University of Amsterdam

MSc System and Network Engineering

Research Project 1

Portable RFID Bumping Device

Authors:
Romke van Dijk
romke.vandijk@os3.nl
Loek Sangers
loek.sangers@os3.nl

Supervisor:
Ari Davis
adavies@deloitte.nl

February 7, 2016

Abstract

This research focuses on creating a portable RFID Bumping device, and researching whether this device can successfully clone RFID tags within five minutes, specifically the MIFARE Classic and MIFARE Classic EV1. To create the portable device a custom antenna was made that extends the operational range of the device to about seven centimetres. An attack framework for the device was created and this framework can successfully clone about three tags that are within the operational range of the device, under some limitations, within five minutes. The attack framework exploits a few known weaknesses of the tags and can easily be extended.

Contents

1. Introduction

Radio-frequency identification (RFID) is a technology which is used for identification from a distance. It is based on radio waves for communication and can be compared to bar-code technology. The main difference between the two is, RFID does not require line of sight for the identification process [19]. It is being used for many applications, e.g. tickets for major events, public transportation, road pricing, access control systems and electronic passports [18].

The inspiration for this research comes from the article from Wired [14] which describes a device that is able to clone a RFID tag by bumping into a person. After the bump, the device has collected enough information to create a clone of the RFID tag and get access to a building.

When looking at access control systems, these are either based on low frequency tags or high frequency tags. Low frequency tags normally offer a very simple identification protocol. Whereas high frequency tags offer a more complex protocol, which includes mutual verification and transport encryption. The high frequency tags are also referred to as contactless smart cards.

In this research we will look at the feasibility of building a device as described above. To do so, there are a few aspects which have to be taking into consideration. The first aspect is time; the less time it takes to clone a tag the less chance of being detected. When talking about bumping, we mean a maximum of five minutes of tag interaction, as this is easily gained by for example starting a conversation or standing next to someone in a crowded train.

Another aspect is the distance between a RFID reader and the tag; the longer the reading range, the less likely it will be for someone to notice his tags are being cloned. With a large reading range, it might happen that there multiple tags within the range.

1.1 Research Questions

The main research question is: ”Is it possible to clone a RFID tag within five minutes using a portable device?”

The characteristics described above are translated in the following sub questions:

1. What is the maximal possible distance between the reader and tag?

2. How many tags can there be within the operational range of the reader?

3. What are possible attack vectors?

4. How much attack time is needed to clone the tag?

1.2 Scope

One of the most deployed high frequency contactless smartcards is the MIFARE Classic [13]. Studies have shown the MIFARE Classic contains weaknesses [13][2][12][5][6]. The MIFARE Classic EV1 is a hardened version of the MIFARE Classic, that does not include those weaknesses, but as shown by [13] an important weakness remains. Those weaknesses can be used in our cloning device. This research will focus on the MIFARE Classic and the MIFARE Classic EV1, because these are two of the most deployed contactless smartcards and have known attack vectors.

1.3 Structure

The structure of this report is as follows. Chapter 2 will discuss the related work. Chapter 3 will explain the literary research and the theoretical framework for our research. Chapter 4 will describe the methods and experiment conducted in this research. The results will be analysed in Chapter 5.

2. Related work

This chapter will briefly describe the related work for each sub question.

2.1 Range

Several papers indicate that the range between a reader and a tag can be extended up to approximately 20 to 30 centimetres. To reach this range, one needs to have a custom antenna, as a standard antenna only has a range of up to a few centimetres in practice [8] [11] [7].

2.2 Multiple cards

The MIFARE Classic is based on the ISO/IEC 14443-A standard. The standard contains an anti-collision mechanism which makes it possible to have multiple cards within the reader's range [10].

2.3 Attack vectors

Studies that have shown weaknesses in the MIFARE Classic are Meijer et al [13], Garcia et al.[5][6] and de Koning et al.[12].

Other papers that explain contactless smart cards and identify possible weaknesses are Tan[18], van Dullink et al.[4], Dimitrov et al.[3] and de Jong et al.[17]. Not all those papers look at practical attack vectors, but they provide general background information on how contactless smart cards work and indicate what the weak points in contactless smart card systems can be.

2.4 Attack time

Courtois[2] explains how much computation time is required to perform the DarkSide attack and Garcia et al.[6] showed the computation time required to perform a nested authentication attack. In 2010, Chih et al.[1] found that with a budget of 3000 euros it took 840 minutes to go through the full key space of the MIFARE Classic using GPUs. Meijer et al.[13] showed this key space can be reduced significantly because of weaknesses in the cipher, thus the computing power required can be reduced.

3. Literary Research

This chapter will describe the literary research that has been done to answer the sub questions. Each section will analyze the related work as described in Chapter 2.

3.1 Range

Range is an important aspect in cloning contactless smart cards. Increasing the range makes it both possible to increase gathering time and makes it less likely for someone to notice his cards are being cloned. The advertised operational range for the MIFARE Classic (and other 13,56 mHz based contactless smartcards) is roughly 10 centimetres [8], but practice shows that most readers' operational range is roughly a couple of centimetres. Kirschenbaum et al.[11] found and Hancke et al.[8] confirmed, it is possible to create a 14.8 cm x 20 cm antenna that extends the range to about 20 centimetres. This antenna is portable and thus suitable for bumping. Hancke et al.[8] also shows that it is possible to extend the range to roughly 27 centimetres with a 29.7 cm by 42 cm antenna. The Proxmark community designed a very small antenna that offers an operational range of 7 centimetres using a simple USB cable[1], which already extends the range by a couple of centimetres.

