the card. These hash values are signed by the issuing state with a Document Signer Certificate. The hashes are used to authenticate the integrity of the data groups via Passive Authentication as described in Section 5.2.1.

5.2 Security measures

The information in this section comes from ICAO 9303 part 11[12].

5.2.1 Passive Authentication

Passive authentication authenticates the data on the IC of the MRTD. This makes it possible for the terminal to verify that the data on the IC is signed by the issuing state of the MRTD, in other words the data is not manipulated. These are the steps the terminal has to do to verify the authenticity of the data:

1. Read Document Security Object
2. Verify the signature of the Document Security Object with the public key of the state that signed the data
3. Read relevant Data Groups

4. Ensure that the contents of the Data Groups are authentic and unchanged by hashing the contents and comparing the result with the corresponding hash value in the Document Security Object

The reason why this authentication is called passive, is because the IC does not have to compute anything.

5.2.2 Basic Access Control

Basic Access Control is the first protocol to run between the terminal and the MRTD. It protects against skimming and eavesdropping. It provides an encrypted secure channel between the IC and the terminal. The encryption key is stored in the Machine Readable Zone (MRZ) which is a string of characters visually accessible on the MRTD. It prevents against skimming when a skimmer does not have visual access to the document contents and thus does not know the key used for communication.

5.2.3 Active Authentication

Active authentication (AA) prevents against copying and cloning of the IC. When AA is available on the IC, data group 15 contains the public key used for AA. the terminal can verify this public key with Passive Authentication of DG15. The private key is stored in secure memory of the IC.

Below are the steps to verify the authenticity of the IC:

1. Optically read the MRZ, compute its hash and compare it with the hash of DG1 in the Document Security Object. (Passive Authentication)

2. The terminal reads the public key from DG15, computes its hash and compares it with the hash of DG15 in the Document Security Object.

3. The terminal challenges the IC with a unique crafted message.

4. The card signs this message with its private key and sends it back to the terminal.

5. The terminal verifies this signature with the public key from DG15 (Step 2).

Privacy Issues with Active Authentication

Active Authentication introduced a privacy issue called Challenge Semantics. It uses the fact that each IC has a unique response to a challenge. If a challenge is sent twice to an IC, the response is the same the first and second time. The problem with this is that this can be used as proof that someone was at a certain location at a certain time. The terminal can save the challenge and response. If the reader wants to know if it has challenged the IC before, it could send the saved challenge(s) to the IC and if it recognizes the response the terminal knows for sure that it has seen the MRTD before. As the challenge can be anything, the terminal can put information on date and time in the challenge and this way remember when and where it has seen the MRTD before.

5.2.4 Extended Access Control

Extended access control is designed to protect access to the biometrics on the IC like fingerprints and irises. The specifications on how to implement this are not in ICAO 9303 part 11. In the EU, extended access control consists of two parts, namely Chip Authentication and Terminal Authentication[4]. This section only looks at how EAC is implemented in the EU.

Chip Authentication (CA) is an Diffie-Hellman key agreement protocol which authenticates the MRTD chip to the terminal and provides secure communication between the two. CA replaces Active Authentication because it prevents challenge semantics and it also generates strong session keys. The public key for CA on the MRTD is stored in DG14. The private key is stored in secure memory. The chip authenticates itself by proving that it has the private key corresponding to its (signed and already authenticted) public key during the Diffie Hellman key exchange.

In Terminal Authentication, the MRTD chip challenges the terminal and the terminal encrypts this message with its private key. The chip contains a list of trusted terminals and their public keys. The chip decrypts the response from the terminal with the corresponding public key and if the plaintext is the correct challenge, the terminal has authenticated itself to the chip and the chip will grant access to the protected biometric data.

Chapter 6

The attacks on different cards

This chapter first describes the setup for the skimming and the relay attack in detail in Section 6. Section 6.2 shows the technical properties and the results of the attacks for each card.

6.1 The attacks

6.1.1 Skimming

To gain information on the type of card, we use the skimming method from Chapter 2. To do this, we use TagInfo by NXP, introduced in Section 2.2. To use this app, open it and just hold the RFID card to the device for a few seconds.

