



Mälardalen University
School of Innovation Design and Engineering
Västerås, Sweden

Thesis for the Degree of Bachelor of Science in Engineering - Computer
Network Engineering 15.0 credits

CLONING ATTACKS AGAINST NFC-BASED ACCESS CONTROL SYSTEMS

Sebastian Leclerc
sic19001@student.mdu.se

Philip Kärström
pkm19003@student.mdu.se

Examiner: Mikael Ekström
Mälardalen University, Västerås, Sweden

Supervisor(s): Shahriar Hasan
Mälardalen University, Västerås, Sweden

Company Supervisor(s): Martin Jinnestrand
Basalt AB, Enköping, Sweden

07/06/2022

Abstract

The wireless communication methods Near Field Communication (NFC) and Radio Frequency Identification (RFID) are today used in different products such as access cards, smartphones, and payment cards. An effective attack against this type of technology is cloning attacks. Cloning attacks can deceive access control systems which may cause serious damage to organizations such as information leakage and financial loss. This type of attack attempts to deceive a system with an illegitimate cloned card that may be an identical copy of all the data on a card, parts of the data, or perhaps only by using its identification number. Therefore the existing security flaws that cloning attacks exploit are an important threat for organizations to acknowledge and manage.

This thesis focuses on evaluating three different access control systems in use and demonstrates security flaws that exist in these systems. The systems are evaluated by how data can be extracted from the access control cards, this includes the time to collect all the data, reading distance, and interfering objects. Systems are also evaluated by what information the systems validate. Compatible equipment for evaluating the different systems is necessary such as readers, writers, and other penetration testing tools. The type of card that the systems use is called Mifare classic whereas two of the systems used a 1K version and one a 4K version, specifying the amount of available memory on the card itself. The equipment also made it possible to perform and verify cloning attacks through different processes such as simulation and sniffing to explore what information certain access control systems deem necessary on the access cards.

Rigorous experiments on the systems and the results reveal that crucial information on the access cards could easily be extracted, reused, and simulated for accessing two of the systems. One system proved to be more secure since it required more advanced methods to clone cards that the system accepted. The results of this thesis demonstrate that the evaluated access control systems cannot be considered secure without additional layers of security added to them, instead, it is important to keep the back-end system maintained through various applicable means.

Acknowledgment

We would like to thank Basalt AB and especially Martin Jinnestrand who has supported us throughout the thesis work and provided us with the necessary equipment. Also, a big thanks to Sara Lundahl who believed in us and is someone who is always there for her students. It's been a tough three-year period but at the same time very valuable and fun. Thanks to all our teachers for providing a good program for us, all the classmates, and long days in Netcenter. Finally, we wish to thank our supervisor Shahriar Hasan for helping us with the writing process and being there to bounce ideas.

Contents

1. Introduction	1
2. Background	2
2.1 RFID and NFC	2
2.2 General vulnerabilities with NFC and RFID	3
2.2.1 Reading, cloning and modifying cards	3
2.3 Cards, hardware and other tools	4
2.3.1 Cards	4
2.3.1.1 Mifare Classic 1K and 4K	4
2.3.1.2 Magic Cards	5
2.3.2 Readers and writers	5
2.3.2.1 RC522 RFID module and Arduino	5
2.3.2.2 ACR122U	6
2.3.3 Other penetration testing tools	6
2.3.3.1 ICOPY-XS	6
2.3.3.2 ChameleonTiny	6
2.3.3.3 RFID Diagnostic Card	6
2.3.3.4 NFC Tools and Mifare Classic Tool	6
3. Related work	8
4. Problem formulation	9
5. Method	10
6. Ethical and societal considerations	11
7. Implementation details and experiments for carrying out cloning attacks	12
7.1 Simulating an access control system	12
7.2 Cloning and evaluation of an access control system in a university	14
7.2.1 Cloning using the RC522 module	14
7.2.2 Cloning using the ACR122U	17
7.2.3 Cloning using the ChameleonTiny	18
7.2.4 Sniffing and simulating an access card	19
7.3 Keys and a public transportation system	20
7.4 Cloning and evaluation of a Mifare 4K access control system	23
7.5 Measurements and RFID protection	23
8. Results	25
8.1 University access control system	25
8.2 Public transportation system	27
8.3 Mifare 4K access control system	28
8.4 Analyzing cloning attempts	28
8.5 Evaluation of range, time and interference when cloning	28
9. Discussion	30
10. Conclusions	34
11. Future work	35
References	38
A Appendix one	39

List of Figures

1	A flowchart of the research method.	10
2	The Arduino, breadboard, and RC522 module setup with some cards.	13
3	The database user table in the simulated environment.	14
4	Simulation responses.	14
5	The extracted UID from the legitimate student B access control card	15
6	Mifare Classic Tool extraction verifying the previous dump data.	16
7	Student access cards A and B.	16
8	Student access card A data in sector 0, block 0.	16
9	Student access card B data in sector 0, block 0.	16
10	Changing the data on sector 1 on card A using the MifareClassicValueBlock script.	17
11	The ACR122U reader connects via USB to a computer.	18
12	The ChameleonTiny and its application interface.	18
13	The ICopy-XS with its protective plastic component covering the antenna.	19
14	First sniffing capture of RFID traffic.	19
15	Second sniffing of RFID traffic.	20
16	A partially censored public travel card.	21
17	Some sectors could not be read on the travel card.	21
18	Unknown keys were discovered.	22
19	Unknown keys were recovered.	22
20	A selection of the data on the travel card is saved in a JSON file.	23
21	All of the cards used for measuring.	24
22	First successful cloning onto a magic card.	26
23	Entry was granted with cloned card when presented to the university reader.	26
24	Testing requirement limitations of the card data.	26
25	Modifying the access bits resulted in strange behavior.	27
26	Attempting to clone a travel card.	27
27	RFID communication errors while reading.	33
28	Old keys did not work.	33
29	Comparison of travel card dumps.	33
30	The data structure of a Mifare Classic 1K card.	39
31	The data structure of a Mifare Classic 4K card.	40

List of Tables

1	The MFRC522 library scripts used and a brief description of their purpose.	13
2	An overview of the results of the cloning attempts.	28
3	Evaluating the maximum reading distance.	29
4	Evaluating the maximum amount of time to read all card data.	29
5	Results of using an RFID protective sleeve.	29

1. Introduction

Radio Frequency Identification (RFID) or Near Field Communication (NFC) is probably used every day without any awareness of the technology. This type of technology is widespread and incorporated in products and services such as entrance cards used to access buildings and offices, tags that protect valuable assets, and it can also be seen as a contactless payment solution in stores or when traveling by public transportation to name a few examples [1][2]. Both RFID and NFC are two wireless communication methods while NFC can be seen as a subcategory to RFID. The operating frequency used by RFID can vary, e.g between 120-140 kHz or 3.1-10.6 GHz [3]. The range of the technology is dependent on the frequency and power source. RFID using a passive power source with an operating frequency of 10.6 GHz can communicate over a range of 10 m while instead, using an active power source the technology can communicate over 100 m. Unlike RFID, NFC has a set frequency and can only communicate over a distance of about 10 cm [2]. The short-range communication of the NFC technology makes it a viable candidate for transmitting sensitive information such as bank account details or when identifying an individual accessing premises. The technology can replace many of the cards in wallets and instead be incorporated into smartphones. This however raises questions such as how safe it is to use and if any vulnerabilities need safeguarding.

There is a number of previous work that has investigated RFID and NFC from different perspectives. While researching the RFID and NFC technologies and their vulnerabilities it was therefore noted that there are many different research paths to explore. This thesis narrows down its focus on the investigation of cloning attacks. This type of attack is performed by reading the necessary data from a legitimate access control card to either write the copied data onto a new card or by simulating the copied information through a tool [4]. The ultimate goal of the attack is to trick the system into gaining unauthorized access. Therefore access control systems are vulnerable to cloning attacks to a greater extent in comparison to more sophisticated attacks. Thus, first, an understanding of the vulnerabilities that cloning attacks use and how to hinder them is necessary before delving into more advanced attacks such as relaying and code injection. This thesis aims to address vulnerabilities found in access control systems and help organizations and researchers understand and prevent threats such as unauthorized access.

To understand the structure of this thesis, a brief overview of each section is given below. In Section II an explanation regarding the RFID and NFC technology is provided. This section also covers some of the known vulnerabilities and a brief discussion regarding the topic of cloning attacks. Different hardware and tools were required to perform the various cloning experiments and these are described here as well. In Section III, some of the earlier research regarding NFC and RFID cloning and the current state-of-the-art is presented. Namely, a brief overview of what has been done earlier and explored concerning cloning attacks. The problem formulation and Research Questions (RQs) that were chosen based on the research up to this point are presented in Section IV. For each of these RQs, a brief discussion is provided to theorize how each of them best should be handled. Section V presents the chosen research method and the tools used to tackle the problem formulation and why this path was chosen. In Section VI the ethical and societal considerations for the thesis work are discussed. A description of the work that has been done and how the cloning attacks and related experiments were performed on the different systems is seen in Section VII. The three different access control systems that were tested and how the tools were used are shown here. Section VIII describe the results from the cloning attempts experiments and measurements. Section IX outlines some of the technical issues and noteworthy results. Finally, Section X attempts to tie everything together and conclude the thesis work while Section XI discusses interesting future work that could be explored.

2. Background

This section outlines the background information. It will cover some of the technical details of the RFID and NFC technologies, some known vulnerabilities, and the type of hardware and tools used during the implementation phase.

2.1 RFID and NFC

Radio Frequency Identification (RFID) is a wireless technology that often uses (smart) cards or tags¹ together with readers and possibly an underlying back-end system [3]. These cards can either be active with their own power source or passive without any power source. When the cards are passive, the power is gained from the reader's transmitted energy. RFID technology can operate using many different frequencies such as 125 kHz, 13.56 MHz, or even 2.45 MHz. RFID technology is widespread in our society, for example within the domains such as logistics, healthcare, military, etc. RFID can also be seen in many different situations such as when tracking animals, automated toll systems on roads, or even as airport radar when given enough power [2].

Near Field Communication (NFC) is a subcategory of RFID that also communicates wirelessly but at a shorter range [5]. NFC devices can both act as readers or emulate cards. The prominent use of NFC can be seen in smartphones with contactless payments. It can also be seen with access control systems such as when accessing rooms and offices in a business. It is mainly used with systems that carry sensitive data. The biggest benefit of using NFC over other wireless communication protocols is its simplicity [6]. One example is when using contactless payments, the transaction is initialized automatically by touching the reader with a card.

NFC has three modes of communication which are: Peer-to-peer, Reader/Writer mode, and Card Emulation mode [1].

- Peer-to-peer: Bi-directional communication between two active devices. However, the limitations are high protocol overhead and the use of proprietary libraries.
- Read/write mode: Uni-directional data transfer between, for instance, a passive card and an active reader.
- Card Emulation mode: Smartphone device may act as a contactless smart card when used in card emulation mode. This is frequently used as a payment and ticketing system on smartphones.

There are many standards under the RFID and NFC umbrella for different types of implementations. A common standard seen with contactless smartcards is ISO 14443 [7]. This is an international standard that describes the communication stack of contactless smartcard systems, for example, the radio frequency and transmission protocol.

Since both technologies are similar to each other, it is important to understand some key differences between NFC and RFID [2]. NFC must communicate in close proximity of approximately 10 cm while using a fixed frequency at 13.56 MHz as opposed to RFID which is more flexible and the frequencies can vary. Depending on the power output and frequency that is used, RFID can communicate up to approximately 100 km in extreme cases when given a high amount of power. Therefore, NFC can be seen as a more secure solution against attacks such as eavesdropping and man-in-the-middle due to the shorter range it facilitates. Even though differences exist it should be known that many developers, vendors, and literature use the two terms, RFID and NFC, interchangeably which may cause some confusion.

¹From this point onward, the term card will be used to refer to tags as well.

2.2 General vulnerabilities with NFC and RFID

A large part of the security with NFC lies under the assumption of a limited communication range between a card and its reader [8]. NFC cards can communicate wirelessly with readers over a distance of between 5 to 10 cm. This can be seen as a form of security through obscurity where the security mainly lies in the secrecy between the card and reader, i.e. no one else should be able to listen. This is generally considered a bad security principle to follow when for example looking at recommendations from the National Institute of Standards and Technology (NIST) [9].

Many of the well-known threats to generic computer systems can also be found when looking at RFID and NFC. For instance, sniffing, spoofing, replay and relay attacks, tracking, denial of service and malicious code [10]. Since the communication between the card and reader is wireless it is possible to intercept the sent messages in transit as a man-in-the-middle attack. The purpose here might be to relay these messages to another device in form of a relay attack, to spoof a system using the captured messages, to manipulate the data before re-transmission, or simply to log and save the data. There is also the denial of service types of attacks that attempt to tamper with the overall functionality of a system. The latter might be accomplished by signal jamming or with code that for example shuts a back-end system down. Due to the short-range nature of NFC, there have been attempts to increase this range by researchers [8]. This can for example be done by developing more powerful readers or by attempting to amplify the signal in some way.

