university of groningen

faculty of mathematics and natural sciences

University Card security

Bachelor's thesis Computing Science

June 29, 2015

Student: Thomas Rinsma

Primary supervisor: dr. F.B. Brokken

Secondary supervisor: prof. dr. G.R. Renardel de Lavalette

Abstract

Over the past couple of decades, systems and protocols based on personal Radio-Frequency Identification (RFID) tags and specifically on contactless Smart Cards have seen a large increase in use. One such system is the so-called University Card of the University of Groningen. Every employee and student of the university owns such a card. These are contactless Smart Cards based on a chipset by NXP called Mifare DESFire EV1. The cards can be used to pay for drinks at beverage machines at all the university buildings, for parking lot access and for printing and photocopying. Systems like this are very convenient for their users, but when implementing them one has to be extra diligent about keeping every aspect of the system secure. This thesis tries to find out whether that is the case in this system. It contains an analysis of the University Card and of the various systems around it. Based on this analysis, several methods of user impersonation are described, including the cloning and emulation of University Cards. These methods are brought to practice and tested on the system itself, resulting in successful authentication of a user without their original University Card at both the printer systems and the parking lot gates.

CONTENTS

1

INTRODUCTION

1.1 INTRODUCTION

Systems and protocols based on personal RFID tags have seen a large increase in use over the past couple of decades. There has especially been an increase in the adaption of so-called contactless Smart Cards [7]. As the name suggests these cards are like regular Smart Cards (i.e., credit-card sized cards containing an integrated circuit [8]), except that the communication with a terminal occurs via 13.56 MHz radio waves instead of a contact pad [10]. So that instead of inserting them into a slot, they can be accessed by holding them in front of a terminal. This ease of use is probably a big factor in the increase of their use in various systems around the world.

One such system is the so-called University Card of the University of Groningen. Every employee and student of the university owns such a card [35]. These are contactless Smart Cards based on Mifare DESFire EV1. The cards can be used to pay for drinks at coffee and tea machines, for parking, and for printing and photocopying. Each card is tied to the user's university account, which includes a money balance. Users can add money to their balance through an online system. The payment terminals thus have access to this system, and can use this system to determine which user is the owner of a scanned card before deducting the appropriate amount of money from their account.

Systems like this are very convenient for their users, but when implementing them one has to be extra diligent about keeping every aspect of the system secure. Especially so when money or sensitive data is involved, which is the case here. One aspect to keep in mind is the communication between a contactless Smart Card and the terminal, which happens over the air. Compared to a traditional contact smart card this is much easier to intercept. For this reason some contactless Smart Cards have hardware implementations of cryptographic ciphers like (triple-) DES, AES and/or proprietary ciphers to secure their communications and stored data. Both DES and AES are publicly defined symmetric key encryption algorithms. Their main advantage over proprietary algorithms is that everyone is able to independently verify their security. Proprietary encryption algorithms are not publicly available and have the advantage of obscurity: they make it harder for an attacker to analyze the algorithm and therefore harder to devise an attack.

Yet even though some of these ciphers are currently regarded as secure by themselves, they have to be implemented correctly. An example of such a flawed implementation is the use of a bad pseudo random number generator, as is the case in Mifare Classic 1K cards. [15] Furthermore, even if a card does in fact support secure communication or even challenge-response based cryptographic authentication, these features have to be used.

1.1.1 *Project overview*

In this thesis we take a look at the University Card system with these and related aspects in mind. We pose the following questions:

- Is it possible to impersonate another user by presenting the system with information that links to his or her account using techniques like card cloning and emulation?
- If so, what are the security and privacy implications of such an action?

To answer these questions, we do the following:

- We use publicly available information like technical documentation of the various terminals and the chipsets they support to gain a deep understanding of these components and the system as a whole.
- We use ordinary tools like an NFC-capable Android smartphone to read the contents of a University Card and to analyze the structure of, and differences between various (types of) University Cards.
- We analyze the current state of various contactless smart cards in terms of standards compliance, supported features and the security related research that has been done on them.
- We devise two attack strategies on the University Card system for the impersonation of a user, based on available hardware and security research.
- We perform the attacks on three different systems in which the University Card is used for authentication.

1.2 RELATED WORK

In order to better analyze the security of the system at the University of Groningen, we need to gain an understanding of the state of systems like it elsewhere in the world and the research that has been done on them. Looking at the different ways in

which contactless Smart Cards are used, we find a highly varied ecosystem: many different protocols and technologies exist that build on top of the hardware level protocols. Depending on the use-case, each of these technologies has its advantages and disadvantages.

Examples of contactless Smart Card use include identification for access in corporate and academic environments (e.g., for access control to certain buildings or rooms) or for governmental purposes [31] [5] [26] [9] [19], presence and attendance logging [20] [3], but also for example to track garbage disposal use per household [18]. Another use for these cards is authentication for payment. In these systems the card acts as a wallet, allowing the user to pay by tapping their card in front of a reader. Depending on the system the reader will either change the contents of the card, change the user's balance in the back-end [11], or perform some combination of these two operations. Many different chips and standards are in use for this purpose.