6.1.2 Cloning

We use the Card Emulator application described in section 4.4.4 to clone the UID of the OV chipkaart. This is used in the attack on MyWheelsOpen cars.

6.1.3 Relay

This section describes in steps how we execute the relay attack. The relay is done with two Android NFC enabled devices connected on the same Wi-Fi network. For this attack we use the Relay Reader and the Relay Writer application as described in Section 4.4. To execute a relay attack, the following steps form the basis.

1. You need two NFC-enabled devices with Android 4.4 or higher

2. On the mole device

   Run the Relay Reader application

   Find out what the IP address of this device is

   Set this IP address and a chosen port in settings. This is the configuration to start a server to which the relay device can connect.

3. On the relay device

   Run the Relay Emulator application

   In the settings, set the IP of the mole device and set the chosen port

   Press connect to server

4. Both devices should give a message on the screen to confirm the connection

5. Hold the card to the Mole device and do not remove it until the relay has finished. The mole device should give a message 'Tag discovered'.

6. Hold the Relay device to the reader and the relay will start automatically.

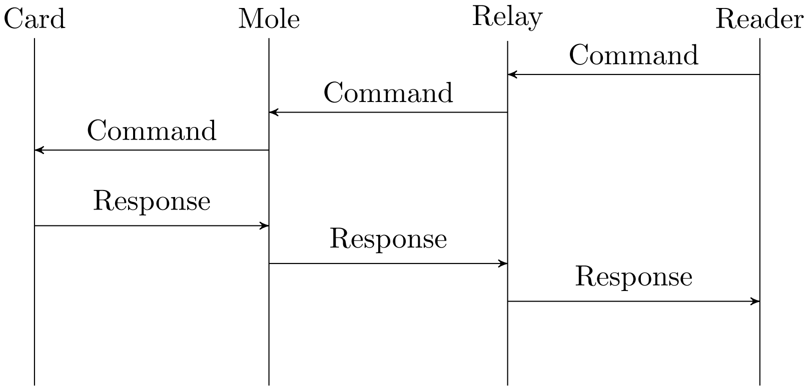


Figure 6.1: Message sequence diagram for a relay attack on RFID cards

Cardpeek

Before we execute the relay attack in a real world situation, we try to test it. To test the relay, we use the computer program Cardpeek[1]. Cardpeek is a program for Windows, MacOS and Linux. It reads the contents of

[1] `https://github.com/L1L1/cardpeek`

ISO/IEC 7816-4 contact or contactless cards using a card reader connected to the computer. Cardpeek has scripts to read EMV cards and to read an e-passport (or a Dutch ID card). Figure 6.2 shows how Cardpeek is normally used without a relay attack. The card is in range of the card reader which is connected to the computer running Cardpeek. In Cardpeek we select for which card we want to run the script.

Figure 6.3, shows the setup for performing a relay attack on an EMV card. The communication now goes through the two Android devices. Cardpeek sees that there is a different ATR at the start of the communication but the rest of the communication is the same.

AID Routing

In section 4.3, we introduced a technique to our relay application which routes all AID requests to the relay app. This technique should make it possible to relay not only the cards for which the application was specifically designed (EMV cards), but all cards using AID select as specified in ISO/IEC 7816-4.

6.2 The cards

6.2.1 EMV bank cards

Relay attacks have already been successfully executed on Dutch EMV contactless cards[19]. Replicating this is the starting point of our experiments.

Properties

Supported technologies according to NXP TagInfo scan:

- MAESTRO card
- ISO/IEC 7816-4 compatible
- ISO/IEC 14443-3(A) compatible
- ISO/IEC 14443-4(A) compatible

Results

Skimming results Aside from the data from NXP TagInfo, the specialized application **Credit Card Reader**[2] extracts more information from contactless EMV cards. This application tries to extract the bank account number, the expiration date of the card, some technical information and