2.2.1 Reading, cloning and modifying cards

One effective attack against systems that use RFID or NFC technology is called cloning attacks which in essence attempts to maliciously steal the information stored on legitimate access control cards that the access control systems accept [11]. Typically the technology is implemented as the communication method between the legitimate access cards and the reader of the system protecting access to premises or services. Cloning attacks against RFID and NFC are performed by an attacker first copying the necessary data, or parts of it, from a legitimate card [4]. The copied information can then either be written to another card or simulated with different tools. The duplicate card, or simulated information, can later be used in the same fashion as the original card and show similar or exact characteristics. A clone attack threatens RFID and NFC systems that only use the uniqueness of a card to verify the owner of the card. RFID and NFC systems are widely used with access control systems in different facilities or services. A cloned card can allow an unauthorized person to get access to a facility or service if no other authentication but the card is needed. This means that cloning attacks can cause serious damage to an organization such as sensitive information being leaked or financial losses. Several different methods such as eavesdropping and reading devices are needed during a clone attack to obtain the data from the original card and create a copy. Eavesdropping refers to when someone in the vicinity of a device tries to intercept or receive the information between the communication channel [1]. Reading devices such as Proxmark² are needed to write the information from the original card to another card [4].

It is also possible with NFC to store data in the form of NFC Data Exchange Format (NDEF) records [12]. Different hardware vendors may use different types of records. To present examples, Android smartphones can use records to exchange information between devices over WiFi, the records can be used to store an URL or used to store plain text messages. The records may also be chained together for a larger amount of information and stored on sectors on Mifare cards.

Two approaches that attempt to impede or hinder cloning attacks are: prevention and detection [13]. Prevention methods are trying to provide different security measures such as cryptography and encryption technology to smart cards. But this approach has not stopped the cloning attacks completely and it is a problem with low-cost cards due to the storage and power constraints. Detection methods are instead trying to identify the cloned cards via different checks [14]. Therefore, detection is the more suitable approach for low-cost cards. Detection relies more on pattern matching and machine learning rather than encryption to detect a counterfeit card.

²Proxmark is a well-known tool for RFID security research.

2.3 Cards, hardware and other tools

The following section describes the different cards, hardware, and tools that were used during the implementation and testing phase.

2.3.1 Cards

There are many cards and tags available on the market. Some are standardized and follow strict protocols and guidelines whilst others are lesser copies or use black box type of systems. To go more in-depth regarding possible security flaws some technical knowledge is required regarding how the cards are structured. Here follows a brief background regarding the different access card types that were used in the experiments.

Cards can store a certain amount of data on their chips, on passive cards this is usually stored and organized in different sectors and blocks [10]. The exact amount of total storage differs for different types of cards, but to provide some estimation it is within the ranges of 128 B to 128 kB [15]. The sectors and blocks, or parts of them, may be read-only, write-once, and read-multiple or write-multiple and read-multiple. A typical RFID system first reads the data from the card to its RFID reader. The reader then processes the data with some middleware application. This step is done to for example authenticate the card. If everything is in order, it may then be sent to a back-end system for logging, as an example. The majority of the contactless cards used today are based on the ISO 14443 standard which is intended to work at a distance of less than 10 cm [16].

It is difficult to summarize and generalize common characteristics that our cards and tags have. However, one important characteristic that all the cards had that were encountered during this thesis was their unique identification number or ID [17]. This ID can look different on different types of cards but in general, it consists of a value that the manufacturers set during the production of the card. Most of these IDs consist of 4 bytes and because of this restriction, not all card IDs are unique after having been mass-produced.

2.3.1.1 Mifare Classic 1K and 4K

The smartcard system vendor NXP Semiconductors is one of the world's biggest suppliers and a majority of their products are based on the previously mentioned ISO 14443 standard [18]. They supply three-quarters of the world's contactless cards [19]. One of their first smart cards is the so-called Mifare Classic products that come with between 1-4 kB of available memory on the cards and a Unique Identification Number or Unique ID (UID) to identify a particular card and a certain card lifetime [20]. The UID can be found on sector 0 block 0 on the card in 8-digit hexadecimal notation [21]. It varies in size depending on the type of card, however, for Mifare 1K and 4K, it ranges between 4 and 7 bytes. For an overview of the cards structure see Appendix one. All the information stored in a Mifare 1K and 4K card is organized by the Electrically erasable programmable read-only memory (EEPROM) [22]. It contains 16 sectors where each consists of 4 blocks and one block can store 16 bytes of data. The UID is a manufacturer-written and locked block on the cards which are part of their security. The Mifare cards use a stream cipher called CRYPTO1 between the card and the reader for authentication and encryption that has been reverse-engineered so that the secret keys are easy to discover [23].

To read and/or write data on a Mifare card sector, two individual keys are required known as "Key A" and "Key B" [11][22] whose locations are illustrated in Appendix one. These keys consist of a 6 byte hexadecimal string and are located in the sector's last block called the trailer, one such example can be seen below. Here the access conditions are also specified for the sectors blocks in the form of access bits, i.e. what operations can be performed on that particular block such as read, write, increment and decrement, etc. With the help of a brute force attack, the keys can be found within milliseconds according to Jain et al in [11]. Once it is possible to read the information stored on the card, it is also possible to alter and copy most of the data. There are also several other vulnerabilities that we will briefly summarize in Section III. Instead of the Classic cards,

Mifare has developed different newer types of smart cards with stronger security, based on some form of encryption as one example.

Mifare Classic 1K and 4K factory default "Key A" and "Key B":
FF FF FF FF FF FF

To write a new UID to a card, a Block Check Character (BCC) value must be taken into consideration [21]. This is because an invalid BCC value will cause certain cards to be bricked. This means that the card is no longer usable. The BCC value is calculated by performing some XOR logic operations on the UID value and is used as a form of checksum for validating the card. It can normally be found on the fifth byte on a Mifare 1K card, see Appendix one.

Another security measure that many ISO 14443 cards have, such as the Mifare cards, is a Select Acknowledge (SAK) value [24]. They consist of a 2-digit hexadecimal read-only value typically found on the sixth byte on a Mifare 1K card that the manufacturers set during production, see Appendix one. The manufacturers set the SAK depending on what type the card is e.g. the SAK for Mifare Classic 1K is 08 and the SAK for 4K is 18 in hexadecimal [25]. The values can be used to prevent cloning and as a detection mechanism for many readers.

2.3.1.2 Magic Cards

After the popularity of Mifare Classic cards came to be, manufacturers began creating chips that were capable of forging the previously manufacturer unique UIDs [26]. The first generations of Mifare Classic had a special unlock sequence to be able to write to these locked fields. These unlock sequences eventually became well-known and used to forge cards; in response to this, the sequence was also used as a detection mechanism for cloned cards. This in turn spurred new generations of magic cards that allowed writing data anywhere on the cards. This includes setting a custom UID, BCC, and SAK. There are many different types of magic cards that can be used to clone UIDs and re-write UIDs onto the same magic card. Here, it is implied that cloning all the data from a 4 kB card at least requires the equivalent memory space on the magic card. The structure of the data in other types of cards than Mifare may also differ and therefore the equivalent type of magic card is required, e.g. to clone a Mifare card a Mifare magic card is needed.

2.3.2 Readers and writers

During the experiment different types of readers/writers and related equipment were needed to clone cards and perform the different experiments. These different tools will be discussed in this section.

2.3.2.1 RC522 RFID module and Arduino

The RC522 module is a 13.56 Mhz RFID reader/writer which is often used in attendance systems[27]. It is designed to specifically communicate with ISO 14443 systems. The reader can typically gather information from a card at a range of a maximum of 5 cm but depending on the antenna size and tuning the range can vary. The reader can only read passive tags that operate on the 13.56 Mhz frequency. This module has many publicly available libraries which makes it rather easy to install and use and complements Raspberry Pi or Arduino projects.

The Arduino is a simplistic micro-controller board with basic capabilities such as input and output, a reset button, and a small amount of processing power and memory capabilities[28]. It is marketed for entry-level engineering projects, as an educational tool and is also suggested to be used in other development projects. There are many different Arduino boards and kits available for different needs. One way of creating projects with the Arduino is in offline mode via a USB-B cable. For this to work some drivers must be installed on the connected PC along with an Integrated Development Environment (IDE) to upload C-like code to the Arduinos 32 kB flash memory.

2.3.2.2 ACR122U

The ACR122U is a contactless smart card reader and writer that supports Mifare, ISO 14443 cards and others [29]. The device can effectively read and write to various NFC cards. It offers different features such as a Read/Write speed of up to 424 Kbps, and a built-in antenna that can read cards up to a distance of 5 cm depending on the card type. ACR122U is using common drivers which are called CCID and PC/SC to operate, which already are built inside the Windows Operating System. The ACR122U can be connected to a PC through a USB cable. When the device is connected to a PC it will serve as an intermediary device between a contactless card and the computer.

2.3.3 Other penetration testing tools

There are many penetration testing tools available on the market for RFID and NFC technology. These are often more specialized with advanced functions. For the European market, one distributor of such tools we used is called Lab401.

2.3.3.1 ICopy-XS

One powerful standalone tool is called the ICopy-XS [30][31]. This is a handheld device that is based on Proxmark software. Proxmark was originally developed in 2009 as a multi-purpose low- and high-frequency tool to for example analyze signals, eavesdrop on traffic or simulate cards. It has since then been further developed into Proxmark3 with additional support for different ISO types and more capabilities. The ICopy-XS supports a large number of cards and tags with a build dictionary for already known passwords/keys. It can also be connected to a PC and be used with command-line tools, many of which can be found on the devices' memory.

2.3.3.2 ChameleonTiny

The ChameleonTiny is a powerful and flexible tool that can be used for RFID and NFC security analysis [32]. It can create clones of many of the existing cards used today. The tool can be used for many different attack scenarios such as replay, relay, and sniffing communication. The tool provides human-readable commands to configure its behavior and settings. The ChameleonTiny can internally store up to eight different contactless cards, each configurable or acting as an active NFC reader. The ChameleonTiny may also be used with an application on an Android smartphone that connects via Bluetooth.

2.3.3.3 RFID Diagnostic Card

The RFID Diagnostic Card is a simple credit card-sized tool that can aid in easily identifying whether NFC or RFID reader or terminal is using high- or low frequency for communication [33]. The card has two different LED indicators that light up with increasing intensity as more power is gained from the transmitting reader or terminal.

The LEDs can also be used to indicate the duty cycle of a reader or terminal [33]. When the RFID Diagnostic Card is placed on a typical Android Phone the LED will flash on and off approximately every half a second. This indicates that the duty cycle is checking for cards every half a second. In comparison a typical Mifare reader will make the LED light-up constantly, indicating that there is no duty cycle and the reader draws more power to always be available.

2.3.3.4 NFC Tools and Mifare Classic Tool

NFC Tools is a smartphone and PC application that can be used to write and read NFC cards [34]. NFC Tools can read and recognize several different types of cards from different manufacturers. The information given when reading a card is data such as the card type and manufacturer, information regarding the memory, and if the tag is writable or not. The write function allows for storing NDEF records such as contact details, a phone number, or even a location on a card. The writing functionality can also be used to program NFC cards to automate actions, e.g. setting an alarm or controlling the volume on a phone after scanning the programmed card.

Mifare Classic Tool is a smartphone application that can be used to interact with Mifare Classic cards [35]. The application offers different features such as reading, writing, and cloning. The tool is designed to be user-friendly for users with little knowledge about this type of technology. To read any Classic cards, a specific key is needed for the card. The application comes with some standard keys and a list of well-known keys. The application can write to magic cards which allow for writing to the first block and sector and can therefore fully clone a Mifare card.

3. Related work

There has been significant research regarding the security of RFID and NFC in the past couple of years. As previously explained, this thesis mainly deals with cloning attacks, and the state-of-the-art works related to these topics are discussed in this section. Several related works have succeeded in carrying out cloning attacks, e.g. [23], [36] and [19].

Garcia et al. showed that cloning attacks on Mifare classic cards were fast enough to allow an attacker to wirelessly "pickpocket" a victim's card [23]. In other words, they showed that a cloning attack can be performed by standing next to a victim's card and within a few seconds an access card can be cloned. This was suggested based on four different attacks, each with a different set of circumstances. In contrast to the previously suggested cloning attacks, Garcia et al. only needed access to a legitimate card to clone the content. To tackle the vulnerabilities they suggested system hardening procedures relating to cryptography, diversification of Mifare keys, and integrity checks in the back-end system.