3.2 Multiple cards

When the operational range is increased, it is more likely that multiple tags will be within this range. It is important that the reader is able to distinguish tags. As described earlier, the MIFARE family is based on the ISO/IEC 14443-A protocol [10], which has a mechanism to allow for both distinguishing tags and selecting one tag. This is called the anti-collision mechanism. This mechanism, in theory, allows for any number of tags within the readers range, as long as each tag has an Unique identifier (UID).

A collision can occur when two tags reply at the same time but with a different answer, possibly making the resulting response invalid [9]. At the end of the anti-collision mechanism a tag can be selected using its UID. This

[1] https://github.com/Proxmark/proxmark3/wiki/antennas

card goes into the "active" state and will respond to high level commands while any other cards with a different UID will remain in an "idle" state [12].

Even though the anti-collision protocol allows the reader to select one card and communicate with just that card, the protocol does not support reading multiple cards concurrently. The difficult part with this is determining which reply belongs to which request. This is difficult because not all cards reply after the same interval.

To detect all tags within the operational range a Binary Tree Working Algorithm (BTWA) can be used. The ICU[9] describes such an algorithm and it operates as follows:

1. A reader chooses a '0' or a '1', the reader transmits this k-length prefix.

2. A tag response, with its UID if it kth bits of the tags UID is equal to the k-length prefix.

3. If the reader detects a collision, it will step into that branch and extends the prefix once with a '0' and once with a '1'. If no response was received, the branch is ignored. If no collision occurred, the reader can assume only one tag replied.

4. The reader repeats this procedure until all branches are searched through. Figure 3.1 is a simple representation of this.

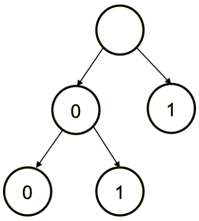


Figure 3.1: Binair Tree Working Algorithm

When working with passive tags[2], such as the MIFARE Classic and MIFARE Classic EV1, the amount of tags is limited by the power of the reader and the amount of tags that fit within the operational range of the reader.

3.3 Attack vectors

The attack vectors can be divided into two groups:

1. Weaknesses in low frequency contactless access control tags

[2] Passive tags are tags that needs to be powered by the reader, where active tags are self powered

2. Weaknesses in high frequency contactless access control tags (focus on MIFARE family)

3.3.1 Low frequency

Most low frequency tags do not contain cryptographic functions, when activated these will send out an UID. It would only require to activated the tag and capture the UID to be able to clone these tags.

3.3.2 High frequency

The MIFARE Classic is compatible with the ISO/IEC 14443-A standard except for the transmission protocol. It has implemented its own secure transmission protocol. The security is based on a proprietary stream cipher called CRYPTO1. Studies show that the cipher contains weaknesses [13][5][6][12].

In essence, a MIFARE Classic tag is a memory chip which offers authentication and secure transmission. Its memory is divided into segments, which are divided into blocks. These tags come in different sizes, for example the MIFARE Classic 1K which offers 1024 bytes of storage or MIFARE Classic 4K which offers 4096 bytes of storage. Each sector is protected by two keys, key A and key B. Each key can be used to allow certain operations, such as reading and writing. When a reader wants to perform an action, the reader has to authenticate itself first. The authentication process involves knowing the relevant key [5].

Acquiring one key only gives access to one sector with the access permissions that the key offers. Thus, for cloning a tag, an attacker is required to acquire all relevant keys of sectors that are used to, for example, verifying someone's identity in an access control system. After getting the relevant keys, all the relevant data needs to be copied. The following steps summarize cloning a contactless smart card:

1. Getting the keys
2. Getting the data
3. Recreating card

As stated before, we will focus mainly on the MIFARE Classic high frequency contactless smartcard, because it is one of the most deployed contactless smartcards. We will also look into the hardened version of the MIFARE Classic (called the MIFARE Classic EV1).

The procedure to clone a MIFARE Classic contactless access control card is as follows:

1. Check if default keys[3] exist on any sector (for example empty sectors or the first sector)

[3] Default keys are keys that are, for example, written on the card during manufactory. An example is: $0xFFFFFFFFFFFF$

2. If no default key is found: use for example the DarkSide attack [2] to retrieve the first key
3. Perform the nested authentication attack [6] to retrieve other sector keys
4. Use those keys to get the data sectors of the card
5. Copy the data and the keys onto a blank card to the same sectors

According to the MIFARE Application Directory[15] key A of sector 0 should be set to a default key ($0xA0A1A2A3A4A5$). In addition to this Garcia et al.[6] states that most deployed systems use default keys for unused sectors. Combining these observations shows that most implementations of the MIFARE Classic for access control will use at least one default key on the card. Thus, in most cases the DarkSide attack can be skipped.

If no default key is found, a DarkSide attack [2] could be performed to get one key. The DarkSide attack uses the weak pseudo random number generator, the parity bits, and the error codes given with wrong parity bits to find this one key. The attack needs about 300 queries to the MIFARE Classic card which takes about 10 seconds. After gathering the data, the computation needed is in the order of 2^{22} which on a laptop takes a few seconds.

After the first key has been found using either the DarkSide attack or just finding a default key, the nested authentication attack [6] can be performed. Using this attack all other keys used on the tag can be found. This attack is also based on the weak pseudo random number generator and the leaking of information through the parity bits. The MIFARE Classic sends parity bits which are computed over the clear text. The parity bits are encrypted with the same keystream bits which are also used to encrypt the next bits of plain text. This leaks information about the key and can narrow down the list of possible key candidates.