[2] `https://github.com/devnied/EMV-NFC-Paycard-Enrollment`

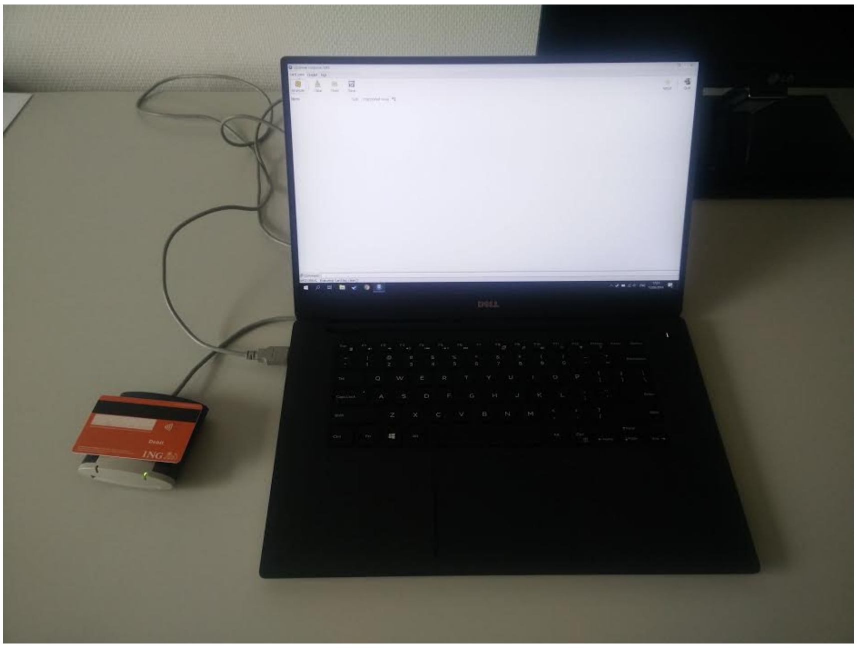


Figure 6.2: Cardpeek setup to read EMV card

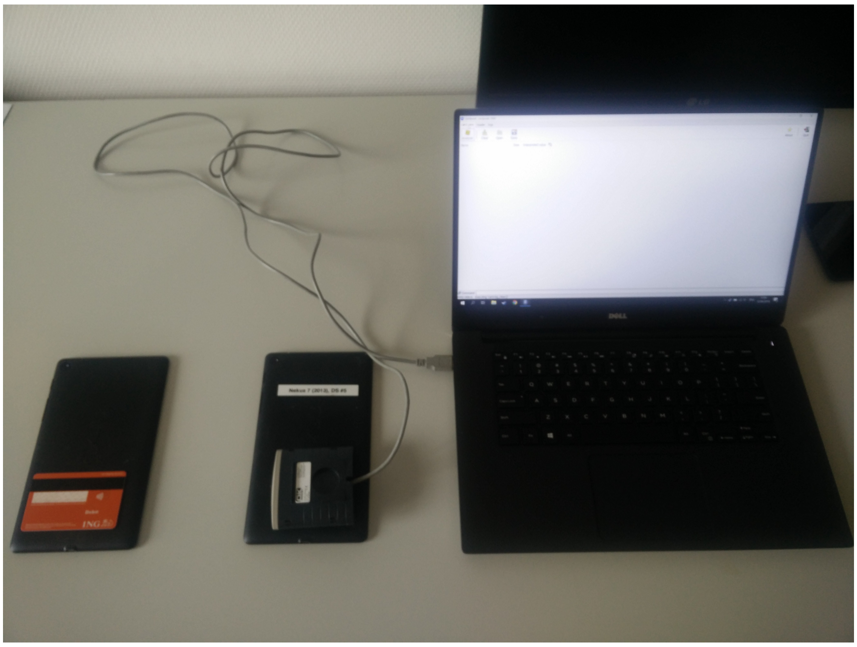
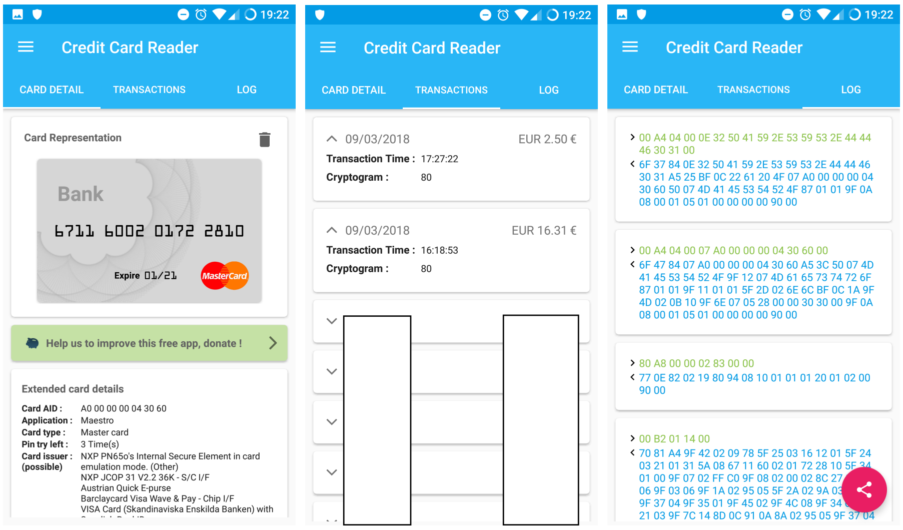