Pereira et al. explore the vulnerabilities using RFID technology in universities [36]. They present several problems regarding RFID that can compromise the effective use of the technology as an authentication method, such as the lack of awareness among IT professionals. The authors show that it was quite straightforward to clone and thus exploit the RFID cards of the students at a university. With low-cost hardware and open-source software, it was possible to clone an ID card. To carry out the cloning attack they used a Raspberry Pi and an RC522 module. The suggested system hardening improvements that Pereira et al. suggested were similar to those of Garcia et al. such as extending the functionality of the cards by adding data onto the unused sectors and blocks.

Abellon et al. focused on the risks of leaving Mifare cards out in the open and therefore being vulnerable to cloning [19]. As mentioned in Section II Mifare is a well-known brand of contactless cards. The Mifare cards are also fairly known for their vulnerabilities which can enable an attacker to both read and potentially write to the cards. The authors performed an attack by reading the contents on a Mifare Classic 1K card and then copying a UID. This was performed with a Prox-grind Chameleon Mini-rebooted, the RC522 module, and a UID-writable card e.g. a magic card. The cloned card was tested on a simulated access control system which consisted of a CH340G Arduino board, the RC522 module, and some LEDs to represent if access is allowed or not. The cloning attack was successful since the access control system acted in the same fashion as it would with the original card. Abellon et al. recommend testing these attacks against actual systems and also testing the possibility of reading multiple cards at once. This is because people normally tend to store multiple cards stacked in a wallet.

In comparison to these three related works, this thesis has evaluated and compared three different access control systems in use and performed measurements of variables that determine a successful cloning attack. The experiments used some of the same equipment, but this thesis also compared these tools to different specialized penetration testing tools. The underlying technology is the same and the type of cards have been from the Mifare Classic line of products throughout this thesis and the related works.

4. Problem formulation

RFID and NFC technologies are widely used in our daily lives which is already discussed within the scope of Section II. The technologies are often used as a payment solution in stores and with local transport systems. Organizations also use RFID and NFC technologies with access control systems for accessing their premises. There are several vulnerabilities with these technologies that expose them to threats such as relaying, sniffing, and cloning attacks. This thesis focuses on the cloning attack applied to access control systems and transportation systems. Cloning attacks can deceive access control systems by using a counterfeit card to gain access to a facility or service. Due to the vulnerabilities and the possible gain, the RFID and NFC technologies are lucrative targets to exploit. The goal of this thesis is to evaluate cloning attacks, compare them to state-of-the-art work, and compare different access control systems using various hardware, tools, and scripts. This is to gain a better understanding of the security flaws that can be observed today in real-life scenarios. The aim is to acknowledge that a threat exists and help organizations prevent the threats like unauthorized access which consequences may lead to economical loss. Considering the problem statement defined above, we can formulate the following Research Questions (RQs):

- RQ1: How can data be extracted from access control cards with the equipment used?
- RQ2: What parts of the extracted information are validated in an access control system?
- RQ3: Can any system hardening suggestions be identified for the potential weaknesses?

To answer RQ1, the equipment described in Section 2.3 will be used such as the RC522 module to read and write and the ICopy-XS for cracking and other tasks. Given the background information of how the technology works and the related work, the first sub-goal is to understand how data can be extracted from different cards available to us. For this sub-goal, different variables control the limitations of what data can be extracted and how it can be extracted. Considering real cloning attack scenarios these variables include distances, time, card location, keys, etc., these will therefore be addressed. As different systems can use different types of cards comparison of cards in use and different types are also considered.

To address RQ2, work is conducted with certain access control systems in a university, in an organization, and in a public transportation system. The assumption is that system vendors do not publish all technical details of how their product works, as this would only make the systems more vulnerable to would-be attackers. Therefore, given how and what data can be extracted in regards to RQ1, RQ2 aims to investigate if the readers, middleware, or back-end components of an access control system can differentiate between a real and cloned card or limitations there within.

To tackle RQ3, an analysis of the results was conducted and then compared to the various hardening suggestions from the related work aiming to address the vulnerabilities. Assuming that no system can be fully protected, various tactics to impede or hinder an attacker are therefore discussed. This is necessary to attempt to fulfill the aim of the thesis, namely to aid in mitigating the prevailing threats.

To gain a better understanding of the underlying technology a simulated access control system will be configured with available hardware. Copying or modifying cards will not be considered for contactless payment systems such as those using common credit cards. As stated in the previous section, vulnerabilities can be exploited in several different ways and therefore a limitation here will be by only using suitable methods with the available tools. For example, when carrying out the cloning attack attempts, smartphone tools and scripts will be used in a non-destructive and non-harmful way. This means the creation of new more advanced exploits is beyond the scope of this thesis. The budget for equipment and materials for this thesis is also limited and evaluating middleware or back-end systems is also beyond the scope. Instead, the focus will lie on the cards themselves and how their respective readers respond to various experiments.

5. Method

This section presents the research method applied to this thesis work. The research method was first approached by gathering information to understand the underlying RFID and NFC technologies and evaluate the possible vulnerabilities to examine further. An overview of different related works was then studied from the possible interesting topics to gain a better understanding of those existing vulnerabilities, available tools, and recommendations for future work. Further information gathering was then conducted to attempt to understand the state-of-the-art for the chosen topic. Once this research had been conducted a discussion regarding identifying possible research gaps was initiated. Here, it was also noted that some hardware and software components were required to perform the experiments and evaluate the systems and cards. With the aid of a company, some useful hardware was then identified and ordered. Some time was spent understanding how these different tools work and their limitations. Due to the nature of our experiments, some of the results formed new ideas and thoughts which in return caused a need for other experiments to be performed and additional information to be gathered. Therefore the research method was performed cyclically as described in Figure 1.

The experiments were conducted with help of various tools that had been purchased. The decision regarding which tools to purchase was decided based on the access control systems and cards that were accessible to the thesis work. A brief control of our cards showed that, despite all the security concerns, a majority of our cards were from the Mifare Classic line of products. We have also seen several sources such as [23], [36] and [19] that show how common they are in our society. Therefore the majority of the equipment that was acquired was chosen for its compatibility with the Mifare line of products. Tools such as the RC522 module and ACR122U were used to read from- and write to Mifare cards. To experiment with writing to cards and cloning them, compatible magic cards were also chosen. Different applications on a smartphone were used for verification and measuring purposes, but also tested to perform cloning attacks. Other useful tools such as the ChameleonTiny and ICopy-XS were acquired which offered different functionality such as cloning, sniffing and much more as earlier mentioned in Section II. When attempting the first cloning attacks it was conducted on a simulated access control system environment. This approach was performed to acquire a better understanding of the technologies before attempting the attacks on systems that were in use.

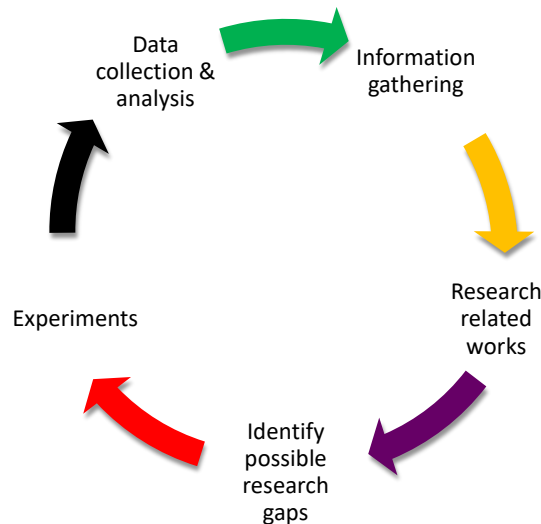


Figure 1: A flowchart of the research method.

6. Ethical and societal considerations

As discussed cloning attacks are a threat that attackers perform by copying information from a legitimate access control card to gain unauthorized access or cause economical loss. These types of access control systems can be found in organizations, on public transportation systems, or even used by individuals as a part of their home security systems to present a few examples. The knowledge about the underlying technologies and how secure a particular system is not well-known as the vendors do not publish technical details of their systems; these systems can therefore be viewed as black box types of systems. Since the purpose of these systems is to protect premises and assets there are many societal considerations that could affect many different areas in our society where these systems are used. During the research of this thesis different systems were encountered in various areas in our society ranging from critical systems controlling the physical access to the heating systems within a municipality to lesser impacted systems protecting groceries and clothes.

During the implementation phase of the experiment, it is important that no unauthorized individuals can access the collected data. This information will only be used for research purposes and will be deleted or protected after being analyzed. Any sensitive information that is extracted when reading the various access cards will be censored to protect the various organization and cardholders. The main focus of the thesis lies in exploring and comparing vulnerabilities of different access control systems, therefore to whom the systems belong will be anonymous. However, the findings of the research will be published for future researchers and organizations to understand system limitations and possible security hardening procedures. The aim is to reveal the prevailing threats and to create awareness of how to impede or hinder cloning attacks based on RFID or NFC technology. To induce reproducibility, the details of how the experiments are conducted will be shared throughout this thesis as well.

7. Implementation details and experiments for carrying out cloning attacks

In an attempt to answer the different research questions, information needs to be gathered and various experiments are performed with the different tools mentioned earlier. This section is covering a description of all the experiments. It is first divided into four subsections for each of the access control systems. These include a simulated environment, a university system, a public transportation system, and a Mifare 4K access control system. The fifth and last section covers the different measurement experiments.

7.1 Simulating an access control system

Before exploring cloning attacks on access control systems in use the thesis has decided to build a simulated system to gain knowledge regarding how a system could operate in a realistic environment. This will be simulated using equipment compatible with the Mifare 1K card type. The simulated environment is set up with the help of the RC522 module which is connected to an Arduino Uno Rev.3 SMD with a breadboard and jumper wires according to specifications seen here [37], see Figure 2 for the setup along with some of the legitimate and magic cards. Once the Arduino is set up it is connected to a computer, then the Arduino software along with its associated drivers are downloaded from the official website [38]. With the Arduino software installed, different libraries can be downloaded and tested from various GitHub projects. Therefore research is conducted to decide which library would be the best fit for cloning Mifare cards. The conclusion is that the library called MFRC522 would be suitable [39]. This is due to the library seemingly being one of the more popular libraries used in many different projects with a history of over three years of updates and bug fixes. As with many Arduino libraries, the MFRC522 library contains many useful pre-made example scripts used for analyzing some of the Mifare type of cards, an overview of the ones used in this work can be seen in Table 1. Initially, in our case, there were some troubles with getting the RC522 module and sample scripts to work as they only print connection errors to the terminal. It was discovered that this was due to a faulty connection on this particular RC522 module that had to be soldered and, after each pin connected properly the scripts started working.

After the Arduino and RC522 module is set up and the RC522 module can gather information from the cards with the MFRC522 library, a program called XAMPP is downloaded from the official website found at [40]. XAMPP is an open-source web server that provides necessary software components for developing MYSQL and PHP projects with an Apache webserver. With XAMPP a local database is created with a table that contains the information that can be seen in Figure 3. Once the database is configured the thought is to make the information on the cards be compared with the table information on whether access should be granted or not. To achieve this, the first step is through writing the extracted information from the cards to a file. The Arduino software does not have the functionality to write to a file, therefore another application called CoolTerm is downloaded. CoolTerm is a terminal application that can exchange data with hardware connected to a serial port, e.g. together with the Arduino [41]. CoolTerm is using the scripts by the MFRC522 library to capture and write the gathered serial communication to a file. Since a card contains several bytes of data, an idea is to only write the first four bytes of block 0 where the UID resides to a file. This is because the UID is considered to be the most vital data on the cards as discussed in Section II. To accommodate the needs of this project the ReadNUID script from the MFRC522 library is modified to only read and write the UID to the terminal which CoolTerm captures to a file. A PHP script is then created to first query the database with the latest captured UID found in the file and then print the result based on the query. Finally, a Bash script is executed continuously to control when the UID file is updated and then it runs the PHP script when a file change has occurred, e.g. when a new user has presented a card to the reader. The simulated environment behavior can be described in the following fashion: the user places a card on the reader, the reader sends the UID information from the card to a file with the help of CoolTerm, the Bash script notices that the file is modified and then runs the PHP script to query if the UID exists or not in the local database. If the UID exists the card has access otherwise it is denied

whose responses can be seen in Figure 4.



Figure 2: Here is an Arduino Uno Rev.3 SMD connected with seven jumper wires to the breadboard with the RC522 module. The jumper wires are connected according to [37]. Furthermore, depicted are the two magic cards, Card A and Card B, along with the legitimate student B access control card.

Scripts	
ReadNUID	Reads the UID on a card to the terminal.
DumpInfo	Reads all the sectors and blocks on a card to the terminal.
ChangeUID	Writes a new UID to a card.
MifareClassicValueBlock	Writes new values to any block in any sector except the trailer blocks.
rfid_default_keys	Reads a block with the help of specified keys.
ReadAndWrite	Writes to a specific block with values chosen by the user.

Table 1: The MFRC522 library scripts used and a brief description of their purpose.