Using all the keys found using the nested authentication attack, all readable data of the tag can be read and therefore dumped. The dump of this data and the found keys can then be copied to an empty tag.

As described above the MIFARE Classic is vulnerable for a couple of attacks, as described in Meijer et al.[13] it is possible to defend against those attacks, but the CRYPTO1 cipher remains weak. In the same paper a ciphertext-only attack is published against the MIFARE Classic EV1 cards. This attack requires approximately 10.000 to 20.000 encrypted nonces to extract a sector key with a high probability. This will take 6 to 12 minutes of card interaction.

This attack does not necessarily make it possible to clone a MIFARE Classic EV1, because to clone a tag you would need all the keys and all data. If, however, it is possible to get access to the same card twice it is possible to split the attack. During the first interaction gather the the encrypted nonces and the second time gathering the data using the found keys.

Some high frequency contactless smart card based deployment only use the UID without using any cryptographic functions. To be able to clone these it is only needed to capture the UID. This makes it very similar to cloning low frequency tags. When talking about MIFARE Classic deployments, this goes against the deployment guidelines of the manufacturer NXP[15].

Another contactless smart cart is the MIFARE DESFIRE, which can be deployed in either 3DES mode or AES mode. Oswald at al. [16] show it is possible to extract the 3DES key from a MF3ICD40 with approx. 250.000 traces. This takes a couple of hours to gather, making this attack infeasible for a bumping attack. As for now, no other card attack seems to be available in the public domain.

3.4 Attack time

The time to clone a card is already described in Section 2.

4. Methodology

For this research a Proxmark 3[1] was used. It enables sniffing, reading and cloning of RFID tags. The Proxmark 3 was connected to a LG Nexus 5 running Android version 4.4.4 together with Nethunter[2]. The Nexus offers a good battery, multiple methods of connection, enough computing power and is portable. This makes it a good solution for the controlling part of the bumping device.

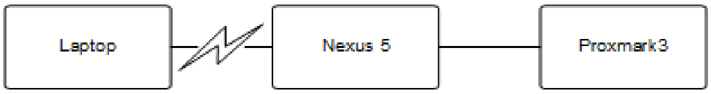


Figure 4.1: Bumping device

Figure 4.1 shows the bumping device. The Nexus 5 contains the Proxmark3 client software and is connect to the reader (the Proxmark 3) via USB. The laptop can be used to connect to the Nexus 5, for example over a WiFi connection, to execute commands, but a laptop is not required because the Nexus 5 itself can also be used to execute commands.

In all the experiment either a blank MIFARE Classic 1K or a MIFARE Classic EV1 1K was used.

4.1 Range

The range will be increased using a Hirose usb cable[3]. It should extend the range to roughly 7 centimetres. This will be verified using multiple tags and measuring the operational range of the reader.

4.2 Multiple cards

As for now, the Proxmark firmware does not support reading multiple tags, it has implemented the ISO/IEC 14443-A anti-collision mechanism, but in a limited way. If multiple tags are within the reader's range, it will only interact and recognize one tag. Which tag depends on multiple aspects, such

[1] http://www.proxmark.org/

[2] https://www.offensive-security.com/kali-linux-nethunter-download/

[3] Design can be found on https://github.com/Proxmark/proxmark3/wiki/antennas

as the range between tags and reader and the computing speed of the tags. By implementing the BTWA as described in section 3.2, we make it possible to recognize multiple UIDs using the Proxmark3. After implementing this protocol, we will research how many cards can be in the operational range of the reader.

4.3 Attack vectors

During the literature study we found multiple vulnerabilities of the MIFARE Classic. Most of them are already implemented in the Proxmark software. Based on the steps described in Section 3.3.2, we designed an attack framework. This section will describe the implemented parts of the framework, the full designed framework can be found in Appendix A.

After starting, the framework comes into the main loop, which will search for any low frequency or high frequency tags within the operational range of the reader. Once it detect a low frequency tag, it will copy the UID, check whether the UID already exists in the database and save the UID to the database if it did not exists. The low frequency attack steps are shown in Figure 4.2

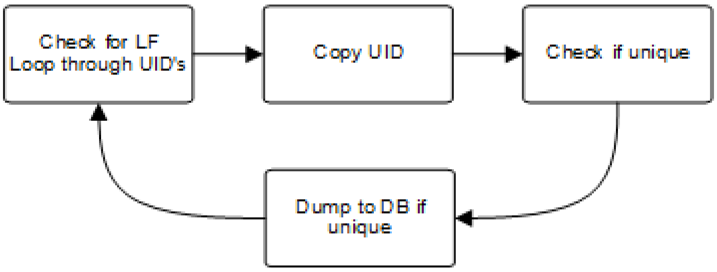


Figure 4.2: Attack steps for low frequency tags

Attacking a high frequncy card requires more steps. Firstly, it will perform the BTWA to detect all the UIDs within the operational range of the reader. Secondly, it will loop through all the tags and checks if any of them are uncompleted. Uncompleted means tags that have not been fully cloned (having the keys and data of all the sectors). The next step is to check for any default keys on the tag. If there are any, a nested authentication attack can be executed. If that attack fails, a hard nested authentication attack can be performed. This attack still requires the leftover keyspace to be solved after the tag interaction. Once all the keys are recovered, the data can be read from the card and saved to the database. Figure 4.3 is a graphical representation of those steps. We have implemented the high frequency attack steps in the Proxmark client and firmware.