Examples of such payment systems are the various public transportation cards used around the world like the Dutch OV-Chipkaart and the London Oyster card which were both initially based on NXP's Mifare Classic technology (the latter has now switched to Mifare DESFire cards) [33] [34]. The Mifare Classic card has been a target of security research for a while, and has by now been thoroughly broken. Nohl et al. exposed several design-flaws in its security after performing image analysis on the circuits, demonstrating the difficulty and futility of keeping such a cipher secret, even in hardware implementations [21]. Thereby validating Kerckhoffs's principle: A cryptosystem ought to be secure even if everything about the system, except for the key, is public knowledge [12].

Garcia et al. [4] managed to reverse engineer Mifare Classic's authentication protocol, its proprietary symmetric cipher, and its initialization mechanism, allowing them to "[recover] the secret key from just one or two authentication attempts with a genuine reader in less than a second on ordinary hardware and without any pre-computation". They were able to travel for free on the London subway and on Dutch trains using cloned cards [4].

More recent than the Mifare Classic cards is NXP's Mifare DESFire family of cards. These are supposed to be a more secure alternative to Mifare Classic cards. Several transit systems are based on these as well [2] [29] [30]. The Oyster card as mentioned above has also switched to this technology (specifically the DESFire EV1) in 2009. Security-wise, Mifare DESFire's main advantage over Mifare Classic is that it uses the well known 3DES cipher as opposed to the proprietary *Crypto1*. Nevertheless, the original DESFire (MF3ICD40) has flawed security as well. Using non-invasive side-channel analysis it is possible to recover the secret key of the 3DES algorithm [27]. NXP has since

discontinued this card and recommends their customers to use DESFire EV1 instead, which is a newer backwards compatible technology which has not yet been found to be vulnerable to such side-channel attacks [24].

Other public transportation systems are based on Sony's FeliCa chip, including Hong Kong's Octopus Card, Singapore's EZ-Link and the Japanese family of so-called IC cards [25] [14] [16]. Interestingly, many of these have seen a spread in use outside of their originally intended scope of transit systems. They are starting to be accepted in shops and convenience stores as well, making them generic rechargeable payment cards [39]. Singapore's EZ-Link is even used for identification purposes as well [6]. This spread of use means that there is more money and more private information at stake, making its security extra important. Not to mention the issues around customer privacy that come up in such a centralized system. No serious flaws in Sony's FeliCa have yet been published however.

2

THE UNIVERSITY CARD SYSTEM

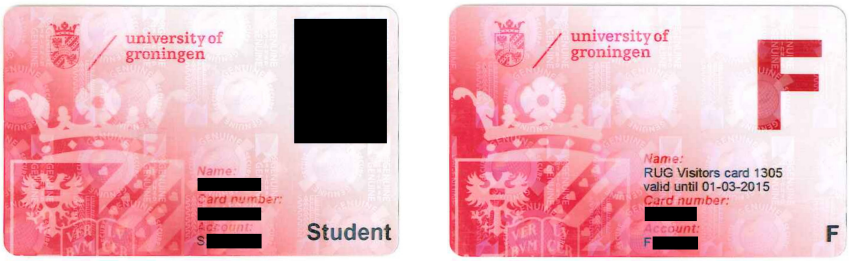
For the purpose of analysis, the system at the University of Groningen around the University Card can be divided into two parts: the card itself and the devices that can interact with it. We take a detailed look at both of these, collecting all relevant information to paint an accurate picture of the system as a whole.

2.1 THE UNIVERSITY CARD

The University Card was first introduced for employees in early 2013. By the start of the academic year 2013-2014 it was fully rolled out for students as well [13]. It was meant to replace various systems which each had separate passes and cards by combining them into one card per person.

2.1.1 *The physical card*

On the outside, the University Card has a several pieces of information printed on it: The card holder's name, their function (one of either *Student*, *Employee* or *Visitor*), their account-number (starting with respectively `S`, `P` or `F`), a card number and finally a barcode on the back containing a similar yet slightly different number. Additionally, if the card holder is not a visitor their photo is printed on the card. Shown in figure 1 are the front sides of a student card and a visitors' card. Employee cards are visually similar to student cards but differ only by the first letter of the account number and the absence of the word "Student".



(a) My anonymized student card (b) An anonymized visitors' card

Figure 1: The different types of University Cards

2.1.2 *The card contents*

We use a standard NFC-capable Android smartphone to read the contents of University Cards. Specifically, the *NFC TagInfo* by the Research Lab Hagenberg is used because of its support for Mifare DESFire EV1 and its structured way of displaying card contents.

As explained in the introduction, the University Card is based on Mifare DESFire EV1 technology, specifically the variant with 4KB of non-volatile memory (model number MF3ICD41). This memory is exposed by the card's operating system as a series of *applications*, with each of those able to contain a series of files. The term *application* might be slightly confusing in this context, but it is the term used by NXP for the standardized data-structures stored on Mifare cards that can have certain properties like specialized access keys.

In this case the card contains a single *General Issuer Application*, containing two files: *Card Holder* containing the name of the card holder (i.e. the student or employee) and *Card Publisher* which contains the string `University of Groningen`. Both of these strings are basically stored in *plain text*: They can be read by any reader without the need to provide a key.