				ID	FNAME	LNAME	UID
<input type="checkbox"/>		Redigera		Kopiera		Radera	23 Tony Stark C577F8A3
<input type="checkbox"/>		Redigera		Kopiera		Radera	25 Bruce Wayne C5A3E955
<input type="checkbox"/>		Redigera		Kopiera		Radera	26 Frodo Baggins 31D759FF
<input type="checkbox"/>		Redigera		Kopiera		Radera	27 Luke Skywalker F4A9B033
<input type="checkbox"/>		Redigera		Kopiera		Radera	28 Jack Sparrow 2DF70BE2

Figure 3: The MYSQL database table of users who has access in the simulated environment. Note that these are not authentic UIDs that we have encountered.

```

:~$ ./bash.sh
Welcome in Tony (UID : 'C577F8A3')
Welcome in Jack (UID : '2DF70BE2')
You are not Welcome (UID : '12312312')
Welcome in Jack (UID : '2DF70BE2')
You are not Welcome (UID : 'FFFFFFF')
Welcome in Bruce (UID : 'C5A3E955')
You are not Welcome (UID : 'C5A3E956')
Welcome in Tony (UID : 'C577F8A3')
Welcome in Frodo (UID : '31D759FF')
You are not Welcome (UID : '31D759F8')
You are not Welcome (UID : 'F4357798')

```

Figure 4: Responses in the simulated environment when access cards were presented to the RC522 module.

7.2 Cloning and evaluation of an access control system in a university

The first cloning attacks on systems in use are conducted with an access control system in a university that is using Mifare Classic 1K cards. At the university, there are many rooms where an access card is required. Some of the rooms require both an access card and personal identification number (PIN) authentication e.g. student computer rooms while other rooms only require a card as authentication to gain access to the room. An access card can also be used for different services such as for printers and copiers where the access card can be used for charging a print job for example. In this section, different tools and methods are described how the system is evaluated.

7.2.1 Cloning using the RC522 module

To evaluate the university access control system using the RC522 module, the same hardware and setup are used as seen in Figure 2. With the help of the MFRC522 library, one legitimate access control card, and one magic card the cloning process is attempted in two steps. Both of the cards are of the Mifare 1K type which this system uses. The cloning is first performed by executing the DumpInfo script given by the library and then by placing a card on the RC522 module to read and dump the information to the terminal. After the script is executed the UID from a card is presented in the terminal in hexadecimal notation as "C5..." along with all the data stored on the sectors, see Figure 5 for an extract of the terminal output. The validity of the previously extracted information is also verified with the Mifare Classic Tool application as seen in Figure 6 where the first two sectors are shown in hexadecimal notation and the same UID is displayed. With the UID information, the ChangeUID script from the library is ready to be executed. Before executing the script the previously extracted UID from the DumpInfo script is written to the ChangeUID script. When the ChangeUID script is executed and a new card is placed on the reader it writes the new UID to the card in sector 0. The results of the cloning attempt can be seen in 8.1.

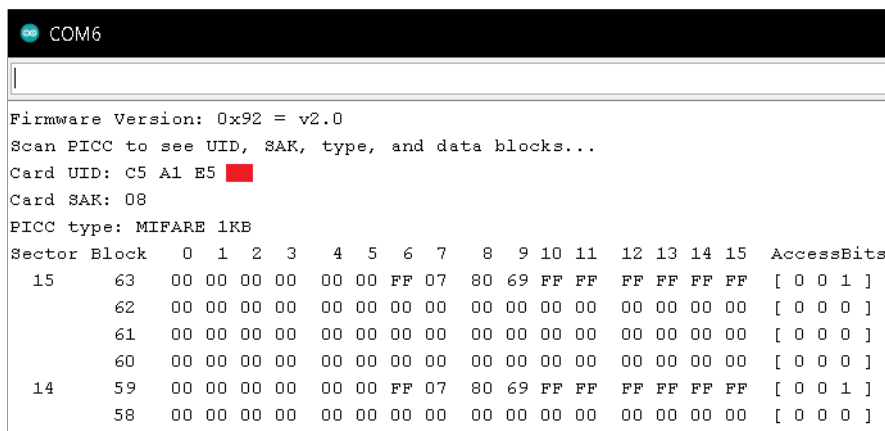
To further test the access control systems limitations, two access control cards with the same access rights are used, namely student access cards A and B which can be seen in Figure 7. After analyzing the information via the DumpInfo script it was observed that the only differentiating

information between student access cards A and B are stored in sector 0, block 0 according to the terminal output as seen in Figure 8 and Figure 9. This introduced a hypothesis of exactly what parts of the access control cards can be modified while still working as intended. In other words, does the access control system validate anything besides the information on sector 0, block 0?

To test this hypothesis an iterative process is implemented to overwrite, or scramble, more and more data on a cloned access control card while continuously controlling when the functionality breaks i.e. when access is no longer granted. This is tested with the MifareClassicValueBlock script and the previously mentioned ChangeUID script. The card chosen for this experiment is the magic card A, as depicted in Figure 2, which now contains a copy of the information of student access card A. The MifareClassicValueBlock script is used to change the data on all the blocks correlating to sectors 1-15, the structure of the data can be seen in Appendix one. The script changes the data by adding and subtracting the existing values in the blocks and then overwriting the data onto the card, see Figure 10 for one overwritten sector. After every couple of overwritten sectors and blocks, the card is tested with the reader. Eventually, only the data on sector 0, block 0-3 remains intact. Subsequently, the same method is used to modify the data on blocks 1-3 in sector 0. Finally, the only unmodified data is located in sector 0, block 0 on the magic card where the unmodified UID, BCC, and other values are stored such as on student access card A originally.

From here the same method is attempted again to scramble the final data on sector 0, block 0 using the MifareClassicValueBlock script, this is however rejected by the code. Subsequently, the ChangeUID script is modified to successfully write 15 bytes to the magic card along with the correct UID. Attempting to use the DumpInfo script to read the now completely scrambled data failed and in fact, the card is no longer recognized by the Mifare Classic Tool or the access control system at all. It turns out that the card is bricked and in the worst-case scenario no longer usable. Fortunately, the ICopy-XS features an erase function that can restore some bricked cards. Using this function the card is completely wiped and given a completely new UID, corresponding BCC, and pseudo-manufacturer data in block 0. At this point, the ChangeUID script is executed once again to re-write the correct UID onto magic card A without any issues.

To further test modifying the information on the card, the ReadAndWrite script is used. In contrast to the previous experiment, it was later observed that the trailer blocks on all the sectors were not modified by the MifareClassicValueBlock script since information in the sectors still could be read as usual by the MFRC522 scripts. This indicates that the keys and access bits are unmodified. The ReadAndWrite script can however modify all the data by manually entering which block to modify and what data to write, including the trailer blocks. This is therefore used to overwrite the three trailer blocks of sectors 0-2. The location of the access bits and keys is illustrated in Appendix one. The results of these iterative overwrites are described in 8.1.



```
COM6
Firmware Version: 0x92 = v2.0
Scan PICC to see UID, SAK, type, and data blocks...
Card UID: C5 A1 E5 
Card SAK: 08
PICC type: MIFARE 1KB
```

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
14	59	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]

Figure 5: The extracted UID from the legitimate student B access control card consisting of C5 A1 E5 ...

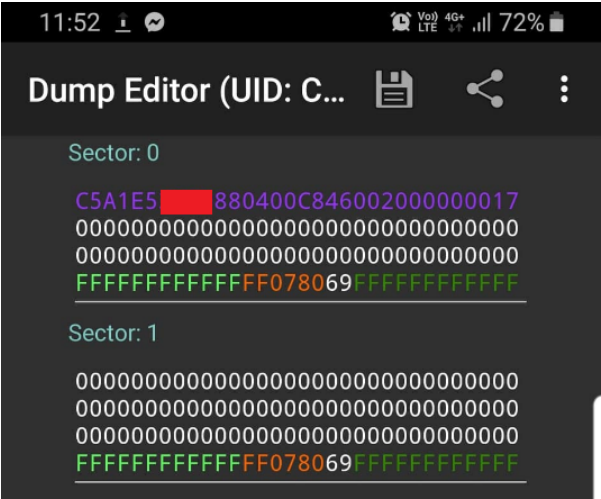


Figure 6: A snippet of the information dump from the Mifare Classic Tool verifying the previously extracted data. The UID is visible as the first eight hexadecimal characters in purple.

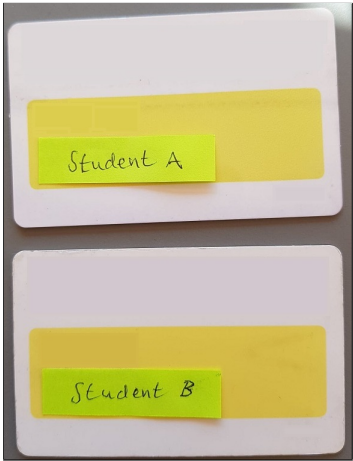


Figure 7: Student access cards A and B.

0	F4	A8	75		88	04	00	C8	07	00	20	00	00	00	20	[0	0	0]
---	----	----	----	--	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---

Figure 8: Student access card A data in sector 0, block 0.

0	C5	A1	E5		88	04	00	C8	46	00	20	00	00	00	17	[0	0	0]
---	----	----	----	--	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---

Figure 9: Student access card B data in sector 0, block 0.


```

Scan a MIFARE Classic PICC to demonstrate Value Block mode.
Using key (for A and B): FF FF FF FF FF FF
BEWARE: Data will be written to the PICC, in sector #1
Card UID: F4 A8 75 
PICC type: MIFARE 1KB
Authenticating using key A...
Current data in sector:
  1      7  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
          6  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          5  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
          4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]

Reading sector trailer...
Writing new sector trailer...
Authenticating again using key B...
Reading block 5
Formatting as Value Block...
Reading block 6
Formatting as Value Block...
Adding 1 to value of block 5
New value of value block 5 = 1
Subtracting 10 from value of block 6
New value of value block 6 = -10
  1      7  00 00 00 00 00 00 19 67 8E 00 00 00 00 00 00 [ 0 1 1 ]
          6  F6 FF FF FF 09 00 00 00 F6 FF FF FF 06 F9 06 F9 [ 1 1 0 ] Value=0xFFFFFFFF6 Adr=0x6
          5  01 00 00 00 FE FF FF FF 01 00 00 00 05 FA 05 FA [ 1 1 0 ] Value=0x1 Adr=0x5
          4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]

```

Figure 10: Changing the data on sector 1 on card A using the MifareClassicValueBlock script.

7.2.2 Cloning using the ACR122U

Another reader that is used during the cloning of the university access control system is the ACR122U. The operating system that is chosen for testing this device is Linux with a Ubuntu distribution. The first step is setting up the communication between the reader and the computer, thus it is necessary to download PC/SC drivers for the ACR122U on the official website at [42]. Additionally, it is necessary to download the pcscd service which coordinates communication with smart card readers that are connected to the system [43]. Once the drivers and the service is installed, a tool called pcsc-tools is downloaded to verify that the reader can communicate with the computer. With the help of the pcsc-scan command from pcsc-tools it is observed that when a card is presented on the reader, information regarding the card is shown in the terminal such as the card type and the UID. To attempt to start writing to a card using the ACR122U, the library Libnfc is downloaded from the GitHub repository found at [44]. After Libnfc and the associated dependencies are installed the commands that can be used to write to a card, unfortunately, experience errors. Therefore other libraries and scripts concerning the ACR122U are located on GitHub and both installed and evaluated. The reading functionality works on the majority of the libraries and scripts but when attempting the write functions different errors are always produced.



Figure 11: The ACR122U reader connects via USB to a computer.

7.2.3 Cloning using the ChameleonTiny

To begin using the ChameleonTiny an application called Chameleon is downloaded on an Android smartphone. With this application, the device is connected to an Android phone via Bluetooth. It offers different functionalities such as UID-clone, cracking, sniffing and, detection, where the device and interface can be seen in Figure 12. The detection function is used to identify the card type that the access control system uses. Once the type of card is found, the UID Clone functionality is used to clone the UID from a card to a specific slot on the ChameleonTiny. Unfortunately, during the experiments, the device stopped working as intended. With further research on the subject, it seemed that technical errors with the magnetoresistive random access memory (MRAM) had occurred [45]. The issue that appeared when trying to clone an access card with the ChameleonTiny is that the UID is incorrectly modified. This hindered the thesis work from testing other functions that are available with the tool.