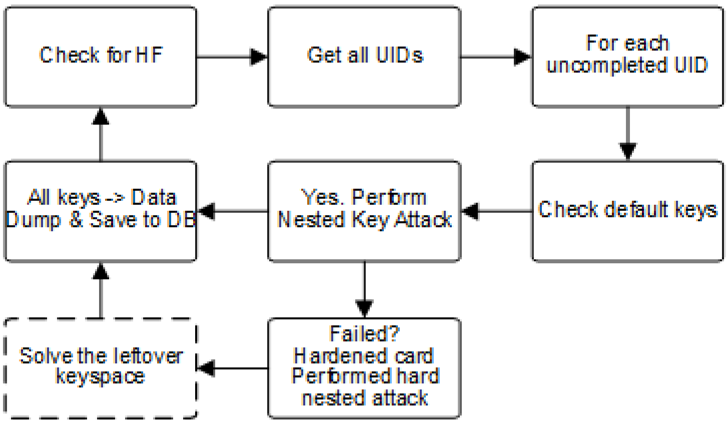

Check for HF
Get all UIDs
For each uncompleted UID
Check default keys
Yes. Perform Nested Key Attack
All keys -> Data Dump & Save to DB
Failed? Hardened card Performed hard nested attack
Solve the leftover keyspace

Figure 4.3: Attack steps for high frequency tags

4.4 Attack Time

The time it takes to perform a full attack is very important for a bumping device. To be able to measure the duration we designed and conducted the following experiments:

- Attempting to clone a MIFARE Classic tag with default key A $0xA0A1A2A3A4A5$ at sector 0 and n random keys A for the other sectors. For each n the experiment was conducted 100 times.
- Attempting to clone a MIFARE Classic EV1 tag with default key A $0xA0A1A2A3A4A5$ at sector 0 and n random keys A for the other sectors. For each n the experiment was conducted 100 times.

For the first experiment we created a shell script which generates n random keys. Key k is the k-th random key and will be written key A of block $k * 4$. For the first experiment a maximum of 60 iteration for the nested authentication was configured. This was because timing is critical for a nested attack, it is based on exploiting the pseudo random number generator. The Proxmark software tries to find the exact timing for the tag to always give the same nonces, but sometimes fails to do so. The maximum is added to protect against a never ending loop.

For the second experiment a maximum of 10.000 gathered encrypted nonces was configured. This is to prevent the Proxmark software to gather nonces indefinitely. The Proxmark software tries to collect enough nonces with a certain property, but in some cases it requires a vast amount of encrypted nonces. For more details see Meijer et al.[13].

5. Results and Analysis

5.1 Range

Using the designs from the Proxmark community, we created a HF Hirose Antenna with a budget of 5 euros. It increases the operational range from 3-5 centimetres (with the default antenna) to roughly 6-8 centimetres when interacting with one tag. The range depends on the type of tag and placement compared to the antenna. When a tag is placed above the center of the antenna, the range is at its best. When placed near the sides of the antenna, the range decreases but it is still able to detect a tag. As described by the Proxmark community, the antenna is optimized for card shaped tags.

5.2 Multiple cards

We successfully implemented the BTWA and are able to interact with multiple tags using the Proxmark 3. Our setup is able to interact with three tags stacked on top of each other within a range of 4 centimetres. During the research we noticed that we were able to interact with more than three tags, when there was enough space between the tags, but we did not research this further because when cloning RFID tags while bumping, it is more likely that RFID tags will be stacked together (for example in a wallet).

5.3 Attack vectors

We have implemented a framework which loops through all the available tags within the operational range of the reader and attempt to clone those cards based on the steps described in section 3.3.2. Combining the changed Proxmark firmware, the antenna, and the phone we were able to create an easily portable bumping device. This device is able to consistently bump and clone up to three cards within the antenna range. All the gathered data is written to a database on the phone and can be accessed by connecting to the phone. The phone can also be used to send the data needed for heavy computations to a dedicated off site machine.

Also, because we use a database, the framework is able to keep track of state. Whenever a card is offered that is already attacked and fully cloned,

keys	mean	standard error
1	6.45	0.815
2	45.81	8.452
3	67.54	10.443
4	93.98	16.364
5	133.1	28.33

Table 5.1: Average time in seconds for a nested attack on x keys

the framework skips this card. If the card is not fully cloned, the framework can continue where it left off previously.

5.4 Attack time

The experiments we conducted into the time needed to attack the two types of MIFARE Classic tags will be analysed in this section. Firstly, the standard MIFARE Classic will be discussed and secondly the MIFARE Classic EV1.

5.4.1 MIFARE Classic

We were able to conduct the experiment as described in Section 4.4 with changing up to 5 keys. Changing more keys was not possible due to the battery of the phone running out of energy before being able to clone the tag 100 times. Table 5.1 shows the found duration of the attack in seconds.

In Figure 5.1 the average time for a nested attack on x keys is plotted. It also shows the fit-line ($f(x) = 30x - 21$) for the gathered data points. This line helps to predict that, on average, within five minutes 10 keys can be found.

Given that for the nested attack at least one key needs to be known, we tried to predict the time it will take to clone a card with only non-default keys except one (31 non-default keys). Calculating this results in: $f(31) = 909$, which is approximately 15 minutes.

In total we performed this experiment with 2006 keys and were able to retrieve 1628 (81%) of them successfully. This is due to the fact that the nested authentication attack is based on exploiting the weak pseudo number generator. It has to find the exact timing that the same nonces will be repeated. Sometimes the Proxmark simply fails to do so. When the success rate of the Proxmark calibration increases, so will the success rate of our setup.