Apart from the aforementioned files the card's memory is not used. Compared to the amount that is printed on the outside of the card and the amount of available memory inside the card, this is a very small amount of information. Perhaps more importantly, it is all static information, meaning that there is no such thing as a changing credit or a balance being stored on the card.

If the terminals use the card's contents for identification purposes, they must then be using the *Card Holder* string because it is the only piece of content that differs between cards. However because this string is not guaranteed to be unique – since multiple people can have the same initials and surname – we can rule out this possibility. Authentication must therefore be happening purely on the basis of values obtained by *PICC-level* commands. These are commands defined in ISO 14443 which are exchanged during the initialization process to communicate certain properties of the card to the terminal and to help the terminal identify the card and its type to prevent *collision* when multiple cards are within range.

Examples of such values are the Unique Identifier (UID), Select Acknowledge (SAK), Answer To Request type A (ATQA) and Answer to Select (ATS) values. The latter three of which are usually used to identify the *type* of a card [23], but they could also be used in combination with the UID to uniquely identify a card.

The University Card can be used to interact with several devices within the premises of the University of Groningen. These so-called terminals are manufactured by several different companies according to different specifications. We look into each of these systems separately to find out what their relevant use-cases are how they use the University Card.

2.2.1 *Beverage machines*

Early 2013 saw the introduction of new beverage machines in almost all of the university buildings [13]. This is the *Gallery 310*, provided to the university by the Dutch coffee and coffee machine producer Douwe Egberts. It serves several types of hot drinks like coffee and tea. Built into all of these machines is an NFC reader by a company called Inepro. Users are able to buy drinks from these machines by providing their University Card to this reader. When buying a drink, the machine will subtract the cost of the drink from the user's balance in the back-end. The user can also see their current balance by just holding their card in front of the reader without choosing a drink.

2.2.2 *FollowYou printing*

FollowYou printing is a system by a company called Equitrac which allows users to submit a print job from any PC on the University domain and choose at which printer they want the document to be printed. The user logs in at the desired printer by either entering their username and password, or by using their University Card. The system will then allow the user to print any queued jobs and it will subtract the appropriate amount of money from the user's balance. This is the same balance that is used when buying drinks at beverage machines.

As is described in the introduction, the smart card reader that is used for the FollowYou system (figure 2) supports many different types of cards [22]. This is (partly) because in addition to logging in, the FollowYou system has another feature: the ability to link a yet unlinked card to an account.

When providing the reader with a smart card with any of the supported types of chips, it will look in the back-end to see whether that card has already been linked to an account. If it has not, an interface pops up where the user can link this card with an account by logging in manually using their username and password. Once this action is performed, the card or tag in question can be used at the beverage machines as well.

This is a feature that was extensively used by students before the University Card was introduced. At that time it was the only way to link a smart card to a student account, which initially had no card linked to it. In practice many students chose to link their OV-Chipkaart to their accounts. From a security perspective this was worse than the current situation, because as described above, the OV-Chipkaart is based on Mifare Classic, which has been thoroughly broken and can be cloned completely.

Despite the introduction of the University Card resulting in a default linked card for every primary account, this feature has not been removed. Teaching assistants are an example of users who have more than one university account (an `S` and a `P` account). They can and do still use this feature to link their secondary account to one of their own smart cards.

2.2.3 *Parking lot gates*

The third system that makes use of the University Card is the collection of parking lots around the Zernike campus. Employees who meet certain health or travel distance conditions receive the ability to park at these parking lots by using their University Card. The parking lot gates at the entrance and exit of every lot have an NFC reader to which the employee must provide their card before the barrier will open. The terminals at every gate are the *WPS-BC Easy* model by WPS Parking Systems. They support both contactless smart cards and temporary paper tickets [37].

Before the introduction of parking abilities linked to University Cards, employees had a separate Mifare Classic-based card to use for entering and exiting parking lots. Once again, the introduction of the University Card seems to be an improvement in terms of security, considering the broken security of Mifare Classic cards.

Figure 2: The external Equitrac smart card reader which is attached to every FollowYou printer