Figure 6.3: Cardpeek relayed setup to read EMV card



Card Representation

Bank

6711 6002 0172 2810

Expire 01/21

MasterCard

Extended card details

Card AID :	A0 00 00 00 04 30 60
Application :	Maestro
Card type :	Master card
Pin try left :	3 Time(s)
Card issuer : (possible)	NXP PN65o's Internal Secure Element in card emulation mode. (Other) NXP JCOP 31 V2.2 36K - S/C I/F Austrian Quick E-purse Barclaycard Visa Wave & Pay - Chip I/F VISA Card (Skandinaviska Enskilda Banken) with

09/03/2018 EUR 2.50 €
Transaction Time : 17:27:22
Cryptogram : 80

09/03/2018 EUR 16.31 €
Transaction Time : 16:18:53
Cryptogram : 80

[redacted] [redacted]

```
> 00 A4 04 00 0E 32 50 41 59 2E 53 59 53 2E 44 44
  46 30 31 00
< 6F 37 84 0E 32 50 41 59 2E 53 59 53 2E 44 44 46
  30 31 A5 25 BF 0C 22 61 20 4F 07 A0 00 00 00 04
  30 60 50 07 4D 41 45 53 54 52 4F 87 01 01 9F 0A
  08 00 01 05 01 00 00 00 00 90 00

> 00 A4 04 00 07 A0 00 00 00 04 30 60 00
< 6F 47 84 07 A0 00 00 00 04 30 60 A5 3C 50 07 4D
  41 45 53 54 52 4F 9F 12 07 4D 61 65 73 74 72 6F
  87 01 01 9F 11 01 01 5F 2D 02 6E 6C BF 0C 1A 9F
  4D 02 0B 10 9F 6E 07 05 28 00 00 30 30 00 9F 0A
  08 00 01 05 01 00 00 00 00 90 00

> 80 A8 00 00 02 83 00 00
< 77 0E 82 02 19 80 94 08 10 01 01 01 20 01 02 00
  90 00

> 00 B2 01 14 00
< 70 81 A4 9F 42 02 09 78 5F 25 03 16 12 01 5F 24
  03 21 01 31 5A 08 67 11 60 02 01 72 28 10 5F 34
  01 00 9F 07 02 FF C0 9F 08 02 00 02 8C 27
  06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03
  9F 37 04 9F 35 01 9F 45 02 9F 4C 08 9F 34
  21 03 9F 7C 14 8D 0C 91 0A 8A 02 95 05 9F 37 04
```

Figure 6.4: Screenshots from Credit Card Reader application

when available, a transaction history. The Bunq card does store and show a transaction history containing the date, time and the amount of mony of the past 10 transactions while the other tested Dutch bank cards do not. On the other hand, the bank account number of the other tested Dutch bank cards is correct but for the Bunq card it is not. One possible reason the account number for the Bunq card is not correct, is that Bunq makes it possible to use one bank card for different bank accounts, all with a unique IBAN number. See the screenshots in Figure 6.4 from the Credit Card Reader application with most of the payment history censored.