5.4.2 MIFARE Classic EV1

The attack on the MIFARE Classic EV1 has three parts. First a number of nonces has to be gathered, then using those nonces the key has to be computed offline, and finally a second interaction with the tag is needed to clone tag's data. In our experiment we looked at the time it takes to

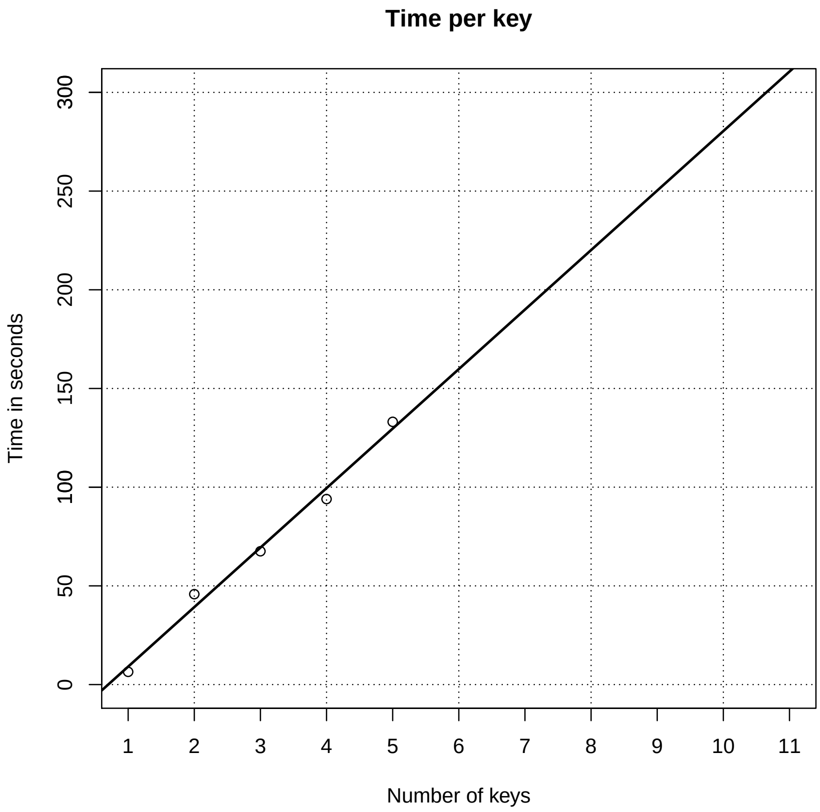


Figure 5.1: Time per key for nested attack on MIFARE Classic tag

gather the nonces and the leftover keyspace. Using our setup, we found that we could gather 55 nonces per second very consistently. In Figure 5.2 the number of nonces gathered per second is shown for one key. In Figure 5.3 the relation between nonces and leftover complexity of the keyspace is shown.

As can be seen in Figure 5.3 there is no linear relation between the leftover keyspace and the number of nonces. This is due to the randomness of the nonces and the way the cipher works. The separation between left and right is caused by the fact that we either wait for the sum property to be high enough (see [13] for more information) or just stop after gathering 10.000 nonces. In case we stop gathering nonces when we hit the sum property threshold, the leftover complexity is in general lower than in case we stop gathering after 10.000 nonces. The complete keyspace is 2^{48} and as Figure 5.4 shows, the keyspace left using the gathered nonces is on average approximately 2^{34}, which will decrease the brute force time needed drastically.

During every experiment the default key check only required 2 to 5

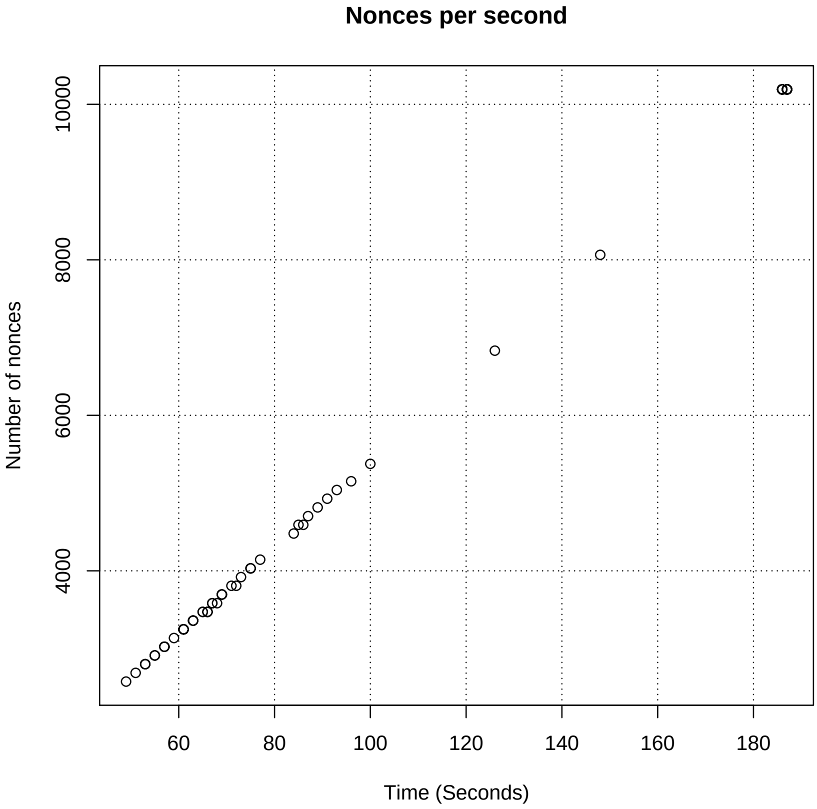


Figure 5.2: Nonces per second for ”hard” nested attack on MIFARE Classic EV1 tag

seconds. Reading the data after all the keys were found only required 4 to 5 seconds. Thus, the most time was spent in acquiring the keys.