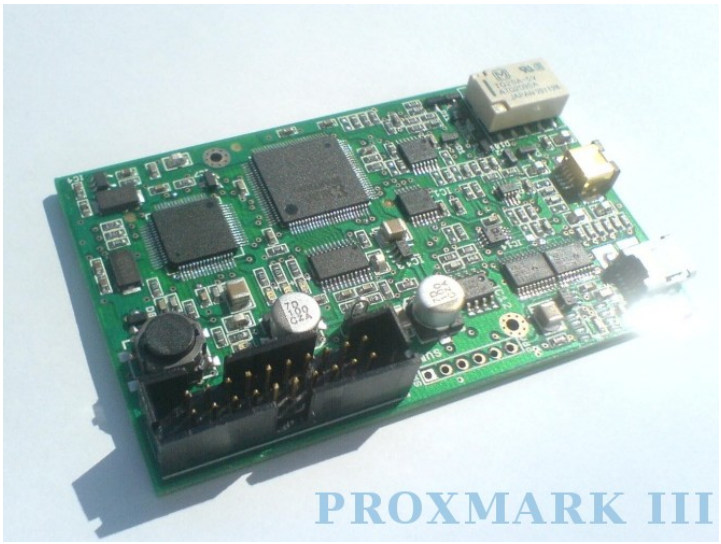


Figure 3: The Proxmark3

2.3 SECURITY CONCERNS

After having analyzed the University Card and all the different system it can be used with, one aspect stands out the most: identification (and thereby authentication) seems to happen simply by comparing certain static low-level values on the card like the UID to their versions stored in the back-end. The issue is that as opposed to a cryptographic challenge-response based system, this does not guarantee the identity of the card holder and is susceptible to impersonation: an attacker only needs to replay the data that is being transmitted during the identification phase (which is in this case the anti-collision phase of ISO14443A [7]) to be able to successfully authenticate himself [17].

Such an attack is not just theoretical but highly feasible. There are many easily available devices that can be used to spoof cards or perform lower level replay attacks. The most popular of which is the *Proxmark3* (figure 3). It is the third iteration of the Proxmark hardware, developed by Jonathan Westhues. Since May 2007 its software and design are fully available and open source, licensed under the GPL. Consisting of a circuit board the size of a deck of cards, it is "designed to snoop, listen and emulate everything from Low Frequency (125kHz) to High Frequency (13.56MHz) tags" [28].

Other similar devices include the *Ghost* by Verdult et al. [36] and the *ChameleonMini* by the Chair for Embedded Security at the Ruhr University in Bochum, Germany [1]. Both devices are less versatile than the Proxmark3 but have similar capabilities.

In addition to card emulation, there is also the process of card cloning. Cards like those in the Mifare Classic and Mifare Ultralight families have their memory divided into blocks. Each of these blocks usually has customizable access conditions and can be writable, except for the first block. This so-called *block zero* is always read-only and contains static values like the card's UID.

This combination of the uniqueness and the read-only nature of the UID of a card is what many identification systems rely on. However, certain types of cards have been reverse engineered to such a level that special backdoored cards can be produced that allow one to modify the contents of block zero, often after supplying a special command. Previously discussed UID-based authentication systems can then be easily defeated because they cannot distinguish between an original and a cloned card.

3 ATTACKS

Using the obtained information about the system and its potential weaknesses, we can set up strategies to test and exploit these weaknesses. The strategies are divided into two categories: attacks based on card cloning and attacks based on card emulation.

3.1 CARD CLONING

As described above, specially fabricated cards are available to clone certain smart cards onto. However for at least Mifare Classic 1K, Mifare Classic 4K and Mifare Ultralight, these can easily be ordered from certain Chinese web-shops. Unfortunately it appears that no such cards exist for either Mifare DESFire or Mifare DESFire EV1, but this does not mean that UID-based authentication on these cards is secure. It is only a matter of time before these cards are reverse engineered as well.

The University Card is based on Mifare DESFire EV1 and thus it would seem that we are currently out of luck in terms of cloning a card. However there is one aspect of the University's terminals that we can use to our advantage. In addition to Mifare DESFire EV1, other types of smart cards are supported by the terminals. This means that if the system does indeed only use the UID of a card for authentication we can use a completely different type of smart card, and as long as it has the same UID as a University Card, it will internally link to the same account.

There is, however, one important difference between the UIDs of different types of cards: their length. Most cards that comply with ISO14443 either have a 4-byte UID or a 7-byte UID. In our case the Mifare DESFire EV1 has a 7-byte UID. Luckily the Mifare Ultralight also has a 7-byte UID and as described above, there are special versions of these cards available with a writable block zero. Contrary to special Mifare Classic cards and tags, these do not have a *backdoor*, i.e. they don't need to receive a special command to trigger the ability to write to block zero. This makes it possible to use ordinary hard- and software for reading and writing Mifare Ultralight cards.

A good strategy would thus be to read the UID from a University Card with an ordinary reader – such as the one contained in many modern smartphones – and use the same hardware to write this UID to a special Mifare Ultralight card. This card would then be indistinguishable from a real University Card to a reader that only looks at a card's UID.

Figure 4: The ChameleonMini

3.2 CARD EMULATION

Our other attack strategy is the use of specialized hardware to try to emulate a University Card. This technique is much more versatile than card cloning because it gives us much more control over what exactly is communicated between the (emulated) card and the terminal. Another advantage is that we can instantly and automatically change the emulated card's configuration as opposed to the card cloning technique where we need to use a separate writer device to rewrite the card's contents.

As with card cloning, it is likely that we don't need to emulate the entire University Card in all its details and contents because we only need to emulate what is communicated and compared against during authentication with the various terminals. If our suspicions about UID-based authentication as outlined in the analysis are correct, we don't even need to emulate the card's contents but only the UID and possibly some other values that are transmitted during the anti-collision (initialization) phase.

To perform such emulation attacks we need specialized hardware. Several of such devices are described above. After weighing the cost and abilities of each of these devices, the ChameleonMini (see figure 4) seems to be the best option in our case.