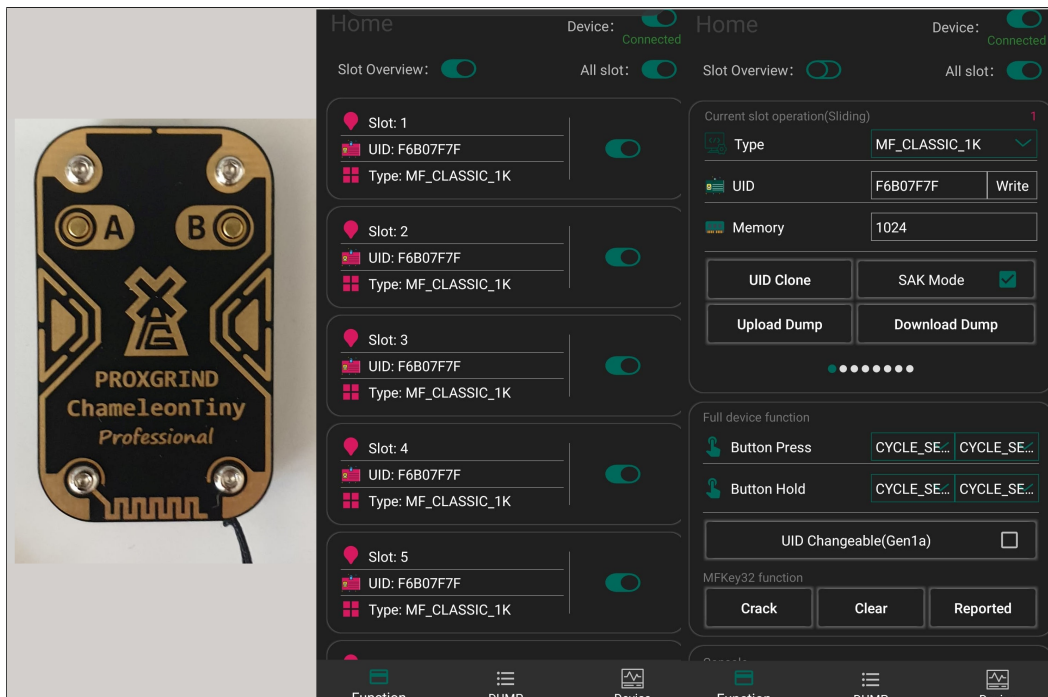


Figure 12: The ChameleonTiny and a snippet of the Chameleon application interface on the smartphone.

7.2.4 Sniffing and simulating an access card

The troubles with ChameleonTiny subsequently lead to evaluating more functions that the ICopy-XS offers to gather and analyze more information. The ICopy-XS can also be used to sniff the wireless communication between a reader and a card. This is accomplished by simply choosing the sniffing option on the device and placing it in close proximity between a reader and a card when the card is read. A selection of the captured information is seen in Figure 14 and in Figure 15 where the UID is exchanged between the reader and a card. When cross-referencing these two captures with other sources such as [46] it is verified that part of these captures represents the anticollision part of the RFID ISO 14443 communication stack where the UID is selected. This function can also be used in the scenario when more than one card is presented to the reader. The only other differentiating information between these three captures other than the UID and BCC is the 2 byte CRC value ("0d 6a" or "bb b6") that follows.

Another useful feature of the ICopy-XS is the simulate function. With this function, one option is to simulate a card after first scanning it. Another option is to manually enter a UID to simulate. Both of these methods are tested in the university access control systems to see if access is granted. The results of this can be seen in 8.1.



Figure 13: The ICopy-XS with its protective plastic component covering the antenna.

[illegible]

Figure 14: A sniff of a successful entry into a university room with student access card B. The last part of the UID and BCC is concealed.

with one button press. This automates the key finding process described above and is used to clone the travel card to be tested against the system. To further test the limitations of the public transportation system a separate experiment is conducted. Namely, using the MFRC522 with the ChangeUID script to only write the identical UID, BCC, and SAK to a magic card. The purpose of this experiment is to compare this system against the results from the two other systems, e.g. if the system validates more information. The results of attempting to use the cards from these two experiments are discussed in 8.2.



Figure 16: A partially censored public travel card.

```

28  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
6   27 PCD_Authenticate() failed: Error in communication.
5   23 PCD_Authenticate() failed: Timeout in communication.
4   19 PCD_Authenticate() failed: Timeout in communication.
3   15 PCD_Authenticate() failed: Timeout in communication.
2   11 PCD_Authenticate() failed: Timeout in communication.
1    7 PCD_Authenticate() failed: Timeout in communication.
0    3 PCD_Authenticate() failed: Timeout in communication.

Card UID: 31 D8 69 [REDACTED]
Card SAK: 08
PICC type: MIFARE 1KB

```

Figure 17: Sectors 0-6 could not be dumped from the travel card according to the errors in communication and authentication.

```
[+] found keys:
[+]  ----|-----|----|-----|----|
[+]  Sec  key A      res  key B      res
[+]  ----|-----|----|-----|----|
[+]  000  a0a1a2a3a4a5  1  -----  0
[+]  001  d3f7d3f7d3f7  1  -----  0
[+]  002  d3f7d3f7d3f7  1  -----  0
[+]  003  d3f7d3f7d3f7  1  -----  0
[+]  004  d3f7d3f7d3f7  1  -----  0
[+]  005  d3f7d3f7d3f7  1  -----  0
[+]  006  d3f7d3f7d3f7  1  -----  0
[+]  007  ffffffffffffff  1  ffffffffffffff  1
[+]  008  ffffffffffffff  1  ffffffffffffff  1
[+]  009  ffffffffffffff  1  ffffffffffffff  1
[+]  010  ffffffffffffff  1  ffffffffffffff  1
[+]  011  ffffffffffffff  1  ffffffffffffff  1
[+]  012  ffffffffffffff  1  ffffffffffffff  1
[+]  013  ffffffffffffff  1  ffffffffffffff  1
[+]  014  ffffffffffffff  1  ffffffffffffff  1
[+]  015  ffffffffffffff  1  ffffffffffffff  1
[+]  ----|-----|----|-----|----|
[+]  ( 0:Failed / 1:Success)

Nikola.D: 0
[usb] pm3 --> hf mf chk *1 ?
```

Figure 18: Here all the "key A" information was found. However "key B" for sectors 0-6 remained hidden with the "res" 0 for these rows meaning that the sectors could not be read.

```
[+] found keys:
[+]  ----|-----|----|-----|----|
[+]  Sec  key A      res  key B      res
[+]  ----|-----|----|-----|----|
[+]  000  a0a1a2a3a4a5  D  6f949a6  N
[+]  001  d3f7d3f7d3f7  D  a38f0da  N
[+]  002  d3f7d3f7d3f7  D  e8ae591  N
[+]  003  d3f7d3f7d3f7  D  41ae1ec  N
[+]  004  d3f7d3f7d3f7  D  b6c6a64  N
[+]  005  d3f7d3f7d3f7  D  5a88081  N
[+]  006  d3f7d3f7d3f7  D  5e05b8a  N
[+]  007  ffffffffffffff  D  ffffffffffffff  D
[+]  008  ffffffffffffff  D  ffffffffffffff  D
[+]  009  ffffffffffffff  D  ffffffffffffff  D
[+]  010  ffffffffffffff  D  ffffffffffffff  D
[+]  011  ffffffffffffff  D  ffffffffffffff  D
[+]  012  ffffffffffffff  D  ffffffffffffff  D
[+]  013  ffffffffffffff  D  ffffffffffffff  D
[+]  014  ffffffffffffff  D  ffffffffffffff  D
[+]  015  ffffffffffffff  D  ffffffffffffff  D
[+]  ----|-----|----|-----|----|
[+]  ( D:Dictionary / S:darkSide / U:User / R:Reused / N:Nested /
tiCnested / A:keyA )

[+] Generating binary key file
[+] Found keys have been dumped to hf-mf-BE971B-key.bin--> 0xff
inserted for unknown keys.
[+] transferring keys to simulator memory (Cmd Error: 04 can occur)
[+] downloading the card content from emulator memory
[+] saved 1024 bytes to binary file hf-mf-31D869-dump-1.bin
[+] saved 64 blocks to text file hf-mf-31D869-dump-1.eml
[+] saved to json file hf-mf-31D869-dump-1.json
[+] autopwn execution time: 21 seconds

Nikola.D: 0
[usb] pm3 --> hf mf autopwn
```

Figure 19: All the keys were eventually found and a dump of the card data was saved into three different files.

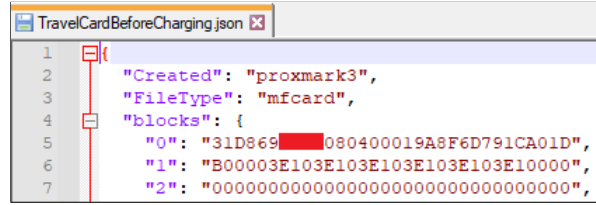


Figure 20: A selection of the data on the travel card is saved in a JSON file.

7.4 Cloning and evaluation of a Mifare 4K access control system

The final system that is evaluated to gain more data is a Mifare 4K access control system. Accessing a room in this system requires PIN authentication and an access card. The data structure is similar to the 1K variant and can be seen in Appendix one. The experiments on this system are executed using some of the same tools and scripts that were used in Section 7.2. The first tests on this system are conducted with ChameleonTiny. This was fortunately tested before the tool stopped working as intended. During the test, ChameleonTiny is first used to detect the card type. Once it was confirmed that the card type is a Mifare 4K access card it is possible to use the UID clone function. As for the RC522 module and the ICOPY-XS Auto Copy function, they require an equivalent type of card to create perfect clones, e.g. 4 kB of data requires 4 kB of memory. Depending on what the system validates as tested with the university system and due to the unavailability of Mifare 4K magic cards, experiments are instead performed with the available 1K cards and the ICOPY-XS simulation and sniffing function. The results of these tests can be seen in 8.3.

7.5 Measurements and RFID protection

There are at least three other important variables when performing a cloning attack. The first variable is the distance between the victim's card and a reader. The second variable is if there are any other interfering materials in between a card and a reader such as textile or other cards. The last variable is the amount of time needed to collect the information. In many organizations using access control systems, there is a requirement of having the access card visible on the employee as a form of identification. In other cases, the cards are usually stored in our wallets or pockets. A form of protective measure found on the market is the RFID blocking products such as sleeves, wallets, or cards.

Using the different available tools it is tested if there are any differences between the cards and readers regarding reading distances, the amount of time it takes for a reader to gather all the data stored on a card and if any materials would interfere with the communication. For distance testing, various tools are used such as a measuring tape and different cards. Four different readers, namely the RC522 module, ACR122U, ICOPY-XS, and a smartphone with the Mifare Classic Tool application are also compared. The measurements are done by controlling at what distance the reader reacts and can gather information from a card. The cards that are compared are the university access card, a magic card, the travel card, and a Mifare 4K card.

The test to measure the amount of time it takes for a particular reader to gather all the information stored in a card is conducted with a timer and the RC522 module, ICOPY-XS, and Mifare Classic Tool. With the RC522 module, the DumpInfo script from the MFRC522 library is used along with the Read Tag function from the ICOPY-XS. The cards that are used during this test are a Mifare 1K card, more specifically a university card, and a Mifare 4K card. The reason why the university card is chosen as the Mifare 1K card instead of the travel card is that both the university card and the Mifare 4K card use the same keys on all the trailer blocks, e.g. the factory default key seen in Section II. Depending on how many different keys are used on an access card the amount of time it takes to gather all the information on the card would differ based on the experiments on the travel card. Therefore it makes more sense to compare the time difference between the readers using two cards with identical keys on all trailer blocks. The DumpInfo script requires manual input of keys in between each execution of the script while the ICOPY-XS uses functions that calculate the keys

automatically.

The experiments that are testing interference from materials between a reader and a card use a wallet, other cards, clothing, and the RC522 module as the reader. This test is conducted in the following fashion: First, a wallet that contains five different cards is placed on the reader, then the wallet that contains one card is placed on the reader and lastly, clothing is placed in between the reader and a card. An RFID protective sleeve is also evaluated as seen in Figure 21. The test is conducted by placing the cards in a protective sleeve and then seeing if the readers react. The readers and cards that are used during this test are the same as in the previous distance test. The results of these measurements can be seen in 8.4.



Figure 21: All of the cards and the RFID protective sleeve that are used during the measurement experiments with some information censored.

8. Results

Here are the results when attempting to use the cloned cards in their respective access control systems. Due to the many different systems, cards, tools, and experiments performed, this section is divided into several subsections for each respective access control system.

8.1 University access control system

The result of the initial cloning with the RC522 module was successful which can be seen in Figures 22 and 23. Our iterative research process showed that not all of the information stored on the access control cards was necessary for the magic card to act as intended and without the access control system noticing any differences. From the previous Figure 10 access was still granted when the data on sectors 1-15 had been modified. The functionality also remained intact when overwriting the data on sector 0, block 1-3. In fact, when all the sectors and blocks except the UID and BCC had been changed as seen in Figure 24, testing this card against the access control system it was still granted access. This was also verified by the ReadAndWrite script experiments which allowed modification of the access bits, the bits that specify if a particular sector can be read or written to, as an example. As seen in Figure 25 the "Key A" and "Key B" have been modified along with the access bits. It is however noteworthy here that the UID "F4..." could not be recognized by either of our scripts after these access bits had been changed even though the card still worked as intended. This test shows that all that stands between entry into this particular access control system is 5 bytes or 8 hexadecimal characters.