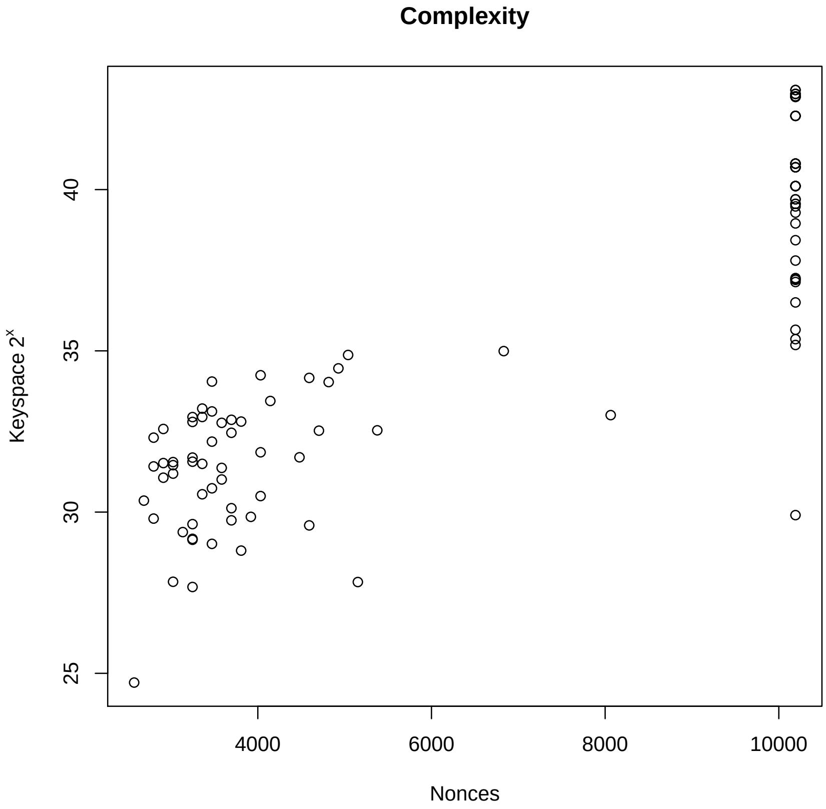


Figure 5.3: Leftover keyspace for number of gathered nonces

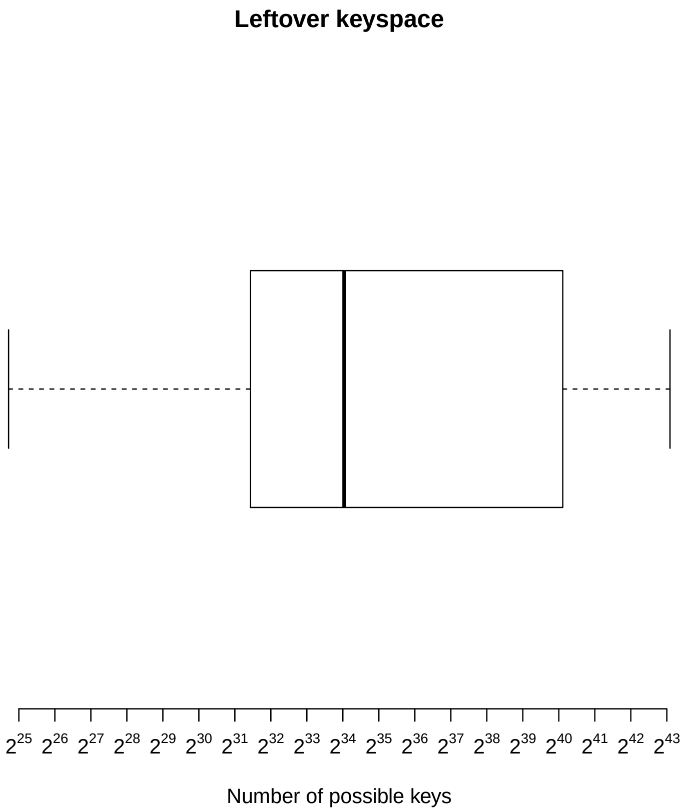


Figure 5.4: Leftover keyspace

6. Conclusion

This research has shown that it is possible to create a portable RFID bumping device that is able to clone MIFARE Classic 1K tags. Within five minutes our device is able to clone tags that have no more than 10 non-default keys. This all with a budget of under 300 euros. We have also found that using our device we are able to clone tags that are within a range of 6 to 8 centimetres. Our device has implemented a BTWA which allows interaction with multiple tags sequentially. Because of limitations of our antenna, the maximum amount of tags that can consistently be attacked is 3.

There are several attack vectors we found for attacking the MIFARE Classic, and one attack vector for the MIFARE Classic EV1. The MIFARE Classic has several weaknesses that are exploited by those attacks, such as a weak random number generator and error codes in specific cases. These weaknesses have been fixed in the MIFARE Classic EV1, but the cipher has remained the same. This cipher also has a known weakness that can be exploited when trying to gather all the keys. This weakness does, however, require offline computation. Thus a trade-off has to be made between the online gathering time and offline computation time. Currently our cloning device is limited to gather enough nonces for two keys within five minutes. If the total attack time can not be reduced to 5 minutes, the attack will require a second interaction with the same tag after the keys are solved.

Our device is keeps track of state, which makes it a practical device. It remembers a tag by its UID and it remembers which keys it has already found. If our device interacts with a tag for the second time, it will know which keys and data it still has to attack and start doing so.