As its name implies, the ChameleonMini is able to emulate a range of different types of smart cards, including their UIDs and cryptographic functions [1]. It has the form-factor of a smart card and can easily be (re-)programmed via USB.

Ideally we would like to emulate a Mifare DESFire EV1 card, as this is what the University Card uses. However, the default firmware for the ChameleonMini does not have support for this card. But, it is fully re-programmable and its firmware is open source, so there is the possibility of (partly) implementing Mifare DESFire EV1 support ourselves.

Alternatively, there *is* support for the emulation of Mifare Ultralight, which – as explained in section 3.1 – is similar to the DESFire EV1 in that it has a 7 byte UID as well. Lastly, the ChameleonMini also has support for Mifare Plus, which is a type of card that can either have a 4-byte or a 7-byte UID. If (some of) the terminals do in fact only use a card's UID for authentication without also making sure it is of the correct type, these configurations could allow us to functionally "emulate" a University Card, with the same consequences as a successfully cloned card.

Additionally, because of the programmable nature and storage capacities of the ChameleonMini, we could store the UIDs of multiple cards in the ChameleonMini, allowing for quick switching between different emulated cards. We could even program it to iterate (randomly or not) over a range of UIDs, in order to find a valid account to authenticate with.

4 RESULTS

This section outlines the process of applying the above attack strategies and the results obtained from this.

4.1 CARD CLONING

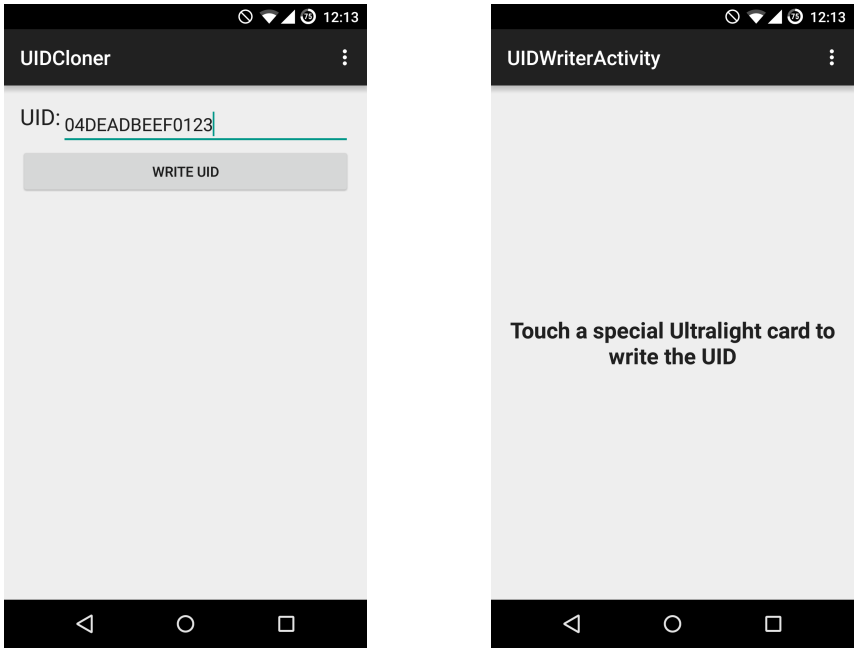
To perform experiments regarding the cloning of the UID of a University Card, a special Ultralight card with a writable block zero was used.

Because no special commands are needed to write to this card, standard smart card writing hardware and software can be used. The smart card writers that are perhaps the most ubiquitous are those found in many modern smartphones. These are especially ideal because smartphones are very mobile, allowing quick on-the-go reprogramming of the card's UID. Various apps exist to perform generic tasks on smart cards like reading, editing and writing memory dumps, using these to manually change the UID of a card is however quite cumbersome. For this reason I have written an app for Android that specializes in one thing: reading the UID of a University Card and writing it onto a special Ultralight card. It uses the standard NFC and Ultralight APIs from the Android SDK to perform these tasks.

Figure 5 shows the two screens of the app. On the first screen, the user can tap a University Card (or any supported smart card with a 7-byte UID) to the phone and the app will read its UID. The user then has the ability to tweak the UID before pressing the `Write UID` button, which launches the second screen. Once on this screen, the user taps their special Ultralight card, onto which the app will write the chosen UID before returning to the first screen. This is everything the app does and it is exactly what I needed to perform the experiments regarding card cloning.

The functionality of the clone card was tested with each of the different types of terminals (beverage, printing and parking). In each of these tests, the UID of an original University Card was taken and cloned onto the special Mifare Ultralight card (the *clone card* in the table) using the Android app described above. For every test, the response of the terminal to the clone card was compared with the response of the terminal to the original card.

Table 1 shows the results of these tests. The result of using the clone card with a FollowYou system is highlighted in bold because this is the only case where the clone card behaves exactly like the original University Card and the only case where we are successfully able to authenticate with it. This confirms our



(a) Screen 1: The start screen of the application.

(b) Screen 2: After pressing the button.

Figure 5: The *UIDCloner* Android application.

suspicion that at least one system that uses the University Card for authentication only uses its UID because the UID is the only identifying aspect that the original University Card and its clone card have in common.