Table 6.1: Results of skimming test

Card	Correct bank account number	Transaction history available
ING	Yes	No
Rabobank	Yes	No
ABN Amro	Yes	No
Bunq	No	Yes, 10 transactions

Relay results A problem came up when first executing the relay attack. The Android phone recognized that it was talking to a payment terminal and it asked which payment app to use. In the list to choose from, there were two options: *Android Pay* and a bank app. These apps 'hijacked' the communication with the POS. To fix this problem, the Relay Emulator needed this line of code in the abduservice.xml:

```
    <aid-group android:category="payment">
```

When this was fixed, the relay worked and an actual payment was relayed.

Analysis of the results

The Bunq card stores a transaction history, which is privacy-sensitive information. The ING does not store a transaction history so that is better in terms of privacy. The ING, however stores the bank account number which is also privacy sensitive. The Bunq card does not store this number because one card can be used for many different bank accounts with each their own account number. On this part, Bunq scores higher in terms of privacy.

6.2.2 eNIK card

The eNIK card is the electronic Dutch identity card. It is a normal ID card with the addition of a Machine Readable Zone (MRZ) and a chip with antenna. The MRZ is a string printed on the card in a font easy to read for a computer. The chip in it stores among others the picture, name, nationality and the Citizen Service Number. The communication between this card and the card reader is encrypted. To communicate with the chip, the reader has to send part of the MRZ, which is the encryption key, to the chip. This is called Basic Access Control (BAC). This prevents skimming without good vision of the card, for example when it is in someone's wallet.

Properties

Supported technologies according to NXP TagInfo scan:

- Electronic identity document (MRTD)
- ISO/IEC 7816-4 compatible
- ISO/IEC 14443-3(A) compatible
- ISO/IEC 14443-4(A) compatible

Relay results

Using the Cardpeek e-passport scan option, we could relay the eNIK card but we had to provide the Cardpeek application the MRZ of the card. So the relay is possible but not without visual access to the card.

6.2.3 IRMA card

The Android application Thalia IRMA Verifier[3] can be used to verify attributes present on the IRMA card. This application can check if a certain attribute is present on the card, for example 18+.

Properties

Supported technologies according to NXP TagInfo scan:

- ISO/IEC 7816-4 compatible
- ISO/IEC 14443-3(A) compatible
- ISO/IEC 14443-4(A) compatible

Results

When executing the relay, the Thalia IRMA Verifier application responded with 'Attribute not present' while the card does actually hold the attribute ‘Age: 18+’. When holding the card directly to the reader (with the IRMA Verifier application), the application gives the notification ‘Attribute verified’.

There was data being relayed back and forth but the IRMA Verifier recorded something different than with direct communication with the card. To find out what causes this behavior, we can analyze the APDUs.

First we analyze the APDUs from when the IRMA card is directly communicating with the IRMA application. We verified the same attribute on the same card three times and compared the sequence of APDUs. We found that verifying the attribute 18+ uses nine APDUs from the application to the card and back. The first and the last two APDUs are the same each time but the APDUs in between are different. This indicates that the messages in between are encrypted based on a nonce in the second message. This should not be a reason why the relay does not work because each message can be relayed as is.

Then we analyze the APDUs that are being relayed by the relay applications. We found out that the relay stops each time during the sixth message from the card to the application. Without the relay, this message was 257 bytes each time. During the relay, this message was 1 byte. The problem here is that the relay applications store the length of the APDUs in one byte. This byte overflowed and resulted in the loss of the whole message except for the first byte.

[3] `https://play.google.com/store/apps/details?id=nu.thalia.androidverifier&hl=nl`

After fixing this bug in the Relay Reader and Relay Writer applications, the relay with the IRMA card succeeded because the Thalia IRMA Verifier verified that the attribute 18+ was on the card.

6.2.4 Radboud student card

The Radboud Student card can be used to authenticate via RFID. This is used to enter a building or get access to the fitness room. In the past, it has been used to authenticate for the printers as well.

