UNIVERISTY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Peeter Vahe

Tartu Smart Bike Share Access Cards Authentication Analysis

Bachelor's Thesis (9 ECTS)

Supervisor: Danielle Melissa Morgan, MSc

Tartu 2021

Tartu Smart Bike Share Access Cards Authentication Analysis

Abstract:

The aim of this bachelor's thesis is to study the cards used for unlocking the bikes in the Tartu Smart Bike Share system and how functional duplicates can be made of the aforementioned cards. The cards examined are the following: a separate ISIC card that is not bundled with a bank card, Tallinn's Public Transportation Card and Tartu's bus card. All the above-mentioned cards are MIFARE cards that use RFID technology. During the course of the work, it will be studied whether and how the aforementioned cards can be cloned into the so-called "Chinese Magic Cards" so that they can be used to open the bikes in the Tartu Smart Bike Share system using the Proxmark 3 RFID analysis tool. The data in the cards and their unique features will be explored and explained. Finally, the study will show how the cloning process works and how the cards are authenticated. Suggestions for improving the authentication process is also provided.

Keywords:
RFID, NFC, MIFARE, Proxmark 3, ACR1252U, ISIC card, Tartu bus card, Tallinn Public Transportation card, Tartu Smart Bike Share, Chinese Magic Cards

CERCS:
P170 - Computer science, numerical analysis, systems, control

Tartu Rattaringluse Ligipääsukaartide Autentimise Analüüs

Lühikokkuvõte:

Selle bakalaureusetöö eesmärk on uurida Tartu Rattaringluses rataste lahti tegemiseks kasutatavaid kaarte ning kuidas teha nendest funktsioneerivad duplikaate. Uuritavad kaardid on järgmised: eraldiseisev ehk mitte pangakaardiga ühendatud ISIC kaart, Tallinna Ühiskaart ning Tartu bussikaart. Kõik eelmainitud kaardid on MIFARE kaardid, mis kasutavad RFID tehnoloogiat. Töö käigus selgitatakse välja kas ja kuidas saab kloonida eelmainitud kaarte Chinese Magic Cards'ide peale kasutades RFID tööriista Proxmark 3, et nendega saaks Tartu Rattaringluse rattaid avada. Uuritakse ja selgitatakse kaartidel olevaid andmeid ning nende ainulaadseid omadusi. Lõpuks näitab bakalaureusetöö, kuidas toimub kloonimisprotsess ja

kaartide autentimine. Lisaks pakutakse soovitusi, kuidas Tartu Rattaringluse autentimisprotsessi parandada.

Võtmesõnad:
RFID, NFC, MIFARE, Proxmark 3, ACR1252U, ISIC kaart, Tartu bussikaart, Tallinna Ühiskaart, Tartu Rattaringlus, Chinese Magic Cards

Võtmesõnad:
P170 - Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

Introduction

In Autumn 2019, a student discovered that a user could copy the card number of a Tartu bus card with access to the Tartu Smart Bike Share program onto their own Tartu bus card or a newly purchased bus card, and it gave them access to the system as the user. This flaw has since been fixed. However, we know that the technology used by the Tallinn and Tartu cards is outdated and it is possible to copy the entire contents of these cards onto special Chinese Magic Cards. But there is no information on whether this method would work for the Tartu Smart Bike Share program.

This thesis aims to describe the background of the technologies used, find possible errors in the Tartu Smart Share Bike system, and learn/find what information is kept on the Tallinn's Public Transportation Card, Tartu bus card, and a standalone ISIC card. How the cloning process would work, what kind of obstacles there will be, and whether the clones would work on the Tartu Smart Bike Share system.

The thesis is structured as follows. The first chapter of this thesis provides an overview of the technologies used in contactless cards: RFID and NFC technologies are introduced alongside technical specifications of the MIFARE cards.

In the second chapter, the hardware used in the thesis is introduced: ACR1252U NFC reader, Chinese Magic Cards, and Proxmark 3 device. Later with the help of an Author created Python script, the Tallinn Public Transportation card, Tartu bus card and the ISIC card are presented in a human-readable manner.

The third chapter details how to clone the MIFARE cards with the Proxmark 3 device by displaying the step-by-step process behind it and show cheaper alternatives to the Proxmark 3.

The fourth chapter describes how to register a bus card to the Tartu Smart Bike Share system and use the Proxmark 3 device to sniff the communications between an NFC card reader and an NFC card. Finally, the chapter will show the testing results for the Tallinn Public Transportation card, Tartu bus card and the ISIC card and whether a full or partial clone is needed.