Interestingly, the same is not true for the other two terminals. The response of the beverage machines is the most surprising because their built-in Inepro reader supports Mifare Ultralight cards, just like the FollowYou printers. The fact that the clone card is not recognized as a valid account can mean that the terminal compares more than just the UID during authentication. It could for example use one or more of the other values exchanged during the anti-collision phase like the SAK, ATQA and ATS values. Which, as explained in section 2.1.2, are often used for the purpose of identifying the type of a smart card. Another possibility is that authentication failed because the special Mifare Ultralight card that was used for this experiment does not fully conform to the specifications of regular Mifare Ultralight cards because of its unofficial nature.

The parking lot gates pose a slightly different problem because they don't show any response at all when provided with the clone card. Yet, when provided with a regular University Card without parking abilities they show an error message. Hence, the most likely reason for the lack of response is that the parking lot gates do not support Mifare Ultralight at all, causing them to ignore or not even detect the clone card.

As stated before, it is not possible to perfectly emulate a University Card with the default firmware of the ChameleonMini because it lacks support for the emulation of Mifare DESFire EV1. Instead, we used the ChameleonMini's ability to emulate other types of smart cards that also have a 7-byte UID. This is a similar approach to the card cloning method.

The default firmware of the ChameleonMini has support for two different card types with 7-byte UIDs: Mifare Ultralight and the 7-byte variant of Mifare Plus 1K. Both of these have been used in the tests. The performed tests are similar to those with the clone card: A regular University Card was scanned and its UID is used by the ChameleonMini as the UID of the card it emulates.

For each of the three different terminals the ChameleonMini is programmed to emulate a University Card's UID using both the Mifare Ultralight configuration and the 7-byte Mifare Plus 1K configuration. The results of these tests are shown in table 2.

Once again, the tests that resulted in successful authentication are highlighted in bold. For the attempts with the ChameleonMini's Mifare Ultralight emulation, we see similar results to those of the card cloning experiments: only the FollowYou system detects the emulated card as a valid account. This is exactly what would be expected because the card that we are emulating in these tests (the Mifare Ultralight) is of the same type as the clone card, causing the terminals to respond in the same way.

The most interesting results however are those of the tests using the ChameleonMini's Mifare Plus 1K emulation. We find the same results as with Mifare Ultralight emulation at the beverage machines and the FollowYou terminals. However, using this configuration at the parking lot gates we are suddenly able to authenticate at, and successfully drive through a parking gate.

	Beverage machines	FollowYou printing	Parking lot gates
Response to University card	Able to request the card's balance and buy drinks.	Able to log in, retrieve list of jobs, name of account-holder and their balance.	Audible beep, gate opens if a car is present.
Response to clone card	Display shows `Unknown account`.	**Able to log in, retrieve list of jobs, name of account-holder and their balance.**	Nothing happens.

Table 1: Results of the card cloning experiments

	Beverage machines	FollowYou printing	Parking lot gates
Response to University Card	Able to request the card's balance and buy drinks.	Able to log in, retrieve list of jobs, name of account-holder and their balance.	Audible beep, gate opens if a car is present.
Response to emulated Mifare Ultralight	Display shows `Unknown account`.	**Able to log in, retrieve list of jobs, name of account-holder and their balance.**	Nothing happens.
Response to emulated Mifare Plus 1K (7 byte UID)	Display shows `Unknown account`.	**Able to log in, retrieve list of jobs, name of account-holder and their balance.**	**Audible beep, gate opens if a car is present.**

Table 2: Results of the card emulation experiments with the ChameleonMini

5 EVALUATION

5.1 CONCLUSION

In our analysis of the system around the University Card we identified three main uses for the card's contactless features: the beverage machines, the FollowYou printing terminals and the parking lot gates. At all of these terminals, the card is scanned to authenticate the card-holder.

The ability to link one's own tag to one's account combined with the lack of identifying contents on the University Cards and the cobbled-together nature of the system in general led us to suspect the use of UID-based authentication in the whole system. By using both card cloning and card emulation we have proved this to be true for two of the three systems: the FollowYou terminals and the parking lot gates.

Interestingly, we were not able to authenticate with the beverage machines using either of the impersonation methods. This indicates that these machines must be reading and comparing one or more other values from the card, in addition to the UID, or alternatively, they might just be using the UID for identification, but they could be using some of the Mifare DESFire EV1's cryptographic features like AES authentication, which we are unable to emulate with the ChameleonMini and our clone cards don't support.

However, the implications of the successful authentication attempts are big: an attacker only needs a few seconds of access to a University Card to permanently be able to park for free in the account-holder's name.

5.2 SUGGESTED IMPROVEMENTS

We have demonstrated that the security of the University Card system is not very high by outlining easily replicable scenarios where an attacker can impersonate the owner of a University Card at two of the three types of terminals. This is only possible because of the way authentication is – or rather, is not – performed by these terminals, i.e, by only using the card's UID to identify a user. An obvious solution to this problem would be to use a different, more secure method of authentication.