Properties

Supported technologies according to NXP TagInfo scan:

- IC type: MIFARE DESfire EV1
- Access control and 8 unknown DESfire Applications
- ISO/IEC 7816-4 compatible
- Native DESfire APDU framing
- ISO/IEC 14443-2(A) compatible
- ISO/IEC 14443-3(A) compatible
- ISO/IEC 14443-4(A) compatible

Results

When holding the Relay device next to the card reader, the card reader did not send a 'select AID' message. Without this message, the Android OS does not realize that there are NDEF messages which are meant for the Relay Emulator application and the HCE does not activate.

6.2.5 OV-chipkaart

The OV-chipkaart is a MIFARE Classic card. This card is used to pay the tickets for the public transportation in the Netherlands. The card contains a credit and optionally a travel product. MIFARE Classic cards contain cryptographic weaknesses which makes them vulnerable to cloning attacks[7][15]. In this thesis we will not be executing these attacks.

Properties

Supported technologies according to NXP TagInfo scan:

- ISO/IEC 14443-3(A) compatible
- MIFARE Classic

Relay results

When holding the relay phone to the check-in terminal and holding the mole phone to the OV-chipkaart, the terminal gave an error and asked for a OV-chipkaart. This was an expectable result because the relay only works with applications that are ISO/IEC 7816-4 compatible.

6.2.6 MyWheels Open

MyWheels is a car sharing company in the Netherlands. They have a website where customers can offer their car and other people can book the cars. Cars with the MyWheels Open label have a board computer and they can be opened with an OV-chipkaart[4]. To do this, the customer has to book a car online, and then hold their card to the card reader. The door opens and the keys are in the car. This only works from five minutes before the booking until five minutes after the booking. For each car it is available for logged in users who booked a car at what time. On the personal settings page of the MyWheels website, we noticed a field: chip card number. The number that was showed, was the UID of our OV-chipkaart. This made us believe that the system might only use the UID of the card and thus is vulnerable to a cloning attack.

Cloning Attack on MyWheels

To test this, we cloned the UID from our OV-chipkaart to our Android device using NFC Card Emulator from Section 4.4.4. Then we booked a MyWheels Open car and tried to open it with our OV-chipkaart. This worked as intended. Then we tried to open it with our Android device without emulating the OV-chipkaart cloned UID and it did not open, as expected. Then we emulated the UID of the same OV-chipkaart and held our phone next to the reader. The car opened and we could drive away.

This shows that the only thing the reader looks at is if the UID corresponds to the UID that booked the car.

Attack scenario MyWheels cloning vulnerability

An attacker can place an RFID skimming device very close to the card reader. This records all the UIDs and the time of the customers renting the car. After a longer period of time, the attacker will have a list of what UID rented the car at what time. Using the calendar of the car, which is available after logging in on the MyWheels website, the attacker can link each UID to the name of the person renting the car. When a person from the composed list rents the car again (or another car), the attacker knows the UID of this

[4] `https://mywheels.nl/autodelen/veelgestelde-vragen/hoe-open-ik-een-auto-met-mijn-chipkaart/`

person and can emulate it with an NFC device. The attacker goes to the car and five minutes before the reservation starts, if the person renting the car is not there yet, the attacker unlocks the car and drives away.

Risk MyWheels vulnerability

MyWheels uses the UID on the RFID Chip to authenticate the renter of the car. This UID can be easily cloned. Therefore the system does not achieved what it was meant to achieve, namely authentication of the user and thus a clear vulnerability. The risk of a vulnerability is determined by a combination of the impact and likelihood of the abuse.

Factors that increase the likelihood of an attack on this vulnerability are:

- Ease of exploitation: To skim and emulate a UID, an attacker only needs a rooted Android device with the NFC card emulator application.
- Ease of discovery: The MyWheels website shows under section Chip Card only the UID of the chip card. This brings up the idea that only the UID is used, which is easily testable.

