CHALMERS

Security in access control systems using RFID

Bachelor of Science Thesis in Computer Science and Engineering

David Alm
Hannes Eriksson
Daniel Fallstrand
Robin Karlsson
Viktor Lindström
Robert Stigsson

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, June 2013

Security in access control systems using RFID

David Alm
Hannes Eriksson
Daniel Fallstrand
Robin Karlsson
Viktor Lindström
Robert Stigsson

Examiner: Arne Linde

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000
Department of Computer Science and Engineering
Göteborg, Sweden June 2013

Abstract

The subject of this bachelor thesis is security in access control systems using RFID. The amount of RFID systems is increasing and RFID is being used in more and more areas. Further, more ciphers and security systems are broken which makes it easier for individuals to obtain the materials and the knowledge needed to attack RFID systems. The requirements and scales of the systems has increased and the security has in many cases not been able to keep up with this development.

The work was conducted as a case study where three different systems were examined, for each system several tests were devised to find and exploit weaknesses in ciphers and implementations. A number of commonly used varieties of RFID were tested, including Mifare Classic and EM4100/EM4200. The tests cover several different attack scenarios, for example copying tags, spoofing tags and destroying tags.

Based on the results of the case study, we discuss the security problems identified and propose a number of possible solutions, both regarding the usage of already existing systems and considerations when purchasing and installing a new system. Unfortunately, due to the sensitive nature of this study, some information about the specific cases can not be disclosed.

In general, RFID as it is used in access control systems today is not secure, but improvements can be made and implementations which are considered cryptographically secure do exist.

Sammanfattning

Ämnet för denna kandidatrapport är säkerhet i passersystem med RFID. Mängden RFID-system ökar och RFID används i fler och fler sammanhang. Vidare, fler krypton och system för kopieringsskydd knäcks, vilket gör det lättare för individer att få tag på det material och den kunskap som krävs för att attackera RFID-system. Kraven och storleken på systemen har ökat och i många fall har inte säkerheten kunnat hänga med i utvecklingen.

Arbetet genomfördes som en fallstudie där tre olika system undersöktes. För varje system konstruerades ett antal test för att hitta och utnyttja svagheter i krypton och implementationer. Ett antal välanvända typer av RFID testades, inklsive Mifare Classic och EM4100/EM4200. Testen behandlar ett antal olika attackscenarion, till exempel kopiering av taggar, spoofing och förstörande av taggar.

Baserat på fallstudiens resultat diskuterar vi säkerhetsproblemen som identifierats och föreslår ett antal möjliga lösningar som både berör användandet av befintliga system och faktorer att ta i beaktning vid inköp och installation av ett nytt system. Tyvärr kan inte all information om de individuella fallen tas upp, på grund av säkerhetsriskerna som detta skulle innebära.

Allmänt är RFID som det används i passersystem idag inte säkert, men säkerheten kan ofta förbättras och implementationer som använder krypton som anses vara kryptografiskt säkra finns att tillgå.

Acknowledgements

We would like to give a special thanks to our supervisor Jonas Magazinius and our examiner Arne Linde for their help with this bachelor thesis. We would also like to thank everyone who let us perform tests on their systems, we could not have done this without you.

Glossary

Arduino - A type of microcontroller

FOSS - Free and open-source software

NFC - Near Field Communication, a subset of RFID

Nonce - A cryptographically secure random number that is sometimes used in an encrypted communication

NOP - ”No OPeration” is an operation that does nothing, sometimes used to delay execution of a computer program

RFID - Radio Frequency IDentification

Social Engineering - Methods used (by an attacker) to manipulate people to divulge secret information and/or performing actions

UID - Unique IDentifier, a number that is unique for each tag and can be seen as the tag’s ”name”

Contents

1

Introduction

The foundations of today's RFID technology were laid during World War II, when Alexander Watson-Watt in 1935 presented a new way of using radio waves. Instead of using it as communication from person to person, he used it to locate physical objects. He called this new technology Radar and during World War II it was used by both Americans and Britons, and also the Germans and Japanese. Radar was used to warn about incoming enemy aircraft, but it lacked the capability to tell a friend from foe, which would be very useful. Out of this need the British came up with IFF (identify friend or foe), a system where aircraft would be supplied with a radio transmitter that responded in an indicative way when it received a signal and the sender would then know if it was a friend or foe.

This is the simple principle behind RFID as expressed by Roberti [1], "A signal is sent to a transponder, which wakes up and either reflects back a signal (passive system) or broadcasts a signal (active system)"

Since World War II the development of Radar and other uses for radio waves has continued and in the early 70's, the first American patents for RFID where issued. Since then, new uses for RFID has appeared, among others as anti-theft in stores, road tolls, identification and branding of livestock and pets and also as access control.

The first commercial use for RFID was Electronic Article Surveillance which was developed in the late 60's. Electronic systems for payment in road tolls is another field where RFID would prove to be useful. The first example of such system was, according to Landt [2], installed in Ålesund, Norway in October 1987.

1.1 Aims

The aim of this Bachelor Thesis is to investigate vulnerabilities and faults in access control systems using RFID. Potential vulnerabilities include, but are not limited to, weak or poorly implemented encryption and flawed copy protection schemes.

1.1.1 Goals

To develop one or multiple working attack vectors against at least one of the studied systems, and inform about security flaws in various implementations of RFID and how to protect against them.

1.2 Scope

The project description is ”Security in access control systems using RFID”. The project scope is limited to two of the most popular types for RFID in access control systems, passive 13.56 MHz and 125 kHz. 13.56 MHz is often used in systems involving payment (e.g. public transport), whereas the latter is commonly used as a substitute for keys in properties and buildings.

We would have liked to look into other types as well, such as active ultra high frequency tags (around 900 MHz, rarely used in access control systems), but we did not have the time nor money to invest in acquiring such hardware, and also we felt that this would steal valuable focus from other parts, since we are on a rather tight deadline.

Since we have limited knowledge of the subject and the resources needed to perform advanced side channel attacks, these will not be covered.

1.3 Methodology

The data for this report was collected by first researching RFID in general and later focusing on a select few RFID systems to investigate. The information acquired is presented as a case study consisting of the research and the specifications of the systems, the design of the chosen attack vectors and their execution. The selection of systems studied was based on the location of the systems, which type of RFID was used and if we could get the consent of the maintainers of the systems to examine them. The data for this thesis was collected from January to May 2013.

Some of the data collected and specifics of systems tested has been excluded from this report due to the confidential nature of the information and to avoid identification of the systems tested.

1.4 Related Work

The paper by Garcia et al. [3] describes the security features of Mifare Classic and possible attacks on the cryptographic function Crypto-1. The paper differs from ours by the use of custom hardware and the focus of the paper is solely on security of Mifare Classic, whereas our focus is on more than one system. With generic hardware now widely available, our work was made easier since we did not have to construct our own reader.

The thesis by Loukusa [4] looks at more than one RFID system, like we do, but her

work includes no real-world tests on systems currently in use. The focus of her thesis is on how to perform attacks rather than countermeasures and/or ways of mitigating the damaged caused by the proposed attacks.

Nohl et al. [5] describes one of the first attempts to reverse engineer the Crypto-1 cipher. By using image analysis of circuits and protocol analysis the researchers were able to find a way to retrieve the keys of Mifare Classic tags. The work presented in their conference article served as a foundation for future work concerning the security in Mifare Classic systems and has by extension influenced our work.

Henzl et al. [6] explains why systems that provide secure communication protocols can still be vulnerable to attacks. The authors emphasize the importance of implementation rather than protocols and standards, poor implementations can cause problems even if the protocol used is verified to be highly secure.

The journal article by Kasper et al. [7] describes a real-world attack against a payment system using Mifare Classic. This article shows the efficient exploits that can be used against Mifare Classic in action, and we have used similar methods in our tests. The researchers use custom hardware, which we do not. Reading this article made us aware of the risks involved when adding functionality that gives attackers financial incentives.

2

Technical background

In the 90's, work to standardize RFID was initialized. The need for standardization arose from the vast increase in commercial RFID systems, claims Chawla and Ha [8]. The majority of this work was done by International Standards Organization (ISO) and International Electrotechnical Commission (IEC).

A number of standards exist for RFID describing systems for radio frequency identification of animals, identification for item management, RFID in libraries and identification cards such as proximity cards and vicinity cards. This thesis is mostly concerned with the standard ISO 14443A, which describes the Mifare family of proximity cards. ISO standards 14443B - 14443F describe other types of proximity cards, including Sony's FeliCa cards and CryptoRF from Motorola/Atmel. In addition to this, the different cases studied include systems implementing the standards ISO 18000-1 (identification for item management) and ISO 15693 (vicinity cards). For systems operating on the 125 kHz band, standards exist for item management (ISO 18000) and identification of animals (ISO 11784) but not for access control systems, which is the main focus of this thesis.