It was also noted that the SAK value, which is stored in the sixth byte, had been changed from 88 to 08 in one of the experiments without any change in functionality as seen in Figure 24. This means that this particular access control system does not use SAK values as a prevention mechanism for detecting cloned cards as suggested in the Section II.

The ACR122U experiments were in the end unfruitful since the device could not be used to write to cards. The cloning tests with the ChameleonTiny on the university access control system proved to be successful. The ChameleonTiny was used to clone three different legitimate cards on this system and showed success each time. This was further proven by the PIN authentication, since it was required to use a PIN to gain access to certain rooms, it was necessary to test that when cloning an access card that the correct PIN associated with the card had to be used. This would confirm which of the legitimate access card had been cloned. The cloning function with the ChameleonTiny took less than two seconds and provided access to the system. The simulation using the ICopy-XS also proved to be successful in the university access control system. Both the tests showed to be a success, in other words, both when scanning a card to simulate it and when manually entering the UID, the ICopy-XS could act as a legitimate card.

COM3

Warning: this example overwrites the UID of your UID changeable card, use with care!
Card UID: A3 F6 82 ████
Wrote new UID to card.
New UID and contents:
Card UID: C5 A1 E5 ████
Card SAK: 08
PICC type: MIFARE 1KB

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
14	59	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	57	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	56	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
13	55	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	54	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	53	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	52	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
12	51	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	49	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	48	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
11	47	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	46	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	45	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	44	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]

☒ Autoscrolle ☐ Show timestamp

Ny rad9600 baudRensa output

Figure 22: The previously extracted UID, C5 A1 E5 ..., was successfully overwritten onto the magic card with the help of the MFRC522 library and its ChangeUID script.

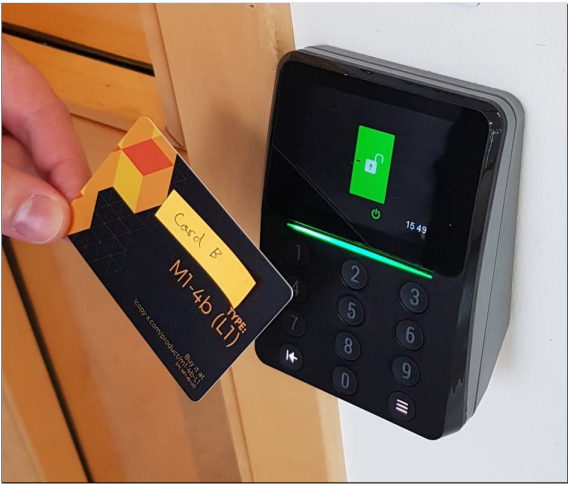


Figure 23: Entry was granted with cloned card when presented to the university reader.

	4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]
0	3	00	00	00	00	00	00	08	77	8F	00	00	00	00	00	00	00	[0 1 1]
	2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]
	1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]
	0	F4	A8	75	████	08	04	00	00	00	00	00	00	00	00	BE	AF	[1 1 0]

Figure 24: The only data that remains the same on the cloned card A is the UID "F4..." and corresponding BCC value.

```

Dump_from_autopwn_broken_card.json
1 {
2   "Created": "proxmark3",
3   "FileType": "mfc card",
4   "blocks": {
5     "0": "E68487F316880400468E45554D704104",
6     "1": "00000000000000000000000000000000",
7     "2": "00000000000000000000000000000000",
8     "3": "0102030405060400468EFF0B0C0D0E0F",
           Key A   Access Bits   Key B

```

```

ReadAndWrite | Arduino 1.8.19
File Edit Sketch Tools Help
ReadAndWrite $
byte sector      = 0;
byte blockAddr   = 3;
byte dataBlock[] = {
  0x01, 0x02, 0x03, 0x04,
  0x05, 0x06, 0x07, 0x08,
  0x09, 0x0a, 0xff, 0x0b,
  0x0c, 0x0d, 0x0e, 0x0f
};

```

Figure 25: The ReadAndWrite script wrote, "Key A" as "010203040506" and "Key B" as "FF0B0C0D0E0F" along with "0400468E" to the access bits in sector 0, block 3. The card is still granted access through the access control system even though the dump script recognizes an unknown UID "E6..." in block 0 as depicted in the dump.

8.2 Public transportation system



Using the proper keys to clone the travel card with travel funds to a magic card with the ICopy-XS Auto Copy function and then attempting to use the magic card was successful. The cloned magic card could be used for traveling. Unlike the university system which only validated the UID and corresponding BCC, the now tampered travel card did not work in the same manner. The card now only had identical UID, BCC, and SAK as seen in Figure 26. When read by one of our readers it registered accordingly, i.e. it was not bricked in the same manner as experienced earlier. However, when attempting to use this card in the public transport system it did not react at all by the reader. The dumps of the travel cards that were taken before charging it with funds, after charging with funds, and after traveling were also compared as described earlier. These dumps were identical to each other, i.e. no data on the card itself had been modified.

0	3	00 00 00 00	00 00 08 77	8F 00 00 00	00 00 00 00	[0 1 1]	
	2	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[1 1 0]	Value=0x0 Adr=0x0
	1	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[1 1 0]	Value=0x0 Adr=0x0
	0	31 D8 69	08 BE EF	DE AD BE EF	DE AD BE B6	[1 1 0]	Value=0x69D831 Adr=0xDE

Figure 26: Sector 0 on the cloned travel card. Note the difference, for example in block 1, compared to the original dump as depicted in Figure 20.

8.3 Mifare 4K access control system

As mentioned earlier we did not have any Mifare 4K magic card accessible to us, it was however possible to clone the information from the first 16 sectors on the legitimate 4K card onto a 1K magic card. This cloned 1K card now contained part of the 4K card, i.e. the one quarter that fit on the 1 kB memory. Testing this card on the access control system granted access, the system did not identify that the card was a clone or that it was a different type of card. The UID-clone function on the ChameleonTiny was also successful on this 4K system. This was tested with three different cards and all the cards showed the same result. PIN authentication was still needed and when using the ChameleonTiny and the associating PIN, access was granted into a room. Attempting to simulate 4K cards with the ICopy-XS was however not successful. Neither by first copying a card onto the ICopy-XS memory and then simulating it nor by manually entering the UID on the device to simulate allowed access.

8.4 Analyzing cloning attempts

To present an overview of the results of the cloning attacks on the different access control systems, this section will attempt to summarize what tools worked during the experiments which can be seen in Table 2. Note that this table does not represent if a reader can be used to create a cloned card on a certain system. When using the RC522 module the only information that was copied to a magic card was the UID and BCC. The RC522 can be used to create perfect cloned cards, i.e. all the information on a legitimate card can be copied to a magic card. The ICopy-XS was used to create perfect cloned cards and used to simulate the UID on cards. The ChameleonTiny simulated the UID, BCC, and SAK.

Access control system	RC522	ACR122U	ICopy-XS	ChameleonTiny
University	YES	N/A	YES	YES
Public transportation	NO	N/A	YES	N/A
Mifare 4K	YES	N/A	YES	YES

Table 2: An overview of the cloning attempts carried out on the three access control systems.
 YES - The tool was used to successfully clone a card that the access control system accepted.
 NO - The tool was not used to successfully clone a card that the access control system accepted.
 N/A - The tool could not be tested due to discussed circumstances.

8.5 Evaluation of range, time and interference when cloning

The results of the distance measurements can be seen in Table 3. This table shows an approximation of the maximum distance when a specific reader reacted to a card that was presented to it. The reading ranges depend on several different factors where the key variable is the minimum amount of received power to activate the chip in the card [47]. Other important variables include the reading antenna characteristics and the propagation environment [27]. The results for the time measurements can be seen in Table 4. This table provides an approximation of the amount of time that was required for a reader to gather all the information for a specific card type. The RC522 module used the DumpInfo script from the MFRC522 library and the ICopy-XS used the Read Tag function. The main explanation of why the time varies is due to the 4K cards having four times as much data to read. The results when using a protective sleeve can be seen in Table 5. This table shows if a specific reader reacted to a card or not when it was completely placed in an RFID protective sleeve. Other materials may also hinder a successful cloning attack. When testing to read a card when placed in a wallet containing several RFID or NFC cards, e.g. a stack of cards, it was shown to be difficult, and results varied. But when a single card is placed in a pocket, or a reader is disguised in a bag or backpack, however, the textile materials do not hinder a successful read.

Card	RC522	ACR122U	ICopy-XS	Mifare Classic Tool
University	4 cm	5 cm	7 cm	3.5 cm
Travel	3 cm	4 cm	5 cm	2 cm
Mifare 4K	2.5 cm	3.5 cm	4 cm	2 cm
Magic	2.5 cm	4 cm	5.5 cm	4 cm

Table 3: Maximum reading distance from a specific card to a specific reader in cm. The left column is showing the card types and at the top of every row, the reader is noted. The Mifare Classic Tool was used on a Samsung Galaxy S8 smartphone.

Card type	RC522	ICopy-XS	Mifare Classic Tool
Mifare 1K	5 s	17 s	4 s
Mifare 4K	20 s	35 s	10 s

Table 4: The amount of time to gather all the information on a specific card type with a specific reader in seconds. The Mifare 1K card used during this experiment was one of the legitimate university cards. The left column is showing the card type and at the top of every row, the reader is noted.

Card	RC522	ACR122U	ICopy-XS	Mifare Classic Tool
University	Protected	Protected	Protected	Protected
Travel	Protected	Protected	Protected	Protected
Mifare 4K	Protected	Protected	Protected	Protected
Magic	Protected	Protected	Protected	Protected

Table 5: RFID protective sleeve evaluation. The left column is showing the specific card type and at the top of every row, the reader is noted.

Protected - The reader did not detect the card when presented.

9. Discussion

The results of our experiments show that cloning attacks could be performed with various tools and without any deeper technical expertise. One access control system proved more difficult to clone cards for and required specialized equipment while the remaining two systems only validated the UID and BCC. In this section the discussion is organized in the order of our RQs, starting with RQ1, and then continuing the discussion with our other tools, experiments, and observations that were formed. Data can be extracted using many different tools while some are better utilized in some situations than others. To read the data on a card to clone, an attacker might utilize social engineering in certain situations which is why factors such as distances, amount of time to read the information and card location are interesting to discuss. The average distance needed to read the data according to our results is approximately 4 cm but when using the ICopy-XS the distance was extended by 1-2 cm. A drawback of this is the price of the device which is more expensive than the RC522 module together with an Arduino and related equipment. The ICopy-XS can however be used completely handheld while the RC522 module is more elaborate to set up and use. Even smaller still is the ChameleonTiny although it, unfortunately, stopped working in our case. Perhaps the most subtle tool for cloning cards is a smartphone, seeing as this would not cause most people to think that there are any malicious intentions with this device as Abellon et al. also discuss in [19]. This however only works for such cards that the smartphone is capable of reading with NFC Tools, Mifare Classic Tool, etc. We noticed throughout our experiments that the cards and all the tools we used in practice at least always managed to extract the UID in a matter of a second. The fact that UIDs are easily extracted is a vulnerability for systems only evaluating them as discussed later in this section.

When measuring the total amount of time to extract all the data on a card, which was necessary for the public transportation system according to our results, a possible explanation of why the differences are noticeable is partially due to the complexity of the code that is executed and partially due to the processing power of the device extracting the data. In other words, the processing power of the Samsung Galaxy S8 is much greater than that of the Arduino. As described earlier the ICopy-XS' Auto Copy performs more advanced functions automatically. Namely, the key recovery process is executed when the device is presented with unknown keys such as those on the travel card. The algorithm for recovering the keys with the ICopy-XS is more complex since it uses many different attacks as depicted in Figure 19. The ideal tool to use and the amount of time taken is therefore also dependent on the keys, e.g. if they are known and pre-programmed into a software dictionary file or not. The ICopy-XS managed to find all six new keys used by the travel card after 21 s. To put this into perspective, Garcia et al. discuss in [23] four different attack approaches for recovering these keys which are described as taking in between a second up to 15 min depending on the attack and circumstances. The keyspace is rather limited since it only consists of 12 hexadecimal characters which are not enough to hinder cloning attacks as we have seen. Yet, we have to consider that without the ICopy-XS the key finding process would have been more difficult with the other tools accessible to us. For example, the Mifare Classic Tool dictionary we found at [35] did not include either of the travel card keys. One method of recovering new and unknown keys is through brute force. The time it would take to perform the brute-force attack and extract all the data would be affected by the processing power of the device used and the efficiency of the software doing the attack as two examples.