To protect against our bumping device, one has to use the MIFARE Classic EV1 tag and change all the default keys to non-default keys. This is necessary, because the hard nested authentication attack requires knowledge of at least one key. As there is no attack publicly known for the MIFARE Classic EV1 to gain knowledge of a key without prior knowledge of another key. A better solution, however, is to migrate to a different technology that does not have known vulnerabilities and has been validated by multiple independent parties.

7. Future Work

During this research we have implemented a framework that can clone certain high frequency RFID tags (MIFARE Classic and MIFARE Classic EV1). There are a few future extensions to this framework and other aspects of this research that can be extended or researched more in depth.

Firstly, the maximum number of tags can be more extensively researched. During our experiments we could consistently read three stacked tags at the same time, but when placing them differently more tags could be read. We expect that it is caused due to tags blocking signal when stack together, but did not research it more in-depth.

Secondly, the implemented framework (modification to the Proxmark) can be extended to include attacks on other tags that are commonly used. The attack time for those tags can then be research as well.

Thirdly, the software to calculate the (leftover) keyspace can be optimized, and using this a better estimation of the actual safety of the MIFARE Classic EV1 can be given. The calculations with software that is publicly known now, take longer than the for this research allowed five minutes. It is, however, possible that other software will be able to calculate and test all possible keys within a much shorter timespan.

Fourthly, to see the real impact of this research, one could use the bumping device in an unstable environment to clone a tag. We did not do this because of ethical reasons.

Lastly, the Proxmark firmware can be extensively researched and tested, in order to optimize it. This could enable better testing of the weaknesses of time based pseudo random number generator within RFID tags.

Acknowledgements

We would like to thank Ari Davies from Deloitte for giving us the opportunity to work on this research and providing all the equipment we needed, Jan van Erven of NFC Nederland for proving extra hardware, , Marijke Duyvendak for helping us getting the paper written by Chih et al.[1], and Dana Geist, Nancy Kluvers, and Marat Nigmatullin for reviewing our report. Furthermore, we would like to thank Blapost for providing us with a solver for the hard nested authentication attack and the Proxmark community for providing us with a lot of information and the opensource firmware and software.

Bibliography

[1] Ming-Yang Chih et al. "MIFARE Classic: Practical attacks and defenses". In: *National Information Security Conference.* Chinese Cryptology and Information Security Association. 2010, pp. 126–132.

[2] Nicolas T Courtois. "The dark side of security by obscurity and cloning Mifare Classic rail and building passes, anywhere, anytime". In: (2009).

[3] Hristo Dimitrov and Kim van Erkelens. "Evaluation of the feasible attacks against RFID tags for access control systems". In: (2014).

[4] Wouter van Dullink and Pieter Westein. "Remote relay attack on RFID access control systems using NFC enabled devices". In: (2013).

[5] Flavio D Garcia et al. "Dismantling MIFARE classic". In: *Computer Security-ESORICS 2008.* Springer, 2008, pp. 97–114.

[6] Flavio D Garcia et al. "Wirelessly pickpocketing a Mifare Classic card". In: *Security and Privacy, 2009 30th IEEE Symposium on.* IEEE. 2009, pp. 3–15.

[7] Rene Habraken et al. "An RFID Skimming Gate Using Higher Harmonics". In: ().

[8] Gerhard P Hancke et al. "Practical eavesdropping and skimming attacks on high-frequency RFID tokens". In: *Journal of Computer Security* 19.2 (2011), pp. 259–288.

[9] Okkyeong Bang ICU et al. "Efficient Novel Anti-collision Protocols for Passive RFID Tags". In: *no. March* (2009).

[10] ISO. *ISO/IEC 14443. Identification cards – Contactless integrated circuit cards – Proximity cards.* 2008. URL: http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=39693 (visited on 12/05/2015).

[11] Ilan Kirschenbaum and Avishai Wool. "How to Build a Low-Cost, Extended-Range RFID Skimmer." In: *Usenix Security.* 2006.

[12] Gerhard de Koning Gans, Jaap-Henk Hoepman, and Flavio D Garcia. "A practical attack on the MIFARE Classic". In: *Smart Card Research and Advanced Applications.* Springer, 2008, pp. 267–282.

[13] Carlo Meijer and Roel Verdult. "Ciphertext-only Cryptanalysis on Hardened Mifare Classic Cards". In: *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*. CCS '15. Denver, Colorado, USA: ACM, 2015, pp. 18–30. ISBN: 978-1-4503-3832-5. DOI: `10.1145/2810103.2813641`. URL: `http://doi.acm.org/10.1145/2810103.2813641`.

[14] Annalee Newitz. *The RFID Hacking Underground*. Wired. Jan. 5, 2006. URL: `http://www.wired.com/2006/05/rfid-2/` (visited on 12/05/2015).

[15] NXP, ed. *MIFARE application directory*. Jan. 6, 2013. URL: `http://cache.nxp.com/documents/application_note/AN10787.pdf`.

[16] David Oswald and Christof Paar. "Breaking mifare DESFire MF3ICD40: power analysis and templates in the real world". In: *Cryptographic Hardware and Embedded Systems–CHES 2011*. Springer, 2011, pp. 207–222.

[17] Dirk-Jan van Helmond Richard de Jong and Matthijs Koot. *An exploration of Radio Frequency Identification (RFID)*. University of Amsterdam, Dec. 8, 2005.

[18] Wee Hon Tan. "Practical attacks on the Mifare Classic". In: *Imperial College London* (2009).

[19] Roy Want. "An introduction to RFID technology". In: *Pervasive Computing, IEEE* 5.1 (2006), pp. 25–33.