Standards regulate how RFID implementations are made and also sets up requirements for the system. The standard can for example regulate frequency, physical requirements, and how the transmission protocol works. A single system can comply with more than one standard (if the standards do not contradict one another).

2.1 Standards

Short descriptions of the standards in question and a number of implementations that occurs in this case study follows below.

2.1.1 ISO 14443

ISO 14443 is common in public transport, payment and access control systems. The tags are well suited for this because of the limited range. The standard is divided in to

the following four parts:

(1) Physical characteristics which specifies things such as size of tags and antennas

(2) Frequency effect and signal interface

(3) Initialization and anticollision

(4) Transmission protocol

2.1.2 ISO/IEC 15693

ISO 15693 specifies cards that has larger reading range than the proximity cards, e.g. ISO 14443. The standard also specifies how the communication should be modulated and is made up of these three parts:

(1) Physical characteristics

(2) Initialization and air interface

(3) Anticollision and transmission protocol

2.1.3 ISO/IEC 18000

ISO 18000 is a much broader standard for RFID technology where the individual parts treats different frequencies. Because of this we chose to focus on the third part that deals with the air interface for 13.56 MHz and the second part which regulates 125 kHz systems.

2.1.4 NFC

Near Field Communication (NFC) is a subset of the standards ISO 14443 and ISO 18092. NFC operate at 13.56 MHz and supports communication between a reader and a passive tag but also peer-to-peer between two readers. NFC is being implemented more and more in cell phones for its possibility to be used in applications such as a electronic wallet as suggested by Want [9].

2.1.5 125 kHz RFID

RFID systems on this frequency band are usually used for simpler systems, such as basic access control, laundry reservations, logistical solutions for automation, time logging in industries, and even as a modern alternative to animal branding.

These tags are very useful due to their simplicity and sole use as an identifier. Compared to tags of the 13.56 MHz frequency band, they are much simpler in basically every aspect. They have more basic construction, the security is sometimes nonexistent as is the data storage, and even when it does exist, it is much more primitive than the 13.56 MHz counterpart.

2.2 Implementations

RFID is a very broad technology and many implementations exists for different tasks. The following implementations are very common in access control systems.

2.2.1 Mifare Classic

Mifare Classic made by NXP Semiconductors Austria GmbH Styria [10] is a very common implementation of 13.56 MHz RFID system using the ISO 14443 standard. The implementation is widespread and used in many common areas such as public transport, access control systems etc. Mifare Classic is available in 1K and 4K, each specifies the amount of space available for storage inside the tag.

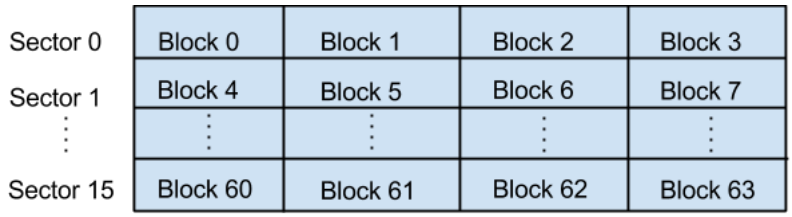

Figure 2.1: Mifare Classic internals

The tags consists of sixteen sectors divided in four blocks each, see figure 2.1. Each sector is secured with two individual keys (A- and B-keys) which can be configured to give access for reading, writing or both. The first of these sectors (sector 0) is read-only and the values in it are set at manufacturing, part of sector 0 is used for the four byte unique identifier, UID.

Security is achieved by the proprietary cipher Crypto-1 which has been analyzed and discussed further by Nohl et al. [5].

Communication protocol

According to Lupták [11], the protocol is initialized when the reader sends a request, and the tag subsequently answers if it is available or not. The reader then sends back information by which it selects a specific tag, and after that the tag responds with its UID and a checksum. The reader then says that it has chosen this UID and adds the checksum so that other tags will not listen by mistake. The tag then says what kind it is, Mifare 1k or Mifare 4k, after which the reader will check and see if it behaves as expected. It is in this phase that the most common attack, described in the next section, can be executed.

Vulnerabilities

Mifare Classic has vast amounts of vulnerabilities as described by Courtois [12], for instance: the pseudorandom number generator is predictable as its value is dependent on

the time that has passed since the chip was powered on. This means that multiple messages can be encrypted with the same nonce. As if this was not enough, there is a known bug in the tags that make them send a known set of bits to the reader when the reader has sent the wrong data and the correct parity bits. This is a vulnerability because this very message is encrypted with the same key that the next part in the transmission is encrypted with, and if an attacker restarts from the beginning of the protocol and makes the tag send the same nonce, he has a part of the whole key.

Once the attacker has the key to one single sector, there are further vulnerabilities in these tags that makes getting the keys to the next sectors a very easy task. Combine all of this with the simple fact that most Mifare Classic tags uses a small set of keys makes it all highly insecure.

If a copy were to be made of a Mifare Classic tag the only thing that would separate the copy from the original is their UID. The UID is set at production and is not changeable. It is possible though to purchase Mifare Classic tags online that have writable UID. With a tag like this it is possible to create a complete clone of the original tag.

2.2.2 Mifare DESFire

Mifare DESFire is produced by NXP Semiconductors Austria GmbH Styria [13] and is an implementation of Mifare that is based on Triple DES or AES according to Henzl et al. [6], just as Mifare Classic, Mifare DESFire uses the ISO 14443 standard and operates at 13.56 MHz. DES is an old cipher developed in the seventies, and was standardized by the American National Bureau of Standards in 1977, and criticized by Diffie and Hellman [14].

The main problem with DES is that it has a very short key (54 bits). This can, however, be improved by using it three times, $ciphertext = E_{K_3}(D_{K_2}(E_{K_1}(\text{plaintext})))$, where K_1,K_2 and K_3 are keys. With a key three times as long, the key length is 168 bits, but because of a meet-in-the-middle attack the security is reduced to 112 bits which is still considered practically safe today.

Mifare DESFire has now been replaced with Mifare DESFire EV1.

Communication protocol

The communications protocol is explained in Henzl et al. [6].

Vulnerabilities

Even though Mifare DESFire uses a more secure cryptographic function it still has vulnerabilities. In this case it is not the cryptography that poses the threat, but the implementation of the protocol that can make the implementation unsafe. An attack is made possible because only data is encrypted and commands, file numbers and offsets is sent unencrypted. This is explained further and discussed in Henzl et al. [6].

2.2.3 Mifare DESFire EV1

This is a newer, more developed version of Mifare DESFire produced by NXP Semiconductors Austria GmbH Styria [15] which exists with different cryptographic functions such as DES/2K Triple DES/3K Triple DES/AES according to Henzl et al. [6]. It is considered cryptographically secure, which does not necessarily mean it is secure against side channel attacks. Mifare DESFire EV1 is based on ISO 14443 and operates at 13.56 MHz.

Vulnerabilities

No papers describing vulnerabilities was found during the time this thesis was written. That does not mean that the system is completely safe. The system is still young and has not been subjected to as much scrutiny as the older systems so there may still be vulnerabilities in this system as well.

2.2.4 EM4100/EM4200

The EM4100/EM4200 protocols were defined by EM Microelectronics Priority 1 Design [16]. The EM4100 protocol specifies tags that have 64 bits of ROM. Out of these 64 bits, 40 bits are set at the manufacturing stage and together they make up the UID. The rest 16 bits are used in the communication. In the communication between an EM4100 tag and reader, the UID will be sent to the reader continuously. There is no anticollision schemes defined for the transmissions, and it is essentially a one way transmission.

EM4200 is a development of EM4100 and is meant to replace it. The tag's ROM has been expanded to 128 bits, and one can choose to use 64, 96 or 128 bits. Furthermore options to change encoding, bitrate and even reading distance has been added. It is worth noting that there exist similarly named protocols, such as EM4005 and EM4102, but we do not discuss these further since we have not encountered them and most of them are being replaced by the EM4200 protocol described in EM Microelectronics [17]. The EM4100/EM4200 tags operate at 125 kHz.

Vulnerabilities

Since these tags uses a simple protocol it is possible to create your own tags by using microcontroller as Loukusa [4] has proved successfully. This means that it is possible to build a tag with a UID that is changeable, for example using an Arduino or a similar microcontroller, and thereby fool the the reader thinking it is a real tag.