Factors that decrease the likelihood of an attack on this vulnerability are:

- The motive of the attacker: To create an account for MyWheels, users have to pay a deposit of at least 250. This could put off attackers.
- Detection: Abuse of this vulnerability will get detected relatively fast because the user who rented the car can not find it and will call the MyWheels service. They can see the car is already opened and has driven away and come to the conclusion that the car is stolen. MyWheels has track and trace devices in the car to monitor its location and they can lock and unlock it remotely.

Even after this analysis, it is hard to assess the severity of the risk of this vulnerability. We believe the risk is present.

6.3 Overview of the results

Table 6.2 gives an overview of the results. Note that MyWheels does not use their own cards but instead use cards that users already possess like the OV-chipkaart. Because of this, the entries for skimming and cloning are not filled in for MyWheels. For the cloning attack, we only tried cloning UIDs. MyWheels only checks for the UID so this is enough to perform an attack. The other cards do not only check the UID of a card so even when the UID is cloneable, this does not result in a practical attack on the system the card is part of.

	Skimming	Cloning	Relay
EMV card	Able to skim transaction history and bank number	-	Success
eNIK card	BAC prevents skimming	-	Success
IRMA card	Verify age = 18+ with Thalia Irma Verifier	-	Success
RU student card	Read basic IC information	-	Relay failed
OV-chipkaart	Read basic IC information	-	Relay failed
MyWheels card	-	Success (UID)	-
All cards	Able to read IC type		

Table 6.2: Overview of the results

The relay attack succeeded on the first three cards because they are all ISO/IEC 14443-4 compliant with on top of that using ISO/IEC 7816 'select AID' messages and without timing countermeasures. The relay attack on the OV-chipkaart failed because it is a MIFARE Classic card and does not support ISO/IEC 14443-4. The relay attack on the Radboud student card is a MIFARE DESfire card and it failed because it did not issue a 'select AID' message.

Chapter 7

Conclusions

The research question was ”How can Android NFC devices be used to compromise the security of systems that use RFID cards?” In this thesis we showed that Android NFC devices can be used for three types of attacks: skimming, cloning and relay attacks. The Sections 7.1, 7.2, 7.3 describe the conclusions on how Android NFC devices can be used for respectively skimming, cloning and relay attacks.

7.1 Skimming

With an Android NFC enabled phone, it is possible to skim RFID cards. There are general applications that can identify the IC type, manufacturer UID and other properties. There are also specialized applications for example to read out an EMV card and check if it contains a transaction history, the bank account number, expiration date and some technical specifications of that card. We found that the Bunq bank card stores a transaction history. This is surprising because that is information that is very private to the owner and not needed for its functionality as other EMV cards do not store this.

7.2 Cloning

We showed that it is possible to clone the UID of a card with a rooted Android NFC device. In Section 6.2.6 we found that the MyWheels Open cars only look at the UID and because of this are vulnerable to the abuse of a cloned UID. We were able to unlock a car and drive away with it using a rooted Android NFC device. A check of a UID is not a secure way of authentication. Active authentication or chip authentication are secure ways of authenticating an RFID card as described in Chapter 5.

7.3 Relay

We modified two parts of the relay applications. First we made it such that not only EMV cards can be relayed but other RFID cards as well. This made it possible to, for example, relay a Dutch ID card. Second, we fixed a byte overflow bug that prevented relaying of the IRMA. Only RFID cards communicating with APDUs and using AIDs according to ISO/IEC 7816-4, can be relayed with an Android device using HCE. The MIFARE DESfire are the only MIFARE cards that have the possibility to be relayed because they support ISO/IEC 7816-4 but do not use it necessarily.

It took quite some work to get the relay applications working because of practical problems we encountered. These problems were caused by the complexity of the Android OS and different versions that behave differently and by hardware problems where the Nexus 7 tablet behaves different than the OnePlus One phone with the same version of Android.

Chapter 8

Future Work

This chapter contains some unanswered questions that might be interesting for future research.

- Can iPhones be used for relay attacks, especially since Apple might open up NFC to third party developers in iOS 11[1]? Are the limitations the same as with Android?
- Are there ways to modify Android such that it can also relay MIFARE cards?
- Can Android NFC devices be used to eavesdrop?