When looking at the access control systems themselves and what information they control concerning our RQ2 the results varied. As seen when sniffing the traffic using the ICopy-XS between a reader and a card it confirms that only the UID and BCC were needed for both the university- and the 4K access control systems. When comparing Mifare 4K- and 1K cards the obvious main difference is the amount of storage space on the cards. Since the 4K cards consist of 255 blocks in total, it takes a longer time to read all of the card data as seen from the results. Generally, more memory is more expensive. However, if the system reading the 4K card only validates the UID and BCC, these cards are as vulnerable as their 1K counterparts since the UID and BCC still consist of five or eight bytes, which in turn means that the additional cost is not justifiable. This was also verified since we were able to clone one quarter from the legitimate 4K card onto the 1K

magic card and still gain access, i.e. there is no reason to use 4K cards together with this system from our experiment. Another important note of discussion for these access control systems that guard university computers or company assets is if PIN authentication is required or not. From what we can see, some organizations or parts of their premises do not require any PIN codes. Such functionality requires additional hardware on all the doors that uses it, i.e. more advanced readers with keypads that also control the PIN, this brings additional costs but may very well be worthwhile depending on what the system is guarding.

The required card information for a given access control system is a determining factor of what information to write onto a magic card or simulated by other tools. During the writing experiments to test this, one such experiment with the ChangeUID script resulted in a bricked card. A possible explanation of this based on our research is that we did not specify a valid BCC value for the selected UID. The script is meant to only modify the 4 bytes, which a UID consists of, and then the script automatically calculates a corresponding BCC to write. We rewrote this script to write our values to the whole block without considering the BCC value at the time. Cards may also be bricked when invalid access bits are chosen in the trailer block [48]. During the later experiments which modified both the keys and access bits on the trailer blocks, this test resulted in unexpected behavior. When first attempting to dump the information on the card via the ICopy-XS it resulted in errors, which is expected with new unknown keys as seen in Figure 28. We then confirmed via the ICopy-XS that the trailer block had been changed with our new keys and access bits but noticed that the UID was displayed as an, to us unknown "E6..." value. The card still allowed access to the university access control system with the PIN associated with it. Our interpretation is that the modified access bits confused the reading process but our known UID was still present on block 0 since we only changed block 3. Through simulating the unknown "E6..." UID with the ICopy-XS access was not granted in the system, verifying that this UID was not in use and that the valid UID was intact from the access bit and key rewrite experiment.

As discussed in the results section the comparison of the travel card dumps depicted in Figure 29 showed no difference in the data after charging funds or using the card. Therefore our results show that the funds are not stored on the card. Rather, a possible explanation of this system is that the card information is validated in a back-end system with some centralized database, i.e. a trip is subtracted or registered as active in a database when using the card. It is also possible to purchase travel funds online through a website without any interaction between the card and the physical system. Since it was also observed that the public transportation system required more information compared to the other systems, a confirmation to see what is sent between a reader and a card via sniffing would also be interesting. Confirming this communication through sniffing was however not tested. More research comparing different card types and other public transportation systems would be valuable to identify security flaws etc.

Overall, the cloning process does not require expensive equipment or any deeper technical expertise in certain situations. Cloning with the ICopy-XS is rather quick and automatic in comparison to other tools since it requires no setup and can be done handheld with only one button and two cards. The RC522 module and necessary libraries to clone cards could be optimized better. During our tests, we had to manually switch between different scripts, manually modify these scripts with keys, etc. and re-upload them onto the Arduino. Depending on the goal and its surrounding requirements some optimization and automation can be done here to save time.

Regarding our suggestions on how to harden these access control systems and attempt to hinder or impede cloning attacks relating to our RQ3, some ideas were formed based on our experience and from what the related work had suggested. Since it was possible to change all the data on the cards except the UID and BCC and still gain access to two of the access control systems, one suggestion is to add cryptographic information on the unused memory on the cards as Garcia et al. also mention in [23]. Our suggestion is to combine the UID with some other piece of information, such as the card holder's domain username, and run the combined string through a hash function which results are stored in the card's available memory e.g. in the form of an NDEF text record. From the tests performed on the Mifare 1K cards, only six bytes are required

for the UID, BCC, and SAK in addition to 256 bytes for the keys and access bits, the remaining 762 bytes should be sufficient to store such cryptographic information on the card without tampering with the functionality of the card. This however requires the underlying access control system to read this information and then control it in the back-end and it will also slightly slow down the system depending on factors such as hardware and type of encryption. Other factors that will affect such a suggestion are the card's lifetime and that more resources would be required.

Other system hardening suggestions to impede cloning attacks would be to purchase systems that control the SAK value, as neither the university- nor the 4K access control system did. The SAK can be used as a prevention mechanism, though due to our experience of how effortlessly the reading and writing process is, this additional layer of security would not slow down an attacker enough. Due to our experiments with the travel card and its keys, another strategy would be to diversify the keys used on the access cards as discussed in both [23] and [36]. However, this is only an effective strategy if the system controls more than the UID and BCC, such as with the public transportation system. As experienced earlier, even if the keys are unknown, the UID can still be found. Thus, changing the keys manually or ordering a set of cards with pseudo-randomized keys would not make any difference for systems such as the university one. Our tests with the protective sleeves showed that this might be one of the better solutions for preventing cloning attacks. If the organization requires the employees to have visible ID badges, which also act as access cards, these badges could be combined with such RFID blocking features. This puts a requirement on employees to take the card out of its sleeve each time it is used but it would protect against cloning attacks.

It was also observed that a stack of cards, or at least two different cards using the same frequency, made it difficult to isolate and read only one of them. More thorough testing is however required here which Abellon et al. also mention in [19]. From our measurement experiments the results varied, e.g. running the DumpInfo script and placing two cards on the reader, the terminal either displayed nothing through communication errors or a part of a card's information depending on the placement of the two stacked cards. One simple but perhaps administratively time-consuming and expensive suggestion is to change the cards, UIDs, or other data associated with them every month as an example. This would not hinder cloning attacks but would make them more difficult. Furthermore, it is recommended to perform regular maintenance of the back-end system such as removing old employee access permissions.

Regarding the other tools, the ChameleonTiny stopped working when we tried to read the information stored in one of its slots with the ICOPY-XS. This was tested to see what information except for the UID the ChameleonTiny is simulating after using its UID clone function. The ICOPY-XS did not react to the ChameleonTiny and did not provide any information. Although it is not confirmed that this was the root cause, briefly after this had been tested the device started acting unexpectedly and according to [45] similar behavior had occurred before. The ICOPY-XS also has a simulation function as mentioned earlier. However, neither when scanning a card with this device nor when manually entering a UID to simulate was successful against the 4K system, therefore in our experience, the ChameleonTiny is a better simulation tool.

While Linux was used during the tests with the ACR122U, we also looked at different scripts that could be used with Windows as an alternative. The research did not provide any working scripts or libraries for that particular operating system. The device could be used to read the university access cards which means that the UID could be extracted. As mentioned earlier, two systems only validated the UID and BCC, therefore the ACR122U could be used for this extraction purpose. But all of the other readers could also extract the UID from an access card and therefore this reader was rather impractical in our experience.

Finally, the RFID Diagnostic Card we acquired was not as useful for our work since we had many other tools available to us such as the ICOPY-XS Scan function that discovers most card types. But it can be useful as a reconnaissance tool if the frequency of the target system is unknown and no other tools are available.

Many of these access control systems can be seen as black boxes, i.e. a system is purchased from a vendor who advertises it as being secure, but few have any deeper knowledge of the underlying technology. During our work, we have encountered vendors who for example sell systems with RFID technology as home alarm systems with inferior tags lacking any meaningful encryption or other security mechanisms between the tag and reader or on the tag itself. Naturally, it is advisable to do some research before buying an access control system and ensure it fulfills an organization's risk assessment.

	28	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	[0 0 0]	
6	27	PCD_Authenticate() failed: Error in communication.		
5	23	PCD_Authenticate() failed: Timeout in communication.		
4	19	PCD_Authenticate() failed: Timeout in communication.		
3	15	PCD_Authenticate() failed: Timeout in communication.		
2	11	PCD_Authenticate() failed: Timeout in communication.		
1	7	PCD_Authenticate() failed: Timeout in communication.		
0	3	PCD_Authenticate() failed: Timeout in communication.		

Figure 27: Communication errors while attempting to read the data on certain sectors and their respective blocks.

```
Nikola.D: 0
[usb] pm3 --> hf mf dump
[=] Using `hf-mf-F4A875-    -key.bin`
[=] Reading sector access bits...
.[#] Cmd Error 04
[#] Read block error
.[#] Cmd Error 04
[#] Read block error
.[#] Cmd Error 04
[#] Read block error

[-] could not get access rights for sector 0. Trying with defaults...

.[#] Cmd Error 04
[#] Read block error
```

Figure 28: The Proxmark script attempted reading the information with the same keys it had earlier registered for the UID "F4..." unsuccessfully. Therefore the script also attempted using its default keys unsuccessfully.




	TravelCardBeforeCharging.json	JSON File	2022-04-20 15:08	9 KB
	TravelCardAfterCharging.json	JSON File	2022-04-20 15:07	9 KB
	TravelCardAfterOneTrip.json	JSON File	2022-04-20 15:06	9 KB

Figure 29: The dumped travel card data before charging with travel funds, after charging, and after traveling with the card were compared. No change was seen in the JSON files.

10. Conclusions

The purpose of this thesis was to evaluate three different access control systems that use NFC technology and compare the security vulnerabilities concerning cloning attacks. The aim was to help organizations understand the security flaws in current systems and to suggest countermeasures against cloning attacks. The scope of the research questions explored how data can be extracted from cards, the limitations of what different access control systems validate, e.g. what information the systems control when reading their respective access cards, and possible suggestions for countering the threats. The systems that were evaluated used both Mifare Classic 1K- and 4K type of cards as authentication. To experiment with the technology different equipment such as readers and more specialized penetration testing tools were used.

When comparing two systems used for accessing different premises in a university and an organization similar vulnerabilities were discovered based on our results. The main difference between these two systems was that one used a 1K card and the other a 4K card. This determines the amount of available memory on the cards as being either 1024 B or 4096 B while the organization of the data is similar. The experiments with the equipment used for sniffing the traffic between a card and reader and the different experiments writing data to magic cards proved that both of these systems only validated the UID and corresponding BCC for the respective cards. Naturally, more memory results in a higher cost, and therefore the more expensive 4K cards are not justifiable when no other card data is controlled. Our experiments show that Mifare 1K cards could be used to access the system that used the 4K type of cards, in other words, an attacker only has to clone five bytes of data, the UID and BCC to create a card with the same functionality.

In contrast to the previously mentioned systems, we found that the third system, a public transportation system was more secure. This was due to this particular system validating more information on the travel card. Attempting to clone a travel card required a more advanced and time-consuming approach. This shows that the Mifare Classic line of cards can be relatively secure if the back-end system is capable of handling more information. Related works researching this topic have suggested different hardening processes such as adding cryptography to the access cards as an additional layer of security. The system itself has to be able to manage this additional functionality however and in general knowledge of RFID and NFC technology is limited.

In the end, the biggest security concern lies with the owner or user of a card, e.g. that the owner does not misplace the card or lose it. Cloning attacks involve many variables such as distances, time, and card placement that determine their success. A user can protect their cards by using RFID blocking products. From the systems point of view, the access cards we encountered are generally not very secure as the UIDs are easily extracted through various means, which is especially a security concern if PIN authentication is not used. Therefore, securing the back-end system is crucial to keep attackers at bay whether it is through changing cards every so often and regular maintenance of the system.

11. Future work

As described in the Section II discussing the different vulnerabilities with RFID and NFC there are many possible attack vectors to explore. In fact, when the topic of this thesis was chosen, different related works exploring such possibilities inspired us to learn more about the underlying technology. One of the early reasons for building a simulated environment was due to the possibility of investigating how code can be injected into the back-end systems from the cards. This vector and others were later put on hold.

Regarding the access control systems that this thesis performed various experiments on there are many research possibilities to investigate further. One of them is to gain access to and explore both the middleware and back-end systems. This would require close cooperation with the organizations aiming to secure their systems. One such example regarding the public transportation system would be to sniff the traffic, control what the back-end system validates and how the overall structure is designed.

During the thesis work, other cards and tag types were also encountered such as ICODE and low-frequency EM410x. Here it would be interesting to see how systems using these types behave against cloning attacks and compare them to this work. Additionally, exploring other similar systems such as different public transportation systems and other Mifare systems would be beneficial. Another area using NFC can be seen with contactless payment cards such as VISA and MasterCard but also with services such as Apple- and Google pay. Therefore, the possibility of cloning these payment solutions should be considered.

Furthermore doing more thorough research regarding measuring distances, time and stacked cards with various equipment is necessary to understand the limitations of performing cloning attacks. This knowledge would aid in developing more secure access control systems, cards, and protective solutions.