3

Attack methods

The method of attacking a system is highly dependent on the system's security as well as what information the attacker is after. The different types of attacks all work differently and a combination of several attacks may be required in order to retrieve the sought information. The attacks are classified together depending on which layer the attack takes place. Physical layer attacks typically refer to attacks obstructing or destroying the RFID tag itself. There are multiple ways to physically disable an RFID tag. Tag jamming, tag interference and tag destruction are a few examples of methods to render a tag useless. Attacks on the network-transport layer are more sophisticated and focus on the communication between the transmitter and the responder. For example this classification includes tag cloning, tag spoofing, eavesdropping and impersonation. Another method an attacker can employ to attack RFID systems is to target the applications used by the RFID to verify the data it receives. An attack of this kind is for example malicious code injection. Such an attack makes use of the weaknesses in the applications used by the RFID system. There are also attacks operating on multiple layers in addition to those already mentioned. These include for example denial-of-service attacks, cryptographic attacks and side channel attacks. Below follows a brief description on a wide variety of attacks with practical examples on RFID systems and recommended defensive countermeasures against the said attacks.

3.1 Sniffing

Sniffing refers to eavesdropping attacks as described by Rieback et al. [18]. Such attacks mainly occur on the communication between two authorized transponders. The information retrieved this way can then be used to reveal security flaws and in turn be useful for generating new attack vectors against the system. Sniffing also includes eavesdropping information from people not willing to share that information.

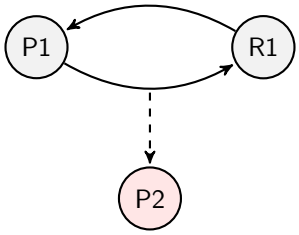

Figure 3.1: Sniffing

3.1.1 Example with RFID

The cards with RFID are intended to be used in such a way that the user can clearly see when the card's data is read. A problem arises when the user is unknowingly eavesdropped on by an unauthorized reader. As seen in figure 3.1, the user P1 is trying to get access from reader R1 while the attacker P2 is surveilling the information that P1 is sending to R1. For example P2 can be a dummy reader placed in the vicinity of a valid reader. The user P1 may then believe the information has been shared in a correct way while in fact critical information may have been shared with a third party.

3.1.2 Countermeasures

There is no possible way to prevent a sniffing attack from getting the data sent. A way to prevent the attacker from receiving any interesting information is to encrypt all the information sent between the tag and the reader.

3.2 Spoofing

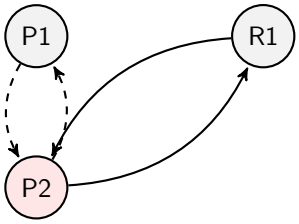

Figure 3.2: Spoofing

A spoofing attack is where the attacker gains access to a system using falsified information. The attacker typically does this by deceiving the security system in such a way that the attacker is believed to be an eligible user. This is explained in detail by Mitrokotsa et al. [19]. This can be done by copying authorization data from an eligible user or in general by manipulating identification data to make it seem as if the data is addressed to/from another user than it truly is.

3.2.1 Example with RFID

RFID systems are vulnerable against spoofing attacks if the reader only requires the information available on the RFID tag. An example of a situation where a system is susceptible to a spoofing attack are in some access control systems. As seen in figure 3.2, if an attacker P2 gets a hold of a tag from an eligible user, such as P1, it is possible to copy and emulate the tag, and thereby get access from reader R1 as user P1. If no additional security is required apart from the identification on the tag then the reader will not be able to tell the emulated tag apart from the eligible one and will thus give access to the emulated one as well.

3.2.2 Countermeasures

It is possible to greatly increase the difficulty of executing a spoofing attack by requiring a manually inputted password in addition to the RFID identification. Another solution is to use advanced protocols with a more substantial authorization of both parties instead of a simple ID number. It is also important to make sure the RFID tag is not cloneable, this usually made by a form of password to read and write to the tag.

3.3 Tracking

According to Rieback et al. [18] a tracking attack is done by gathering information about the victims habits, like geographically determining if the victim is passing through a specific passage or gathering shopping habits of the victim without their knowledge. This information can in turn be used to deduce whether or not that person is of any interest for further attacks.

3.3.1 Example with RFID

An example of this are cards with RFID used for public transport. If it is possible to read information about previous trips from the card then it could also be possible to track that person's movement. Another way to track geographic positions of people is to have a reader which reads from a specific location and then extracts information from the RFID tag about who that person actually is. In such a way there is a potential threat of attackers tracking which people moves about in a specific area.

3.3.2 Countermeasures

The general solution is to make it impossible for an outsider to read the data on the RFID tag. If this is not possible as a consequence of an already implemented system with vulnerabilities then at least no information about previous trips should be stored on the tag itself, but instead in a centralized database.

3.4 Replay attack

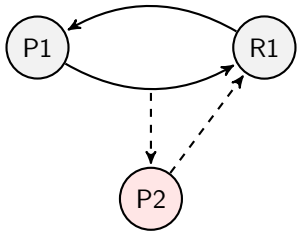

Figure 3.3: Replay attack

A replay attack is a form of attack where the communication between two parties is surveilled, recorded and reused as seen in figure 3.3. The attacker does not need to know anything about the information he replays.

3.4.1 Example with RFID

One example is if you fill a simple credit card with an amount of money and the information exchange between the two parties lacks proper authentication then there is a possibility of a third party recording the communication and later replaying it. If the attacker communicates the correct commands it could be possible to get a deposit of the same amount on the attacker's own card. An other practical example that Rieback et al. [18] poses is England's e-plates, a license plate with RFID tags used in a system to handle congestion charge. In this system you could read somebody else's tag, save it and then play it instead of your own when you drive by a reader.

3.4.2 Countermeasures

In the example with money it is easy to prevent by having the card connected to a server and the card is only used as an ID, and the transactions are monitored closely.

3.5 Denial-of-service

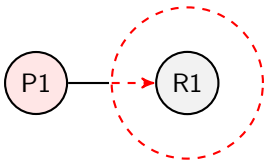

Figure 3.4: Denial-of-service

A denial-of-service attack refers to an attack disrupting regular usage of a system. This is normally done by flooding the system's communications with garbage traffic to

prevent regular traffic from reaching its destination. Furthermore, blocking the medium the communication is broadcasted over to deny any further traffic is an example of a denial-of-service attack.

3.5.1 Example with RFID

A denial-of-service attack can for instance be performed by preventing the tag from being able to communicate with the reader as seen in figure 3.4. This can be done by either shielding the tag by enveloping it in something RFID shielding like the example given by Rieback et al. [18], or by flooding the reader with data to read such that it is impossible for the reader to read the actual tag. Denial-of-service attacks can also be carried out on a higher level, for example by getting the system to blacklist a valid user's card by cloning it and executing an unauthorized operation under that user's name.

3.5.2 Countermeasures

Denial-of-service attacks are generally difficult to prevent but it is not a major problem in the case of RFID. This because of the short communication range together with many spread out subsystems. If either one of the tag or reader is shielded then the shield is usually easily detected and disposed of.

3.6 Man-in-the-Middle attack

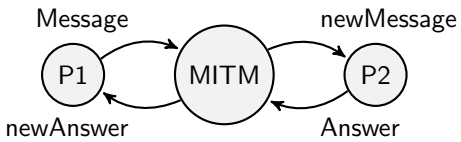

Figure 3.5: Man-In-The-Middle attack

A Man-in-the-Middle attack is performed by intercepting the transmission between two parties as NeoCatena Networks Inc. [20] suggests. As Figure 3.5 indicates, the message sent from the first party (P1) to the second party (P2) is first directed to the person in the middle. The person in the middle then transfers a modified reply to P2 which in turn believes it is communicating with P1. The result of this is both of the parties think they are communicating with each other while in fact the communication is being controlled by the person in the middle.

3.6.1 Example with RFID

One example of this could be could be in combination with a denial-of-service attack. If P1 believes it is communicating with P2 the man in the middle could message P1 and say P1 is not authorized while at the same time interrupting the communication with P2 or just sending bogus messages to P2.

3.6.2 Countermeasures

By decreasing the communication range Man-in-the-Middle attacks will be impractical and will reduce the risk of such an attack accordingly.

3.7 Cryptographic attacks

These attacks mathematically test whether a cryptographic function is weighted or if it leaks information about how the cryptography in the system is implemented.

3.7.1 Example with RFID

If an RFID system uses a cryptographic function with severe weaknesses it might be possible in a reasonable amount of time to decrypt the data available on the tag. One example of this is the cryptographic function CRYPTO1 which has been analyzed by Nohl and several weaknesses has been found. Nohl et al. [5] This is also the most widely used cryptographic function for RFID tags. Newer RFID systems regularly use mathematically proven and open-source standards like Triple DES or AES. One could argue that open cryptographic functions which have been scrutinized by the public for a long time are more safe than secret propriety cryptographic functions. The idea of security through obscurity and Kerckhoffs principle in relation to RFID and smart cards is discussed at length by Courtois [12]

3.7.2 Countermeasures

Since it is difficult to prove that a cryptographic system is secure it is recommended to use an open, well-known and recognized as secure cipher like Triple DES or AES.