  This is currently not possible as Android supports only three modes of operation: Reader/writer mode, P2P mode and Host Card Emulation mode.

[1] https://www.engadget.com/2017/06/06/ios-11-iphone-core-nfc-chip-more-than-apple-pay/?guccounter=1

Bibliography

[1] T. Chothia, F.D. Garcia, de J. Ruiter, van den J. Breekel, and M. Thompson. Relay cost bounding for contactless EMV payments. In *Financial Cryptography and Data Security, vol 8975*, pages 189–206. Springer Berlin Heidelberg, 2015.

[2] S. Clark. Two in three phones to come with NFC in 2018. *NFC World*, 2014.

[3] J.H. Conway. *On Numbers and Games.* Ak Peters Series. Taylor & Francis, 2000.

[4] O. Dagdelen and M. Fischlin. Security analysis of the extended access control protocol for machine readable travel documents. *Information Security: 13th International Conference*, pages 54–68, 2010.

[5] Android Developers. Host-based card emulation. `https://developer.android.com/guide/topics/connectivity/nfc/hce.html#HCE`. Accessed June 9 2018.

[6] M. Engelhardt, F. Pfeiffer, K. Finkenzeller, and E. Biebl. Extending ISO/IEC 14443 type a eavesdropping range using higher harmonics. In *Smart SysTech 2013; European Conference on Smart Objects, Systems and Technologies*, pages 1–8, June 2013.

[7] Flavio D. Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs. Dismantling mifare classic. In *Computer Security - ESORICS 2008*, pages 97–114. Springer Berlin Heidelberg, 2008.

[8] R. Habraken, P. Dolron, E. Poll, and J. de Ruiter. An RFID skimming gate using higher harmonics. In *4th Workshop on RFID Security (RFIDSec), vol 9440*, pages 122–137. Springer International Publishing, 2015.

[9] G. Hancke. Eavesdropping attacks on high-frequency RFID tokens. In *4th Workshop on RFID Security (RFIDSec), vol 9440*, pages 259–288. Springer International Publishing, 2008.

[10] G.P. Hancke and M.G. Kuhn. An RFID distance bounding protocol. *Security and Privacy for Emerging Areas in Communications Networks*, 2005.

[11] ICAO. *Doc 9303: Machine Readable Travel Documents*, 7 edition, 2015. Part 10: Logical Data Structure (LDS) for Storage of Biometrics and Other Data in the Contactless Integrated Circuit (IC).

[12] ICAO. *Doc 9303: Machine Readable Travel Documents*, 7 edition, 2015. Part 11: Security Mechanisms for MRTDs.

[13] Z. Kafir and A. Wool. Picking virtual pockets using relay attacks on contactless smartcard. *SECURECOMM '05 Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, 2005.

[14] E. Lee. NFC Hacking: The Easy Way. Presented at DEFCON 20 in Las Vegas, slides by Blackwing Intelligence `https://www.defcon.org/images/defcon-20/dc-20-presentations/Lee/DEFCON-20-Lee-NFC-Hacking.pdf`, 2012.

[15] C. Meijer and R. Verdult. Ciphertext-only cryptanalysis on hardened mifare classic cards. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 18–30. ACM, 2015.

[16] MIFARE. Mifare desfire ev1 contactless multi-application ic. `https://www.nxp.com/docs/en/data-sheet/MF3ICDX21_41_81_SDS.pdf`, 2015.

[17] MIFARE. Mifare ic product overview. `https://www.mifare.net/wp-content/uploads/2015/02/MIFARE-Product-Overview_12-2018.pdf`, 2018.

[18] NFC forum technical specifications. `https://nfc-forum.org/our-work/specifications-and-application-documents/specifications/specification-releases/`, 2017. Accessed June 9 2018.

[19] J. Van den Breekel. A security evaluation and proof-of-concept relay attack on Dutch EMV contactless transactions. Master's thesis, Eindhoven University of Technology, 2014.

[20] S.A. Weis. *RFID (Radio Frequency Identification): Principles and Applications.* MIT CSAIL, 2007.