One such a method is to use the AES capabilities of Mifare DESFire EV1 cards. For authentication the reader and the card share a secret key which they use in combination with a challenge-response protocol to produce a session key. This session key is used to perform encrypted communications. Any

uniquely identifying piece of data on the card could then be used to identify the user, including the UID [32]. With such a system in place, an attacker would need to know the secret key in addition to the card's UID to perform a successful authentication. This prevents attacks like the one explained above.

6

FUTURE WORK

For two of the three types of terminals the University Card can be used with, we showed that impersonation attacks are possible and very easy to perform. For the third type – the beverage machines – we failed to find an obvious flaw. We concluded that these machines must either be using more information than just the card's UID to identify a user, or they must be using some of the card's cryptographic features to perform authentication. They could even be performing a combination of both of these techniques.

Future work is required to determine which of these is the case and what exactly is going on with these terminals during identification and authentication.

If the first case is true, it could still be possible to perform a similar card-emulation based attack on these terminals as was done on the others. One would just need to know exactly what extra values the terminal is comparing, and emulate those accordingly. In the second case, one would be out of luck because the key that would be used for any of the encrypted communication would be secret.

One could find out which of these situations is the case by analyzing the terminal's firmware and software, but it's probably not easy to get access to these. Alternatively one could sniff communications between a University Card and the terminal to figure out what is being sent and whether it is being sent encrypted. Dedicated hardware like the Proxmark3 can be used to do this.

As for the other two terminals, improvements could be made to the demonstrated attacks. Especially the emulation-based attacks could be improved by customizing the ChameleonMini's firmware. Currently the UID of the card to be emulated is configured at run-time through the serial USB connection. This could be improved by storing one or more UIDs permanently, and allowing the user to switch between them by using the device's programmable button. One could go even further and program a range of UIDs for the ChameleonMini to iterate over automatically with a certain interval, allowing the user to *brute force* the identification procedure.

BIBLIOGRAPHY

[1] Chair for Embedded Security at the Ruhr University in Bochum, Germany. *ChameleonMini wiki on Github*. 2015. URL: https://github.com/emsec/ChameleonMini/wiki (visited on June 3, 2015).

[2] Chech Railways. *In Karta*. 2015. URL: http://www.cd.cz/en/vyhody-pro-cestujici/ (visited on March 9, 2015).

[3] Ervasti, M., Isomursu, M., and Kinnula, M. "Experiences from NFC Supported School Attendance Supervision for Children". In: *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009. UBICOMM '09.* October 2009, pp. 22–30. DOI: 10.1109/UBICOMM.2009.9.

[4] Garcia, Flavio D. et al. "Dismantling MIFARE Classic". In: *13th European Symposium on Research in Computer Security (ESORICS 2008)*. Springer Berlin/Heidelberg. 2008, pp. 97–114.

[5] HID Global. *iCLASS Smart Cards – Proximity Card Credentials – HID Global*. 2015. URL: http://www.hidglobal.com/products/Cards-and-Credentials/iCLASS (visited on March 9, 2015).

[6] Infocomm. *Specification for Contactless ePurse Application (CEPAS)*. 2015. URL: http://www.ida.gov.sg/Infocomm-Landscape/ICT-Standards-and-Framework/Specification-for-Contactless-ePurse-Application (visited on March 9, 2015).

[7] ISO. *Identification cards – Contactless integrated circuit cards – Proximity cards – Part 1: Physical characteristics*. ISO 14443-1:2008. Geneva, Switzerland: International Organization for Standardization, 2008.

[8] ISO. *Identification cards – Integrated circuit(s) cards with contacts – Part 1: Physical characteristics*. ISO 7816-1:2011. Geneva, Switzerland: International Organization for Standardization, 2011.

[9] ISO. *Identification cards – Machine readable travel documents – Part 1: Machine readable passport*. ISO 7501-1:1997. Geneva, Switzerland: International Organization for Standardization, 1997.

[10] ISO. *Identification cards – Physical characteristics*. ISO 7810:2003. Geneva, Switzerland: International Organization for Standardization, 2003.

[11] Jung, Yong-hoon, Kim, Jung-Jae, and Jun, Moon-Seog. "A Study on Authentication Protocol for Secure RFID Tag". In: *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on*. Vol. 1. November 2008, pp. 685–692. DOI: 10.1109/ICCIT.2008.409.

[12] Kerckhoffs, Auguste. "La cryptographie militaire". In: *Journal des sciences militaires* IX (January 1883), pp. 5–83. URL: http://www.petitcolas.net/fabien/kerckhoffs/.

[13] Kristien Piersma. *Koffie, thee? RUGpas mee!* 2015. URL: http://www.rug.nl/science-and-society/centre-for-information-technology/organisation/pictogram/2012-5/rugpas.pdf (visited on June 1, 2015).

[14] EZ-Link. *EZ-Link – Company Profile*. 2015. URL: http://ezlink.com.sg/about-ez-link/company-profile (visited on March 9, 2015).

[15] Merhi, M., Hernandez-Castro, J.C., and Peris-Lopez, P. "Studying the pseudo random number generator of a low-cost RFID tag". In: *RFID-Technologies and Applications (RFID-TA), 2011 IEEE International Conference on*. September 2011, pp. 381–385. DOI: 10.1109/RFID-TA.2011.6068666.