3.8 Side channel attacks

A side channel attack is an analysis of how the system affects the physical environment around itself, as described in length by Spadavecchia [21]. One studies the timing, power consumption, acoustics, electromagnetic fields etc for the physical equipment. Depending on the result of the analysis it can be possible to draw conclusions about how the physical parts inside the system are arranged and thus it may be possible to deduce what kind of security is implemented.

3.8.1 Example with RFID

The following types of side channel analyzes as described by Mitrokotsa et al. [19] could be used to further develop attack vectors on security systems using RFID. A timing attack measures variations in the time it takes for the system to perform cryptographic operations. That information may in turn be used to find the system's cryptographic

keys. With a Simple Power Analysis (SPA) you measure the different internal components power consumption and graph a power curve. With known statistical methods you can then compare the power consumption of the system with known cryptographic methods to deduce which kind of cipher the system is using. Differential Power Analysis (DPA) is used to observe the communication between the tag and the reader by analyzing electromagnetic fluctuations in the communication medium.

3.8.2 Countermeasures

One way to defend the system against timing attacks is to have the cryptographic operations run on a constant number of clock cycles, independent on the values the operation is executed on. This can be achieved by filling up the shorter chains of operations with NOP operations. Protection against SPA can be achieved by letting the execution path be independent of secret values. If that is the case then you avoid the power consuming operations in conditional jumps from revealing parts of the cryptographic algorithm. DPA attacks can be made more difficult by increasing the complexity in the hardware and software as well as balancing and randomizing calculations involving the secret key. Lastly it is possible to shield the components from electromagnetic radiation and thus making any further analyzes more economically demanding.

3.9 Code Injection

An attacker can use several methods to inject malicious code into a system to make it behave in ways it was not intended to. This type of attack is commonly used against web applications and one of the most common forms of it is SQL injection. SQL injection uses vulnerabilities in queries to SQL databases to view, edit or delete posts or tables in a database without authorization. The details of SQL injections is discussed at length by Halfond et al. [22].

Other forms of code injection exist but none of them pertain to this case study.

3.9.1 Example with RFID

Consider an RFID system (using tags that can store data) in which the back end system queries a database using information from one or more of the memory sections of the tags. If the back end system contains poorly written SQL queries that does not defend against injections, an attacker could write malicious code to the appropriate memory sections of a tag and use it to for example delete or modify the database in a way that disables the whole system until the database is restored.

3.9.2 Countermeasures

To counter code injection attacks the system administrator can use many of the methods devised by web developers to protect the back end system. One method is to disallow certain special characters (i.e. ";", "\", "=") in the data sections of the tags to make it

harder for an attacker to put code in them. Another solution is to preprocess the query and use built in functions used to test the input to the query before it is executed.

4

Methods and Materials

To be able to perform the attacks available it is not possible to rely on only knowledge, hardware and software is also needed. When choosing hardware and software the focus has been on what an individual can purchase at a reasonable price. It is irrelevant to use hardware that an individual can not or will not buy or that require special competence.

Factors like price, usability and flexibility has been in main focus when both hardware and software combined were considered. The goal is to show owners and system maintainers of RFID systems that with easy to get tools and software, exploits can be used to compromise the system.

4.1 Hardware

In this section hardware that are used in the project are described. Hardware that is mentioned are commercial RFID readers, hobbyist hardware and products that have RFID as a secondary function. Some of the hardware required preparations before they could be used and these preparations are included here.

4.1.1 13.56 MHz

To communicate on this frequency you need a reader that has support for libnfc which is one of the larger software libraries that is used. The software that is mentioned in this report requires a reader compatible with libnfc. Also used in the case study is a special type of a Mifare Classic tag in which sector 0 is writable. This sector contains the UID and a number of other fields that are supposed to be read only. Unauthorized copies of this kind are sold over the internet and can be used to copy a tag in its entirety, which is not possible using standard Mifare Classic tags, see section 2.2.1.

4.1.2 125 kHz

A highly configurable reader with the capability of transmitting information would be ideal in this case. This to make sure the reader had the required necessities to make a spoofing attack possible. The Arduino Uno microprocessor met these requirements. The Arduino platform is an open-source electronics prototyping platform. It has a microcontroller that is easily programmable and has several output and input pins where accessories or your own created electronics equipment can be connected.

An Arduino Uno was used together with an RFID chip named RDM6300 which is a small RFID module capable of reading 125 kHz RFID tags of the type EM4100, see section 2.2.4. This module is independent of the Arduino and has its own antenna. The module takes care of the reading of tags by itself, meaning it could work with different equipment.

Using an Arduino is suitable for this project because it is easy to find an Arduino with RFID accessories. Because of its popularity there is much material to be found online. Software to be used with the Arduino is available for Windows, Linux and Mac. There are also tutorials, forums and well written documentation about standard functions.

Preparations - Reading tags

With the following datasheet Electrodragon [23] it was possible implement a reader with the Arduino. Results of readings would then be sent to the computer, which displays the results. The first version had to be improved because the information sent to the computer was hard to understand and could potentially come from an incorrect reading. The datasheet had information about which format the reader sent the information to the Arduino. With this information the code was improved with the use of checksums. The messages from the reader contained checksums that were used to verify that the received messages were correct. Adjustments were also made in presenting the information to the user by the computer, making it easier to read. The last version of the code showed the read tag in an easy to read format and did not show any information from faulty readings.

Preparations - Spoofing a tag

It was investigated if it was possible to spoof tags with the Arduino, see section 3.2. A guide showed how to spoof an EM4100 tag with a few electronic components and an Arduino. The guide helped us create a fully functional spoofer with very simple code but this code had to be improved to be more usable.

The first version of the code was simple but it was time consuming to spoof a tag, the entire tag message had to be written manually in binary. To make the code more usable functions were added to offload work from the user. A function to write the UID in binary was created. Two additional functions were added that set the parities in the message. With the final code the user was only required to enter a UID into an array and the remaining work would be performed by the Arduino during runtime.

Preparations - Combining reader and spoofer

Lastly the two parts, reader and spoofer was combined to make a spoofer that can read a tag, and then directly after spoof it. Work that earlier focused on coding the reader and spoofer made this merging relatively easy and only a few adjustments were needed to achieve a successful result.

4.1.3 Cell phone with NFC

NFC in cell phones is becoming more and more common, see section 2.1.4. The thought behind NFC in cell phones is to use it to share information such as contacts, photos and more. Another use is as an electronic wallet and avoiding the use of physical currency. The cell phone used has NFC capability and runs the operating system Android. With this phone it is possible to to read several types of RFID tags without buying specialized hardware. It is also possible to write your own Android software or download existing applications from Google Play.

4.2 Software

There is a wide array of software available for communicating with and penetrating tags, especially for Mifare Classic. The software are designed to exploit the flaws of the tag's design and makes it easy for someone with experience with the operating system Linux to use.

4.2.1 mfoc - Mifare Classic offline cracker

mfoc is a FOSS which is used to find keys to Mifare tags. It uses a weakness in the Mifare Classic protocol which results in that if you know at least one key to the tag, information about the next key will be leaked. This means that for this program to find keys it needs to know at least one key beforehand. In most cases this is not a problem since many tags are programmed with standard keys. The program can input keys that will be tested as an addition to the standard keys. If the key would match a key to a sector on the tag then it will be able to find the keys as usual.

4.2.2 mfcuk - Mifare Classic Universal toolKit

This software is used when no keys are available at all, i.e. no keys of standard value. This is used in combination with mfoc, if no standard keys are found then a brute force attack is made by this software to find one single key which in turn is inputted into mfoc.

4.2.3 libnfc

libnfc is an open-source project aimed at making it easy to use nfc readers. The program library contains several tools to read and modify tags.

nfc-mfclassic

This program is solely to read or write data to tags of the Mifare Classic implementation. Retrieved data can later be analyzed with a text editor. To perform these operations you need the keys for the tag to be able to perform the read and write operations. If the keys are not available then for example mfoc could be used to retrieve the keys.

nfc-mfsetuid

This program is used to change the UID on Mifare Classic tags. But for this you need special tags that have a writable UID. The UID on a Mifare Classic tag is four bytes long and is defined with eight hexadecimal signs, for example 0A234F32.