References

- [1] N. A. Chattha, ‘Nfc — vulnerabilities and defense,’ in *2014 Conference on Information Assurance and Cyber Security (CIACS)*, 2014, pp. 35–38. DOI: 10.1109/CIACS.2014.6861328.
- [2] N. K. Singh, ‘Near-field communication (nfc),’ *Information technology and libraries*, vol. 39, no. 2, 2020.
- [3] S. A. Weis, ‘Rfid (radio frequency identification): Principles and applications,’ *System*, vol. 2, no. 3, pp. 1–23, 2007.
- [4] W. Huang, Y. Zhang and Y. Feng, ‘Acd: An adaptable approach for rfid cloning attack detection,’ *Sensors*, vol. 20, no. 8, p. 2378, 2020.
- [5] H. Geiger, ‘Nfc phones raise opportunities, privacy and security issues,’ *Center for Democracy and Technology*, Apr. 2021. [Online]. Available: <https://cdt.org/insights/nfc-phones-raise-opportunities-privacy-and-security-issues/> (visited on 21/12/2021).
- [6] G. Madlmayr, J. Langer, C. Kantner and J. Scharinger, ‘Nfc devices: Security and privacy,’ in *2008 Third International Conference on Availability, Reliability and Security*, 2008, pp. 642–647. DOI: 10.1109/ARES.2008.105.
- [7] Z. Kfir and A. Wool, ‘Picking virtual pockets using relay attacks on contactless smartcard,’ in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM’05)*, 2005, pp. 47–58. DOI: 10.1109/SECURECOMM.2005.32.
- [8] Y. Sun, S. Kumar, S. He, J. Chen and Z. Shi, ‘You foot the bill! attacking nfc with passive relays,’ *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1197–1210, Jan. 2021. DOI: 10.1109/JIOT.2020.3012580.
- [9] National Institute of Standards and Technology, *Guide to general server security*, Gaithersburg, MD: U.S. Department of Commerce, 2008.
- [10] M. Rieback, B. Crispo and A. Tanenbaum, ‘Is your cat infected with a computer virus?’ In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM’06)*, 2006, 10 pp.–179. DOI: 10.1109/PERCOM.2006.32.
- [11] R. Jain, D. Kumar Chaudhary and S. Kumar, ‘Analysis of vulnerabilities in radio frequency identification (rfid) systems,’ in *2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, 2018, pp. 453–457. DOI: 10.1109/CONFLUENCE.2018.8442623.
- [12] T. Igoe, D. Coleman and B. Jepson, *Beginning NFC: near field communication with Arduino, Android, and Phonegap*. Sebastopol, CA: O’Reilly Media, Inc., 2014.
- [13] H. Kamaludin, H. Mahdin and J. H. Abawajy, ‘Clone tag detection in distributed rfid systems,’ *PloS one*, vol. 13, no. 3, Mar. 2018. [Online]. Available: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0193951> (visited on 17/05/2022).
- [14] X. Ai, H. Chen, K. Lin, Z. Wang and J. Yu, ‘Nowhere to hide: Efficiently identifying probabilistic cloning attacks in large-scale rfid systems,’ *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 714–727, 2021. DOI: 10.1109/TIFS.2020.3023785.
- [15] Q. Zhang and X. Wang, ‘Sql injections through back-end of rfid system,’ in *2009 International Symposium on Computer Network and Multimedia Technology*, 2009, pp. 1–4. DOI: 10.1109/CNMT.2009.5374533.
- [16] L. Sportiello, ‘Internet of smart cards: A pocket attacks scenario,’ *International Journal of Critical Infrastructure Protection*, vol. 26, Sep. 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1874548219300642> (visited on 17/05/2022).
- [17] rfidspecialist, ‘What is the RF chip ID, UID, or NUID? Can I change it or emulate it with a NFC phone?,’ *rfidspecialist.eu*, Mar. 18, 2021. [Online]. Available: <https://rfidspecialist.eu/what-is-the-rf-chip-id-uid-or-nuid-can-i-change-it-or-emulate-it-with-a-nfc-phone--18-03-2021.html>. (visited on 05/04/2022).

- [18] NXP Semiconductors, 'About MIFARE,' *mifare.net*, 2022. [Online]. Available: <https://www.mifare.net/en/about-mifare/>. (visited on 02/02/2022).
- [19] A. P. Abellon, C. J. Ariola, E. Blancaflor, A. K. Danao, D. Medel and M. Z. Santos, 'Risk assessments of unattended smart contactless cards,' in *2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA)*, 2021, pp. 338–341. DOI: 10.1109/ICIEA52957.2021.9436788.
- [20] NXP Semiconductors, 'MIFARE Classic Family,' *mifare.net*, 2022. [Online]. Available: <https://www.mifare.net/en/products/chip-card-ics/mifare-classic/>. (visited on 02/02/2022).
- [21] Lab401, 'PENTESTIPS: DONT BRICK IT - INTRODUCTION TO MAGIC CARDS, UIDS AND BCC'S,' *lab401.com*, July. 16, 2021. [Online]. Available: <https://lab401.com/blogs/academy/pentestips-dont-brick-it-introduction-to-magic-cards-uids-and-bccs>. (visited on 05/04/2022).
- [22] NXP Semiconductors, 'MIFARE Classic EV1 1K - Mainstream contactless smart card IC for fast and easy solution development,' MF1S50YYX_V1 datasheet, Mar. 2014 [Revised May 2018].
- [23] F. D. Garcia, P. van Rossum, R. Verdult and R. W. Schreur, 'Wirelessly pickpocketing a mifare classic card,' in *2009 30th IEEE Symposium on Security and Privacy*, 2009, pp. 3–15. DOI: 10.1109/SP.2009.6.
- [24] Lab401, 'PENTESTIPS - LOADING A CUSTOM SAK WITH THE CHAMELEONTINY,' *lab401.com*, July. 9, 2021. [Online]. Available: <https://lab401.com/blogs/academy/pentestips-loading-a-custom-sak-with-the-chameleontiny>. (visited on 05/04/2022).
- [25] NXP Semiconductors, 'MIFARE type identification procedure,' AN10833 datasheet, May 2009 [Revised Aug. 2021].
- [26] Lab401, 'KNOW YOUR MAGIC CARDS,' *lab401.com*, Apr. 29, 2019. [Online]. Available: <https://lab401.com/blogs/academy/know-your-magic-cards>. (visited on 29/03/2022).
- [27] NXP Semiconductors, 'Standard performance MIFARE and NTAG frontend,' MFRC522 datasheet, Oct. 2009 [Revised Apr. 2016].
- [28] Arduino, 'Arduino® UNO R3,' A000066 datasheet, Jun. 2021 [Revised May 2022].
- [29] Advanced Card Systems Ltd., *Acr122u application programming interface v2.04*, Advanced Card Systems Ltd. Card & Readers Technologies.
- [30] Nikola Lab, 'About iCopy-X,' *icopy-x.com*, 2022. [Online]. Available: <https://icopy-x.com/about-icopy-x-2/>. (visited on 29/03/2022).
- [31] J. Westhues, 'A Test Instrument for HF/LF RFID,' *cq.cx*, Feb. 2009. [Online]. Available: <https://cq.cx/proxmark3.pl>. (visited on 29/03/2022).
- [32] ProxGrind, 'ChameleonTiny Professional (With Bluetooth),' *chameleontiny.com*, 2022. [Online]. Available: <https://chameleontiny.com/product/chameleontiny-professional-with-bluetooth/>. (visited on 05/04/2022).
- [33] Dangerous Things, 'RFID Diagnostic Card,' *dangerousthings.com*, 2022. [Online]. Available: <https://dangerousthings.com/product/rdc>. (visited on 05/04/2022).
- [34] wakdev, 'NFC Tools - Android,' *wakdev.com*, [Online]. Available: <https://www.wakdev.com/en/apps/nfc-tools.html>. (visited on 05/04/2022).
- [35] *MIFARE Classic Tool (MCT)*. (2021). G. Klostermeier. Accessed: Apr. 5, 2022. [Online]. Available: <https://github.com/ikarus23/MifareClassicTool>.
- [36] H. Pereira, R. Carreira, P. Pinto and S. I. Lopes, 'Hacking the rfid-based authentication system of a university campus on a budget,' in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, pp. 1–5. DOI: 10.23919/CISTI49556.2020.9140943.
- [37] Random Nerd Tutorial, 'MFRC522 RFID Reader with Arduino Tutorial,' *randomnerdtutorials.com*, 2016. [Online]. Available: <https://randomnerdtutorials.com/security-access-using-mfrc522-rfid-reader-with-arduino/>. (visited on 05/04/2022).

- [38] Arduino, 'Software | Arduino,' *arduino.cc*, 2022. [Online]. Available: <https://www.arduino.cc/en/software>. (visited on 05/04/2022).
- [39] *MFRC522*. (2021). M. A. Balboa. Accessed: Apr. 5, 2022. [Online]. Available: <https://github.com/miguelbalboa/rfid>.
- [40] Apache Friends, 'XAMPP Installers and Downloads for Apache Friends,' *apachefriends.org*, 2022. [Online]. Available: <https://www.apachefriends.org/index.html>. (visited on 12/04/2022).
- [41] R. Meier, 'Roger Meier's Freeware,' *freeware.the-meiers.org*, 2021. [Online]. Available: <https://freeware.the-meiers.org/>. (visited on 12/04/2022).
- [42] Advanced Card Systems Ltd, 'ACR122U USB NFC READER,' *acs.com*, 2022. [Online]. Available: <https://www.acs.com.hk/en/products/3/acr122u-usb-nfc-reader/>. (visited on 16/04/2022).
- [43] D. Corcoran & L. Rousseau, *Pcsd - pc/sc smart card daemon*, Linux man page.
- [44] *Platform independent Near Field Communication (NFC) library*. (2021). nfc-tools. Accessed: Apr. 19, 2022. [Online]. Available: <https://github.com/nfc-tools/libnfc>.
- [45] Akisame-AI, 'Issues', *github.com*, Aug. 4, 2020. [Online]. Available: <https://github.com/RfidResearchGroup/ChameleonMini/issues/29>. (visited on 02/05/2022).
- [46] G. d. Koning Gans, J.-H. Hoepman and F. D. Garcia, 'A practical attack on the mifare classic,' in *International Conference on Smart Card Research and Advanced Applications*, 2008, pp. 267–282.
- [47] P. V. Nikitin and K. Rao, 'Performance limitations of passive uhf rfid systems,' in *2006 IEEE Antennas and Propagation Society International Symposium*, 2006, pp. 1011–1014.
- [48] T. Cunyat, 'Fixing a bricked Mifare Classic 1K RFID card,' *toni.cunyat.net*, Apr. 10, 2021. [Online]. Available: <https://toni.cunyat.net/2021/04/fixing-bricked-mifare-classic-1k-rfid.html>. (visited on 01/05/2022).

A Appendix one

Mifare Classic 1K 4-byte UID

		Byte Number within a block																
Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Description
15	63	Key A						Access Bits				Key B					Sector Trailer 15	
	62																	Data
	61																	Data
	60																	Data
14	59	Key A						Access Bits				Key B					Sector Trailer 14	
	58																	Data
	57																	Data
	56																	Data
:	:																	
	:																	
	:																	
1	7	Key A						Access Bits				Key B					Sector Trailer 1	
	6																	Data
	5																	Data
	4																	Data
0	3	Key A						Access Bits				Key B					Sector Trailer 0	
	2																	Data
	1																	Data
	0	UID				BCC SAK		Manufacturer Data								Manufacturer Block		

Figure 30: Mifare Classic 1K with 4 byte UID Sectors and Blocks. Each block contains of 16 bytes where each sector contains of 4 block. The last block is called the trailer block and block 0 contains the UID and BCC [22].

Mifare Classic 4K 4-byte UID

		Byte Number within a block																Description
Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
39	255	Key A						Access Bits				Key B						Sector Trailer 39
	254																	Data
	253																	Data
	:																	:
	:																	:
	241																	Data
	240																	Data
	:																	Data
	:																	Data
	:																	Data
32	143	Key A						Access Bits				Key B						Sector Trailer 14
	142																	Data
	141																	Data
	:																	:
	:																	:
	129																	Data
	128																	Data
31	127	Key A						Access Bits				Key B						Sector Trailer 31
	126																	Data
	125																	Data
	124																	Data
	:																	:
	:																	:
	:																	:
0	3	Key A						Access Bits				Key B						Sector Trailer 0
	2																	Data
	1																	Data
	0	UID				BCC SAK		Manufacturer Data									Manufacturer Block	

Figure 31: Mifare Classic 4K with 4 byte UID sectors and blocks. Each block contains of 16 bytes where each sector from 0-31 contains of 4 blocks, but sector 31-39 contains of 16 blocks. The last block is called the trailer block and block 0 contains the UID and BCC.