[16] Morimoto, S. "A case study of the e-money application in Japanese public transportation". In: *e-Business (ICE-B), Proceedings of the 2010 International Conference on*. July 2010, pp. 1–6.

[17] Morshed, M.M., Atkins, A., and Yu, Hongnian. "An efficient and secure authentication protocol for RFID systems". In: *Automation and Computing (ICAC), 2011 17th International Conference on*. September 2011, pp. 51–56.

[18] MPI Label Systems. *RFID for Waste Management*. 2015. URL: http://www.mpilabels.com/mpi-products/rfid-for-waste-management (visited on March 9, 2015).

[19] National Registration Department of Malaysia. *Introduction to MyKad*. 2015. URL: http://www.jpn.gov.my/en/informasi/pengenalan-kepada-mykad/ (visited on March 9, 2015).

[20] Nishihata, H. et al. "Proposed presence system for safety confirmation". In: *Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on*. October 2012, pp. 108–113. DOI: 10.1109/ICIN.2012.6376012.

[21] Nohl, Karsten et al. "Reverse-Engineering a Cryptographic RFID Tag." In: *USENIX security symposium*. Vol. 28. 2008.

[22] Nuance Communications Inc. *Datasheet Equitrac ID Card Reader Product Specs*. 2014. URL: http://www.nuance.com/ucmprod/groups/imaging/@web-enus/documents/collateral/nc_033371.pdf (visited on June 3, 2015).

[23] NXP Semiconductors. *AN10833: MIFARE Type Identification Procedure*. 2014. URL: http://www.nxp.com/documents/application_note/AN10833.pdf (visited on June 1, 2015).

[24] NXP Semiconductors Austria. *MIFARE DESFire D40*. 2015. URL: http://www.mifare.net/en/technology/security/mifare-desfire-d40/ (visited on March 10, 2015).

[25] Octopus Holdings Limited. *Octopus Cards, Hong Kong*. 2015. URL: http://www.octopus.com.hk/home/en/index.html (visited on March 9, 2015).

[26] Organization, International Civil Aviation. "Machine readable travel documents. Part 3, Part 3". In: (2008).

[27] Oswald, David and Paar, Christof. "Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World". In: *Cryptographic Hardware and Embedded Systems - CHES 2011*. Ed. by Preneel, Bart and Takagi, Tsuyoshi. Vol. 6917. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 207–222. ISBN: 978-3-642-23950-2.

[28] Proxmark Community. *Proxmark3 Github page*. 2015. URL: https://github.com/Proxmark/proxmark3 (visited on May 26, 2015).

[29] Public Transportation Victoria. *myki*. 2015. URL: http://ptv.vic.gov.au/tickets/myki (visited on March 9, 2015).

[30] San Fransisco Metropolitan Transportation Commission. *Clipper Card*. 2015. URL: https://www.clippercard.com/ClipperWeb/index.do (visited on March 10, 2015).

[31] Teply, T. and Foit, J. "Autonomous access control system". In: *Security Technology, 2008. ICCST 2008. 42nd Annual IEEE International Carnahan Conference on*. October 2008, pp. 318–320. DOI: 10.1109/CCST.2008.4751321.

[32] Texas Instruments Incorporated. *MIFARE DESFire EV1 AES Authentication With TRF7970A*. 2014. URL: http://www.ti.com/lit/an/sloa213/sloa213.pdf (visited on June 11, 2015).

[33] Trans Link Systems. *OV-Chipkaart*. 2015. URL: https://www.translink.nl/en-GB/OV-chipkaart (visited on March 9, 2015).

[34] Transport for London. *Oyster Online*. 2015. URL: https://oyster.tfl.gov.uk (visited on March 9, 2015).

[35] University of Groningen. *FAQ University Card*. 2015. URL: http://www.rug.nl/education/hoezithet/studiekiezer?tid=4_61 (visited on March 10, 2015).

[36] Verdult, R., Koning Gans, G. de, and Garcia, F.D. "A Toolbox for RFID Protocol Analysis". In: *RFID Technology (EURASIP RFID), 2012 Fourth International EURASIP Workshop on*. September 2012, pp. 27–34. DOI: 10.1109/RFID.2012.19.

[37] WPS Parking Systems. *Driving Parking technology*. 2004. URL: http://www.i2securitysolutions.com/wp-content/uploads/BCEASY.pdf (visited on June 5, 2015).

[38] Yang, Xiaobo. "Advanced public transport system in Singapore". In: *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*. Vol. 2. October 2003, 1660–1663 vol.2. DOI: 10.1109/ITSC.2003.1252765.

[39] Yoon, Donghun. "The trend and user behaviors of Japan's IC-Card system". In: *Waveform Diversity and Design Conference, 2009 International*. February 2009, pp. 183–187. DOI: 10.1109/WDDC.2009.4800341.

A

ACRONYMS

3DES Triple DES

AES Advanced Encryption Standard

ATS Answer to Select

ATQA Answer To Request type A

DES Data Encryption Standard

NFC Near Field Communication

PICC Proximity Integrated Circuit Card

RFID Radio-Frequency Identification

SAK Select Acknowledge

UID Unique Identifier