4.2.4 NFC TagInfo

NFC TagInfo is an Android application that uses NFC to read several different types of RFID tags, see section 2.1.4. The support for it is widely spread and the information it presents is very useful. The application then presents the information about which type of tag was read, its UID and the data contained on the tag. This application is developed by NFC Research Lab [24]. They have developed ticket and payment services in collaboration with NXP among other things.

4.2.5 NFC-V reader

NFC-V reader is an Android application that focuses on tags that are based on the standard ISO 15693, see section 2.1.2. With this application it is possible to read and write to tags compliant with this standard. One of the more interesting features is that the data from the tag's memory can be saved as a file. This file could be edited at a later time and written back to the tag again. This application is developed by STMicroelectronics [25] who manufactures Vicinity tags.

5

Case 1: Library

The first case is a study of a library and the uses of RFID in their books to make thefts harder. Instead of scanning the barcode of every single book being checked out, the system can read the RFID tags of all books stacked on the reader at once. An access control system of sorts (analogous to the systems used in stores to detect shoplifting) sounds an alarm if someone tries to leave the library with a book that has not been checked out.

According to the one of the employees at the library, the RFID system performs better than its predecessor, the barcode system.

5.1 Design of attack vectors

The RFID system installed in the library is very simple and it does not use any advanced security measures. The tag data is for the most part unprotected and rewritable. The data contained in the tag is the same for all tags apart from a single sequence used to identify a specific book. This sequence is the same number as the one the barcode specifies. The barcode number is easy to verify since it is printed on the book.

As an extra security measure the system uses one AFI byte (Application Family Identifier, specified in ISO 15693, see section 2.1.2) in the tag which tells the system whether the book is loaned or not. The AFI field is used for other purposes in other types of systems, but in the context of libraries, two different values, 0xC2 and 0x07, for this byte is defined to signify that a book is out on a loan or that it is in stock. This byte is then read by alarm devices located near the entrances and exits of the library. If an attacker passes the alarm devices with a tag in the state not loaned, as told by the AFI byte, the alarm will trigger and proper measures will be taken by the staff.

5.1.1 Denial-of-service - Deleting Books

Since several thousand books are handled by this system the potential to disrupt it by changing the information or by completely deleting the information is high. An attack where information on tags in books were to be changed or deleted could possibly bring down a number of the system's services.

One of the services that could cease to function is the checkout service. Both the automated self-loan desk and the manned loan desk could be sensitive to such a disruption. The return stations could also be disrupted if the tag data is changed before a books gets returned. The security systems that prevent thefts of books might also be affected if the books are no longer recognized by the system.

The effects of this attack could result in a system wide collapse and large amount of manual work would probably be required in order to get the system up and running again, such as re-entering the books into the system and reprogramming the books tags.

5.1.2 Spoofing - Duplicating Books

A book must be checked out before you are allowed to leave with it. It has to be registered and bound to a person, it is also required to be returned before a predetermined time. When leaving the book will be checked by the security system if it is loaned or not by reading the AFI byte.

By loaning a book and copying the information from its tag and writing it to a book that is in stock, it is possible that the book in stock is now seen as the loaned book by the system. It could be possible to leave the premises with both books by doing this. Through spoofing it is possible to steal books without being discovered until someone else wants the specific book or when the next inventory is done.

This is because the book is still seen by the system as being present in the library and not checked out. When it is discovered that it is missing it could be considered lost or stolen. Even if the book is believed to be stolen there is no way to trace it to the thief since the book has not been loaned by the thief, no connections can be made between the thief and book.

5.2 Execution of attack vectors

The tests were performed with help from members of the staff at the library. A cell phone with RFID applications installed was used for these tests. The library provided assistance in the form of test books and a manned loan-service station with which it was possible to see a book's state in the database of the library system. Some tests were also performed on automated self-service loan stations.

5.2.1 Denial-of-service - Deleting books

The data in the tags of the provided books was manipulated in a number of different ways. The altered books were then tested on the loan stations to be able to see how the

system interpreted and handled incorrect data.

5.2.2 Spoofing - Duplicating books

The tests were aimed at detecting which data fields the service station and the alarm system looked at to be able to see if they were able to detect any discrepancies. This information could in turn be used to formulate new tests where the goal is to see whether it is possible to have one book identified and returned as another book through thorough manipulation of the data fields.

5.3 Results

A summary of the results can be found in table 5.1

5.3.1 Denial-of-service - Deleting books

It was proven that by changing information on the tags the system could be tricked into believing that a certain book was another, or the tag information could be completely destroyed resulting in unknown books or multiple ”copies” of the same books in the system. Some values could not be changed such as the UID and the AFI byte.

5.3.2 Spoofing - Duplicating books

It was possible to change the information on the tags making them look like a different book to the system. This book could then be returned as a different book. Worth noting is that complete spoofing was not possible because some values in the tags, such as the UID, could not be changed.

Denial-of-service	✓
Spoofing	✓

Table 5.1: Overview of results

5.4 Discussion

An advantage of using RFID tags in a library system is that they can handle more wear and tear than a barcode. The user friendliness is improved as well, compared to using a barcode scanner where you physically find the barcode for each book and then scan it. With the RFID system the books only needs to be placed on a surface close to the reader.

5.4.1 Problems

It is possible for a person with malicious intent to overwrite data from multiple tags since the tags do not possess any protection from unauthorized writing. This would cause a lot of problems for the administrators and would render the RFID tags in the books useless.

Even if it is not possible to change the AFI byte, a book can be stolen by loaning it, and then a second book's tag is manipulated to replicate the first book as the one just loaned, when the second book is returned it is returned as the first book. This attack is possible because the system allows a book to be returned when it has not been loaned.

Even though an attacker easily can steal books and get away with it through our devised attack, this is not a huge flaw in the implementation of the system. Consider that it can be done simply by removing the RFID tag from the book and then just walk out of the premises with the book hidden. Effectively, the security would be the same as in a traditional library with no RFID.

The use of tags has also increased the complexity of the system, this means there could be more points of failure and design flaws that could be used to cause problems.

5.4.2 Solutions

Many of the problems above arise from the fact that there is no protection against unauthorized writing to the tags, the only thing that can be seen as a security barrier is that some technical knowledge is needed for reading and writing tags. Although this has been made easier, and one could use an RFID reader in a cell phone to interact with the tags.

In contrast to using authorization for writing to tags, it can be made possible using non-writable tags, or make use of a part of the tag that is not writable such as the UID. To allow such a change, a table to convert the UID to the old indexes when barcodes were used is a solution.

Another issue is that the system allows books that have not been loaned to be returned, a good measure to take is to not allow impossible actions such as the above.

6

Case 2: A public authority

The second case concerns the access control system of a public authority where Mifare Classic tags are used to open doors and different users have access to different parts of the buildings in which the system is used. Hence, not all users can open all doors, and their tags control what they have access to.

PIN-numbers, personal codes, are used to enhance security at locations which are more sensitive, for example rooms/departments where a lot of expensive equipment is stored. None of the tests conducted concern the security of RFID combined with PIN, rather the tests focus on RFID exclusively.

6.1 Design of attack vectors

Several tests were considered but discarded as the time frame did not allow for them and because some tests had potential to cause a lot of harm and thereby would have to be performed in some sort of sandbox system, which could not be produced.

6.1.1 Spoofing - Copying tags

Testing whether the system could distinguish between copies of tags and the originals was one of the main focuses of this case. Tests were devised for a number of different scenarios with copied tags, these tests should give information about which of the security features of the tags that are used in the access control system.

The first test was designed to test if the system would deny access to a copy of a legitimate tag, where the copy has the wrong data in sector 0. Testing this only required copying a legitimate tag to one of the standard Mifare Classic tags that does not allow writing of sector 0. If the system examines the content of sector 0, i.e. the UID, and does not recognize it, access should be denied.

The second test was designed to test a scenario analogous to the first test, but in which sector 0 and the UID are copied from the legitimate tag in addition to the rest of

the tag. If this test succeeds and the first one does not, it shows that sector 0 is somehow used in the authentication process.

The third test examines a scenario in which one legitimate tag is copied onto another legitimate tag, e.g. a person with restricted access copies the tag of a person with more access. Should this test succeed while the first test does not, it means that the system only looks up the UID of a tag to see if it belongs to the set of all trusted tags, and does not relate the UID to a specific tag.

6.2 Execution of attack vectors

All tests were conducted with authorization from, and under the supervision of the system maintainers. The testing session was concluded by a lengthy discussion about security in RFID systems and their views on the questions posed in this thesis.

6.2.1 Spoofing - Copying tags

We were given two legitimate tags and two empty tags by the system maintainers, and proceeded to copy the two legitimate tags onto one standard Mifare Classic tag, one Mifare Classic tag with writable sector 0 and one empty tag each.