Appendices

A. Attack framework

This appendix will give an overview of the whole attack framework. When the framework boots it will first tune the antenna. This is to get the best operational range for the reader. Next, it will start the main loop. Figure A.1 shows this progress[1]. The main loop consist of two functions. LF Attack Loop and HF Attack Loop, which loops either through all low frequency tags or through all high frequency tags.

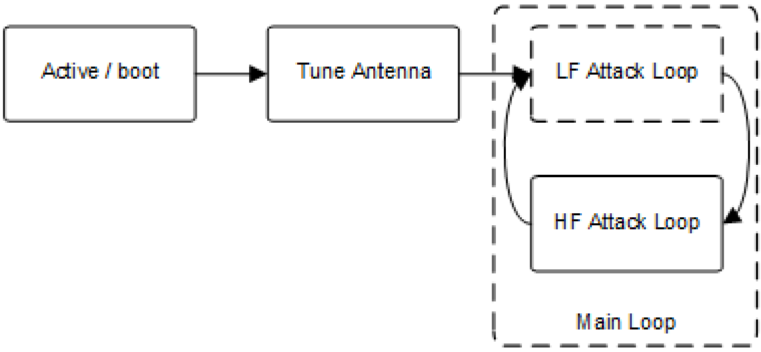


Figure A.1: Main loop

The low frequency attack function is already described in Section 4.3. Figure A.2 shows this function again. Basically when it detect a tag, it will copy the UID and save it to the database.

The high frequency attack function is already described in Section 4.3. Figure A.3 shows this function again. The designed framework will first check for any high frequency tags, if it detect for example an ISO/IEC 14443-A tag, it will determine which specific type of tag it is. In case of a MIFARE Ultralight, it will call the specific function for that card. This framework is extendible, new protocol tags or specific new tags can easily be added.

[1] Dotted functions in any of the images in this appendix indicates that the function is not implemented in our software.

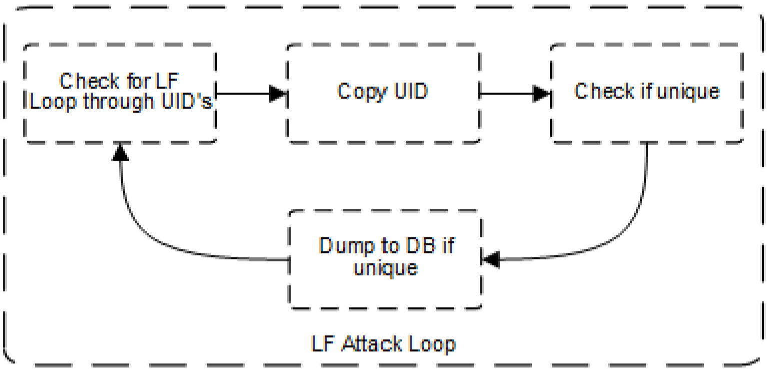
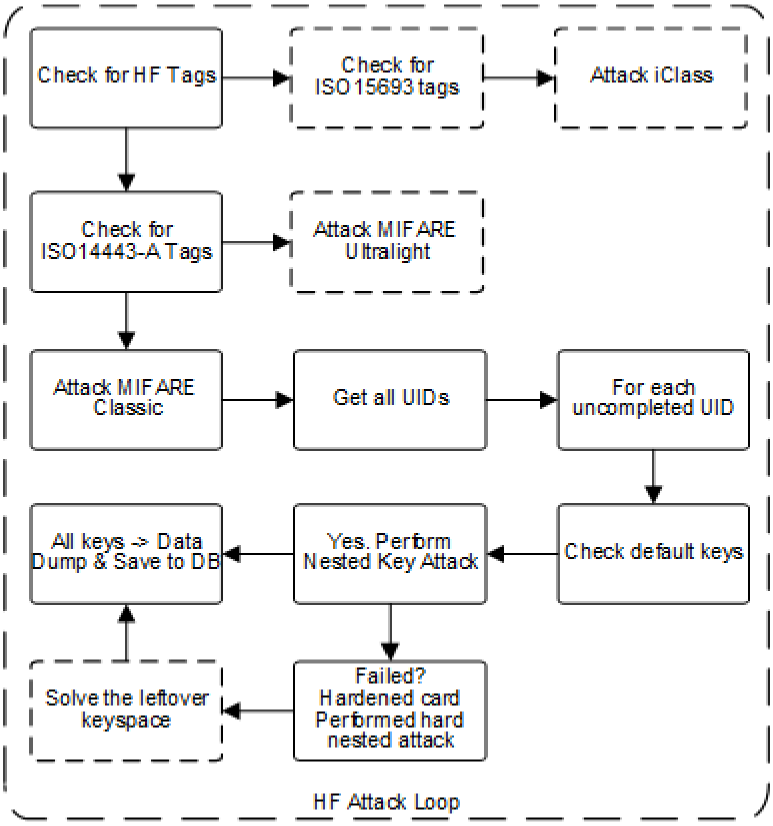


Figure A.2: Low frequency attack loop



Check for HF Tags
Check for ISO15693 tags
Attack iClass
Check for ISO14443-A Tags
Attack MIFARE Ultralight
Attack MIFARE Classic
Get all UIDs
For each uncompleted UID
All keys -> Data Dump & Save to DB
Yes. Perform Nested Key Attack
Check default keys
Solve the leftover keyspace
Failed? Hardened card Performed hard nested attack
HF Attack Loop

Figure A.3: High frequency attack loop

B. Software

All the software that has been created for this project can be downloaded from github.

- https://github.com/zyronix/proxmark3/
- https://github.com/zyronix/rfid-bumping/