We were able to test the tags on three different doors, two doors that required a lower level of access and one door that required a higher level of access. One of the legitimate tags had credentials to open all three of the doors, and the other tag only had credentials to open the two low-security doors.

6.3 Results

A summary of the results can be found in table 6.1

6.3.1 Spoofing - Copying tags

The first test, copying a tag with the exception of sector 0, was a complete success. Both of the legitimate tag were copied onto standard Mifare Classic tag with read-only UID's, and these copies could open the same doors as their legitimate counterparts.

The second test, copying a tag including sector 0, was also a complete success, as was expected after the first test. Both copies could open the same doors as their legitimate counterparts.

The third test, copying a legitimate tag onto another legitimate tag, to gain a higher level of access, was also a complete success, as was expected after the first test. Both copies could open the same doors as their legitimate counterparts.

Spoofing test 1	✓
Spoofing test 2	✓
Spoofing test 3	✓

Table 6.1: Overview of results

6.4 Discussion

Security is an important factor when using RFID in access control systems compared to other uses like for example the library system in case 1. Mifare Classic has a number of known vulnerabilities that allow reading and writing of tags which makes copying tags possible, see section 2.2.1.

Below we discuss the problems caused by these security holes and propose a number of ways to mitigate some of them.

6.4.1 Problems

Our tests shows that copying the tags used in this system is rather easy, it only requires off-the-shelf hardware and open-source software that is freely available online, see section 4.1.1. and 4.2.

Copying and manipulating tags

One important thing to note is that the weaknesses and flaws in Mifare Classic not only make copying tags possible, it also allows an attacker to manipulate the data stored in the tags. A consequence of this is for example that a dummy reader can be set up to overwrite tags and render them useless. This can, in some systems, pose a more serious threat than copied tags.

Consider a scenario where an attacker ”destroys” a large number of tags in a short time. This may cause a situation where doors have to be left unlocked or where an attacker can access secure locations through social engineering.

6.4.2 Solutions

Most of our work has revolved around the details of the Mifare Classic cards, and our knowledge of the reader hardware and the back end systems used is limited. Therefore the result of the first test baffled us quite a bit, as we assumed that the readers and/or the back end would be able to recognize a card with an invalid UID. Based on what we know, there is a number of ways in which the security of Mifare Classic systems can be increased, but whether these theoretical ideas can be implemented and used in practice is a question left open.

UID

Better use of the UID as an identifier will make the use of copied tags harder, if the readers in a system check the UID of tag and compare it to a list of trusted UIDs it is easy to detect copies with the wrong UID (i.e. copies written to standard Mifare Classic tags where sector 0 is read-only).

However, this will not defend against copies written onto cards that are already in the system, where UID is trusted, and an attacker could use this to take a tag with limited access and copy a tag with more access onto it. Furthermore, it does not defend against the counterfeit tags where sector 0 can be rewritten.

Instead of checking the UID against a list of trusted UIDs, one could have the level of access linked to the UID in the back end system, so that instead of depending on information from one of the rewritable sectors, it depends on the UID. This is one possible solution to the problem with tags that are already in the system getting elevated privileges.

Since the counterfeit tags with rewritable UID can be made indistinguishable from a legitimate tag, they pose a more serious problem, and the only real way of detecting copies of this kind is some sort of analysis of the logs of the system where you search for tags being used in suspicious ways. For example, a tag used to open two doors in two different buildings in too short of a timespan

This sort of analysis will defend against some sloppily used copies, but in an environment where a single breach of security can be devastating, this kind of copies pose a serious threat that is very hard to defend against.

7

Case 3: Common access control systems

The third case concerns RFID in common access control systems. RFID is frequently used for access control in places such as businesses, apartment complexes, public institutions etc. These systems are often based on 125 kHz RFID technology, and many of them use the protocol EM4100, see section 2.2.4. They usually need to handle a large amount of individuals passing through, disabling lost tags and entering new into the system.

7.1 Design of attack vectors

A valid tag is required in order to gain access to the premises. The tag is held up to the reader, the UID is checked against a back end and gives access if the tag is recognized. The easiest way for us to affect the system is to focus on the tags and readers and exclude a possible back end.

7.1.1 Denial-of-service - Disrupting readings

It is critical that the reader can perform readings for the system to be operational. Access to the premises is thereby dependent on the readers and their ability to read tags.

One possible way to block the reader from performing a successful reading is to corrupt the message sent from the tags. EM4100 lacks methods to handle or avoid collisions and each tag sends a message containing its UID over and over again. As a consequence of this the tags are vulnerable to interference from other tags in the vicinity. Two or more messages at the same time may then corrupt each other and render the information received by the reader useless.

A very discrete way to disrupt the reader would be to place a device near the reader

that sends out messages continuously. This device could for example be a small tag that is only a chip or a small microcontroller that could send out different messages that collide with the legitimate messages.

Disrupting readings should render the system useless since its primary function is obstructed. However, this type of attack is directed at only one reader, so causing severe damage would require a number of attacks on different readers.

7.1.2 Spoofing - Impersonating a tag

A UID that is considered valid by the system is required to gain access. Such a UID could be obtained by reading a valid tag, it would then be possible to spoof this tag by transmitting the UID. This could be done secretly since the reading only requires you to be near the tag that you want to spoof.

The spoofing would then be done with a device capable of replicating the tag message. Using this device with a valid UID against a reader should be accepted as if it was a real tag. Since the reader would only receive the tag message it has no way of knowing if it is the real tag sending the message or if it is being sent from a spoofing device.

The result of using a spoofer with a valid UID should give access just as if it was a regular tag.

7.2 Execution of attack vectors

The attacks were executed on an access control system where we had permission to perform the attacks, we also had a number of valid tags at our disposal when performing the attacks. The Arduino discussed in the section Methods and Materials (see section 4.1.2) was used to perform these tests.

7.2.1 Denial-of-service - Disrupting readings

The Arduino was used to spoof an unauthorized UID and thereby block a legitimate reading.

A regular EM4100 tag that was unauthorized for the particular system tested was also used to block a legitimate reading.

7.2.2 Spoofing - Impersonating a tag

The execution of this attack was straightforward, a tag was read and then spoofed with the Arduino. The attack was tested on two separate readers and several valid tags were spoofed.

7.3 Results

A summary of the results can be found in table 7.1

7.3.1 Denial-of-service - Disrupting readings

This worked when using a passive tag that was not recognized by the system or with a spoofing device using an invalid UID. Using a passive tag gave noticeably better results than using the spoofer.

7.3.2 Spoofing - Impersonating a tag

Every attempt to spoof valid tags were successful. Spoofed tags could be used on both readers that were available. Neither of the readers denied access to the spoofing device on any of the attempts. Several tests were performed during a period of one week and it never failed.

Denial-of-service	✓
Spoofing	✓

Table 7.1: Overview of results

7.4 Discussion

A lost tag can more easily have its permissions revoked than a physical key. Depending on how the access control system is implemented, a tag can easily be blacklisted and denied access. A lost key require locks and keys to be changed and this is often very costly. Using RFID also makes it possible to track a specific tag. For example tracking can be used to supervise employees' movements at a company and make sure they are paid according to time spent at work.

7.4.1 Problems

The tests performed have proven the simplicity of these systems and how spoofing and denial-of-service can be performed fairly easy.

UID spoofing and generating UIDs

To access areas or rooms where regular keys are used, you need to actually possess a key in order to make use of it or copy it. Compare this to a tag which is possible to read from a distance and no physical contact is needed in order to make a copy. This case study has shown that with some technical knowledge it is possible build a spoofer which is able to copy and spoof tags from a distance with relative ease.

Another way to acquire an authorized tag is to use a hardware spoofer and simulate all possible tags. This kind of brute force strategy would be very time consuming to execute. Consider for example an EM4100 tag that has a UID of 40 bits. This means there are 2^{40} combinations to test. Looking at Priority 1 Design [16] and their description

of the bit length in terms of clock cycles when transmitting, the length of a bit can be 64, 32 or 16 cycles. This means that a shorter bit length should result in a faster transfer of the tag message.

The clock frequency is 125 kHz, that gives a period of 8 μs. If the fastest case would be considered, a bit length of 16 cycles, then the one bit would take 128 μs to transmit. Sending the 64 bit long tag message would take 8192 μs, or 8.192 ms. Going through all the combinations would take approximately

$$2^{40} * 8.192\,\text{ms} \approx 9.007 * 10^{9}\,\text{s} \approx 285.6\,\text{years} \tag{7.1}$$

This approximation assumes that the verification of the UID is fast enough not to cause delays. It could potentially take even longer time if the reader is slow or has a delay added intentionally. The UID could also be larger, increasing the time even more. In order to even consider using this method you would need to make assumptions about the UID that narrows down the range of UIDs to test. If UIDs are chosen in a series, then one known UID can be used to make assumptions about the range of the other UIDs.

Denying access

One type of denial-of-service attack is a spoofing attack (using an invalid UID) over a prolonged period of time, the spoofing attack keeps the reader occupied with work and prevents authorized tags to be read correctly. This kind of attack could be performed using a microcontroller and an antenna or an unauthorized tag. Using an unauthorized tag instead of a microcontroller is a better approach because the timing gets better, a passive tag starts its transmission after getting power from the reader whereas a microcontroller just sends its message at any time. Hence, using a tag to block the reading of an authorized tag is ideal, since they will both start transmitting at the same time. Also, a tag is usually smaller and easier to hide, so the attack might be harder to detect.

7.4.2 Solutions

Systems that use tags without any security features should only be used in common areas where ease of access is more important than to have a secure area. To administer access to a more secure area a PIN-code should be used together with the tag. If even more security is needed then a tag with cryptographic security inside the tag should be considered since the former type of tags have no form of security to prevent reading from or writing to the tag.

To prevent denial-of-service collision detection can be used to separate tags when reading. It might be expensive to add this to already installed systems and will also not work if the attack is done by a device which does not react to this kind of security feature. A solution that could detect both tags and disrupting devices would be if the readers and back end are able to detect and warn if a tag reappear constantly when readings are performed, detect when invalid readings are occurring frequently or when other signals are detected by the reader that could disrupt readings. With this solution

a reader can be checked and tags or disrupting devices can be removed from the reader's proximity.

8

Summary of Results

The results of all our tests were successful (see table 8.1 below), though in some cases some alterations had to be made to the original test specifications.

In Case 1, two tests were performed, a spoofing test where the tag in one book was copied to the tag of another book and a denial-of-service test where all information in a tag was overwritten. Both tests succeeded, the first test rendered two different books identified as the same book by the system and the second test rendered the target book unrecognizable to the system.

In Case 2, several similar spoofing tests were performed, they differ only in how UID is handled in copies of Mifare Classic cards (for details see section 6.1.1). All three tests succeeded, the three copied tags could open the same doors as their legitimate counterparts.

In Case 3, two tests were performed, one spoofing and one denial-of-service test. The spoofing test was a success, with an Arduino microcontroller we were able to read and spoof a tag in such a way that it was accepted by the system. The denial-of-service test, where a second tag was used to corrupt the data received by a reader, was also a success.

Case 1 - Denial-of-Service	✓
Case 1 - Spoofing	✓
Case 2 - Spoofing test 1	✓
Case 2 - Spoofing test 2	✓
Case 2 - Spoofing test 3	✓
Case 3 - Denial-of-Service	✓
Case 3 - Spoofing	✓

Table 8.1: Overview of results

9

Discussion

One thing worth noting is that since we have not studied all available types of RFID systems out there, problems and concerns discussed below might not be relevant to other systems, and these alternatives could void concerns that we have about RFID in general, based on our study. For example moving from Mifare Classic to Mifare DESFire might make it impossible for an attacker to read and write the data in tags within a reasonable time frame.

9.1 Our results

The results of our case study clearly show that a number of popular and frequently used implementations of RFID are vulnerable to attacks that do not require expensive hardware and/or advanced technical knowledge. While the results in the individual cases perhaps could have been foreseen, the security problems concerning RFID in general are alarming.

None of the security flaws exploited in this case study are specific to the systems tested, but rather problems inherent in the different implementations of RFID. It was not a stated goal of this thesis to find weaknesses that affect for example all Mifare Classic or EM4100 systems, but such weaknesses exists for all cases and they are the most interesting ones.

It is possible and perhaps even plausible that further vulnerabilities exist, including weaknesses in the specific systems rather than the implementation chosen, in the systems tested.

9.2 Reasonable level of security

The level of security needed in access control systems is not the same, different situations call for different security measures. Systems with more advanced security features are

often more expensive. When looking at and scrutinizing security you can easily get a little paranoid but when planning the introduction of a new system it is important to look at what is a reasonable level of security. To judge this, one must look at what the system will be used for, including possible future extensions.

For example, consider a landlord that decides to use RFID tags instead of a door code to access the stairwell of an apartment building and chooses a system where copying tags is possible, but requires some knowledge, hardware and a reading of a legitimate tag. The effort needed to copy a tag is quite significant, and the threat of someone getting in to the stairwell is not very serious, since the apartments have regular locks on their doors. In this case, spending a lot more money on a system which makes copying tags harder might not be the best idea, but if we consider a similar case in which the tags also unlock the apartment doors, the level of security in the system will have to be increased.

The validity of this type of analysis does of course depend on the knowledge of different systems and their weaknesses. Like in any other situation, to be able to make good decisions you need to do good research first. Missing information from vendors about security systems could potentially lead to the implementation of expensive systems which do not provide adequate protection. The use of RFID is growing rapidly, and it is crucial that the people tasked with choosing what system to go with are well-educated when it comes to security. One alternative to reading up on RFID yourself is to bring in a third party to help out with the choices and provide unbiased information about the potential weaknesses of the systems being considered.

9.3 When should RFID be used?

In some systems the use of RFID instead of keys or magnetic stripe cards might not be a good idea. The fact that RFID tags can have their data read or overwritten (granted that the cryptographic functions and other security measures of the particular RFID system can be breached) from a distance and without physical access to the tags is certainly a problem. While a magnetic stripe card or an ordinary key can be copied by an attacker, this requires physical access to the legitimate key/card, this is not the case with RFID tags.

Consider for example a system using Mifare Classic, which is a type of system for which attacks are rather easy to perform and the vulnerabilities are well-known. Just by holding up a reader to someone's pocket an attacker can copy a tag without even making the owner suspicious. In contrast, to copy a magnetic stripe card or key, the process would have to involve stealing the aforementioned and copying it, a process which would take time and the owner is likely to miss her key/card. Unwanted access to RFID tags can of course be minimized using a wallet or bag lined with aluminium that shields the tag, but this does not solve the problem completely.

This inherent security problem poses a number of questions, is there applications where the convenience of RFID does not justify the risks? Should RFID be used for

access control systems or should keys and/or magnetic stripe cards be used instead?

There are however applications where RFID can be seen as an improvement, this is often due to RFID not is used as a security measure, as with the studied library case. In that study RFID was not solely used for security but above all to assist and make it easy for individuals to loan their books.

9.4 Challenges

While working on this thesis we have encountered a number of challenges. Some of the systems have information stored in their tags in an encrypted fashion, the information does not always make any sense and it is not obvious how it can be usable. To get around this problem one can perform systematic tests to see how the system reacts if anything changes, this kind of black box testing is however quite difficult and tedious. Furthermore, this can continue to be a problem even though the information is understood since it might be connected to a back end with additional layers of security. In that case another attack method should be considered.

Another challenge is that even if the security is breached on one tag, it can be problematic to reproduce on other tags. To use a solution where many people scan their tags is one possibility, alternatively, a hidden reader which reads tags without the owner noticing could be used. To be able to read tags unnoticed would probably require longer reading range in which case a new antenna would be needed. This is a big challenge since many of the tags use a technique where information is transferred by magnetic induction, and this means that a reading range beyond a few meters is unthinkable.

When examining the systems in our case study the focus has been on front end rather than back end. This is in part due to the fact that we want to produce results that are general and not caused by poor choices in the design of a particular back end system. Another reason is that we have not had the consent of a system maintainer to examine their back end. Yet another reason is the complexity of an analysis of this kind, due to the time limits we would not have been able to properly examine and look for weaknesses in a back end system.

9.5 Doing more damage: code injection

One type of more advanced attack that should be subject to future research is code injection. Seeing that the most commonly used systems have weaknesses that allow an attacker to write arbitrary data to a tag, malicious code could be injected into the system back end (for a more detailed example see section 3.9.1). Using techniques developed to attack for example web applications, an attacker could for example manipulate back end databases. SQL injection is one way to do this and it could possibly be used to cause a lot of damage with only one reading of a manipulated tag.

We did not get an opportunity to test SQL injection but it is still a very interesting idea. A number of different methods to inject malicious code into systems exist and seeing if RFID systems are vulnerable would have been very interesting.

This approach has the potential to do a lot more harm than just manipulating tags since attacks are directed at the underlying structure. Destroying a back end database could potentially bring down the whole system until a backup can replace it. The problems do not end there however, because the system will still be vulnerable to the same kind of attack until the back end code has been changed to address the security holes.

9.6 How to maximize RFID security

To make it as hard as possible for an attacker to exploit an RFID system, we propose some measures making it harder for the attacker.

9.6.1 Use secure cryptographic functions

As discussed earlier, implementations that use known secure ciphers, such as AES and Triple DES, and do not rely on security through obscurity is often best. Most of these ciphers have been thoroughly evaluated by leading researchers and organizations before being introduced and has since then been used in many different applications.

9.6.2 Know the implementation

If a system is used with known security flaws, it is very important to know what this means, and proper measures should be taken to prevent from known attacks. It is important to fully understand the repercussions of for example the flaws in Mifare Classic if you plan on using it.

As discussed above, consulting a third party to help evaluate different options can be very helpful when choosing which implementation to go with.

9.6.3 Be careful when adding functionality

When considering adding functionality to an RFID system, always be sure to do a proper evaluation of the security issues. Introducing for example payment systems can make the system much more attractive to attackers, and what was previously deemed safe may be considered unsafe when the threat changes.

9.6.4 What to store inside a tag

Treat data inside a tag as if the cryptographic function is broken. If the cryptographic function is broken at a later time an attacker will not have the same possibilities and the system is not as vulnerable. Keep as little information as possible inside the tag so that if the cipher is broken, the information that an attacker can read from the tags is harmless.

9.6.5 Use keys properly

Keys poses two big issues, choosing keys and distributing them. Lack of care in choosing good keys has been common in the cases that we have studied. First of all, do not use any default keys at all, this makes it very easy for an attacker to guess the keys. Even though just one key is set to a default value, it may be possible, as with Mifare Classic, to use this to further exploit other security issues in the implementation and find out the rest of the keys.

If it is possible, one should use different keys for as many individual tags as possible. The difference this makes is that if an attacker gets a hold of keys for a single tag, which for example could take a few hours, it does not holds keys for all tags in the system. It is then impossible for the attacker to read and manipulate any other tag. In practice this means that an attacker can not read a tag without stealing it. This approach would defend against the types of attacks that we tested in case 2.

9.6.6 PIN codes

Using RFID together with PIN codes make the system a lot safer by making it impossible to use a copied tag without the PIN of the owner. This can also be used selectively for parts of the system, perhaps not all access control points require this extra layer of security.

9.6.7 Using Unique ID properly

As previously discussed with regard to case 2, how the UID of tags is used can affect the security of a system. Making sure that the UID is checked against a back end every time a tag is read is one way to make it harder for an attacker to copy tags.

Important to remember though is that you should not be overconfident and assume that the UID field of a tag really is read-only. We have seen in two different cases, Mifare Classic and EM4100/EM4200, that this is not true. An attacker targeting Mifare Classic can easily buy counterfeit tags online, and in the EM4100/EM4200 case you can simply program a microprocessor to spoof any UID. The problems with illegitimate tags will surely not end with Mifare Classic, we are quite confident that in the future we will see counterfeit tags for even more systems.

10

Conclusion

The way RFID is used in access control systems today is not secure, methods exist to manipulate tag data for most commonly used implementations, including Mifare Classic and EM4100/EM4200. Tools that exploit security weaknesses in these kinds of systems are easy to obtain and to use, most attacks only require a cheap RFID reader/writer and free software.

Some of the problems with security in RFID systems can be solved by developing better routines for using the systems, but a subset of these problems are direct consequences of poor implementations and weak ciphers. Use of implementations with inherent weaknesses that have been known for years should be discouraged and maintainers of such systems need to be very aware of the ramifications of the security holes. Retailers and manufacturers need to take their part of the responsibility and minimize the number of new installations of broken systems, while customers need to make sure they have the knowledge to communicate their needs and analyze the information from vendors.

More research on RFID security is necessary to make sure that implementations that are considered safe today are not vulnerable to for example side channel attacks. Further, examining code injection attacks on RFID systems could reveal more severe attack vectors against implementations with weaknesses that allow writing arbitrary data to tags.

References

[1] M. Roberti, "The history of RFID technology," *RFID Journal*, 2005. [Online]. Available: http://www.rfidjournal.com/articles/view?1338

[2] J. Landt, "The history of RFID," *Potentials, IEEE*, vol. 24, no. 4, pp. 8–11, 2005.

[3] F. Garcia, G. Koning Gans, R. Muijrers, P. Rossum, R. Verdult, R. Schreur, and B. Jacobs, "Dismantling mifare classic," in *Computer Security - ESORICS 2008*, ser. Lecture Notes in Computer Science, S. Jajodia and J. Lopez, Eds. Springer Berlin Heidelberg, 2008, vol. 5283, pp. 97–114. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88313-5_7

[4] T. Loukusa, "Analys av säkerheten av RFID i inpasseringssystem," 2012.

[5] K. Nohl, D. Evans, Starbug, and H. Plötz, "Reverse-engineering a cryptographic RFID tag," in *Proceedings of the 17th conference on Security symposium*, ser. SS'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 185–193. [Online]. Available: http://dl.acm.org/citation.cfm?id=1496711.1496724

[6] M. Henzl, P. Hanacek, P. Jurnecka, and M. Kacic, "A Concept of Automated Vulnerability Search in Contactless Communication Applications," in *Security Technology (ICCST), 2012 IEEE International Carnahan Conference on*, 2012, pp. 180–186.

[7] T. Kasper, M. Silbermann, and C. Paar, "All you can eat or breaking a real-world contactless payment system," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, R. Sion, Ed. Springer Berlin Heidelberg, 2010, vol. 6052, pp. 343–350. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-14577-3_28

[8] V. Chawla and D.-S. Ha, "An overview of passive RFID," *Communications Magazine, IEEE*, vol. 45, no. 9, pp. 11–17, 2007.

[9] R. Want, "Near field communication," *Pervasive Computing, IEEE*, vol. 10, no. 3, pp. 4–7, 2011.

[10] NXP Semiconductors Austria GmbH Styria, "MIFARE Classic," 2002–2013. [Online]. Available: http://www.mifare.net/products/mifare-smartcard-ic-s/mifare-1k/

[11] P. Lupták, "Mifare Classic Analysis," Unknown. [Online]. Available: http://www.nethemba.com/mifare-classic-slides.pdf

[12] N. T. Courtois, "The Dark Side of Security by Obscurity and Cloning MiFare Classic Rail and Building Passes Anywhere, Anytime," Cryptology ePrint Archive, Report 2009/137, 2009. [Online]. Available: http://eprint.iacr.org

[13] NXP Semiconductors Austria GmbH Styria, "MIFARE DESFire," 2002–2013. [Online]. Available: http://www.mifare.net/products/mifare-smartcard-ic-s/mifare-desfire-d40/

[14] W. Diffie and M. Hellman, "Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard," *Computer*, vol. 10, no. 6, pp. 74–84, 1977.

[15] NXP Semiconductors Austria GmbH Styria, "MIFARE DESFire EV1," 2002–2013. [Online]. Available: http://www.mifare.net/products/mifare-smartcard-ic-s/mifare-desfire-ev1/

[16] Priority 1 Design, "EM4100 Protocol description," 2007. [Online]. Available: http://www.priority1design.com.au/em4100_protocol.html

[17] EM Microelectronics, "EM4200," 2012. [Online]. Available: http://www.emmicroelectronic.com/products.asp?IdProduct=282

[18] M. Rieback, B. Crispo, and A. Tanenbaum, "The evolution of RFID security," *Pervasive Computing, IEEE*, vol. 5, pp. 62–69, Jan–March 2006, issue: 1.

[19] A. Mitrokotsa, M. R. Rieback, and A. S. Tanenbaum, "Classifying RFID attacks and defenses," *Information Systems Frontiers*, vol. 12, no. 5, pp. 491–505, 2009.

[20] NeoCatena Networks Inc., "RFID Security Risks," 2004–2012. [Online]. Available: http://www.neocatena.com/technology/risks/

[21] L. Spadavecchia, "A Network-based Asynchronous Architecture for Cryptographic Devices," 2005.

[22] W. Halfond, J. Viegas, and A. Orso, "A classification of SQL-injection attacks and countermeasures," in *Proceedings of the IEEE International Symposium on Secure Software Engineering, Arlington, VA, USA*, 2006, pp. 13–15.

[23] Electrodragon, "RDM630 Specification," Unknown. [Online]. Available: http://www.electrodragon.com/wp-content/uploads/2012/04/RDM630-Spec.pdf

[24] NFC Research Lab, "Near Field Communication Research Lab Hagenberg," 2013. [Online]. Available: http://www.nfc-research.at/

[25] STMicroelectronics, "RF Memories/Transceivers," 2013. [Online]. Available: http://www.st.com/web/en/catalog/mmc/FM76/CL1766