The fifth chapter will present bugs related to the Tartu Smart Bike Share system found during the testing.

Finally, the sixth chapter will conclude the thesis.

1 Contactless cards technology

In this bachelor's thesis, three contactless cards that unlock the Tartu Smart Bike Share program's bikes are studied:

- Tartu bus card,
- Tallinn Public Transportation card
- A standalone ISIC card not bundled with a bank card.

These cards use RFID technology to transmit information [1]. This section gives an overview of the technologies used in contactless cards.

1.1 RFID Technology

Roy Want [2] writes that RFID (Radio-frequency identification) allows remote identification using radio waves. The RFID tag or RFID chip contains a unique identification number and allows other unique sequences to be stored, such as the manufacturer's name and product type. According to the source, RFID systems can distinguish several different RFID chips in the same area of use.

Roy Want [2] points out that RFID devices are mainly divided into two categories: active and passive. Active RFID devices require a power source to operate. They are either connected to a battery or are directly in a circuit. For example, aircraft have active RFID devices that transmit their country of origin when inquired. Another example is an anti-theft device that uses a GPS with RFID to locate a car in the event of theft. Passive RFID devices operate without a power source and have an indefinite operating time. They receive the power needed from the device reader through electromagnetic waves to transmit data. Passive RFID chips are tiny and can be placed almost anywhere. This chip consists of an antenna, a semiconductor chip attached to the antenna, and a protective case that seals and protects these components. Roy Want's example of a passive RFID device is an access card that unlocks a door when shown to a reader next to that door.

RFID has specific standards with devices operating at different frequencies. Low frequency (LF) is in the range of 125-134 kHz, high frequency (HF) in the frequency 13.56 MHz, ultra-high frequency (UHF) in the frequencies of 433 MHz, 856 MHz, and 956 MHz and Microwaves in the frequencies 2.45 GHz and 2.8 GHz [4]. The cards used in the bachelor's

thesis use high frequency (HF) or 13.56 MHz, and their reading radius is capable of up to 45 cm.

1.2 NFC Technology

The following material is referenced from Roy Want's article [3]. The NFC (Near Field Communication) standard is a subset of RFID standards that operates in the 13.56 MHz (HF) frequency under the ISO 14443, ISO 18092, and FeliCa standards. It supports a maximum data exchange rate of 424 kbit per second with a range of up to 10cm. The NFC protocol supports regular data exchange between an active reader and a passive tag. In addition to that, peer-to-peer communication means that two active readers can read and transmit data to each other. NFC tags can contain read-only memory or read/write memory. Reading these kinds of tags, an active reader gets the UID (unique identifier) and, if requested the tag's data. Writing onto an NFC tag requires the client to know the security keys to each block of data the NFC tag has. The article points out that this assures that it's unlikely the tag's blocks have been tampered with.

Roy Want [3] states that the NFC Data Exchange Format (NDEF) is used to transfer NFC data between an active reader and a compatible device (passive tag or active reader). An NDEF message consists of an unlimited number of NDEF records. An NDEF record contains the length and type information, the type information can be almost anything, but there is a set of *record type definitions* (RTD) in common use. The subset of these relevant to this thesis that Roy Want wrote about in his article is as follows. *Text* type that is represented in ASCII or Unicode along with a parameter defining the language type. *Unique resource identifier* (URI) type that is encoded in a record and can be passed to an application for processing. *Signature* type that defines a format for signing a set of NDEF records, including signature algorithm and certificate types.

1.3 MIFARE Technology

MIFARE is a proprietary contactless card technology owned by NXP Semiconductors and complies with the ISO 14443 standard [5]. In this thesis, the following MIFARE cards are used: MIFARE Classic 1K and MIFARE Ultralight C. The Tallinn Public Transportation Card with

a 4-byte UID and the standalone ISIC card with a 7-byte UID use the MIFARE Classic 1K technology. The Tartu bus Card uses the MIFARE Ultralight C technology.

1.3.1 MIFARE Classic 1K

The following material is referenced from the NXP Semiconductors technical document about the MIFARE Classic 1K card [6]. The MIFARE Classic 1K card is a contactless card mainly used in public transport ticketing, access management, and various other applications with an operating distance of up to 100 mm and a typical transaction time of < 100 ms. It features 1kB of EEPROM memory organized in 16 sectors of 4 blocks, each block consisting of 16 bytes. The document points out that for security, the card uses manufacturer programmed 7-byte UID (Unique identifier) or 4-byte NUID (Non-Unique identifier), mutual three pass authentication (ISO/IEC DIS 9798-2), and an individual set of two keys per sector.

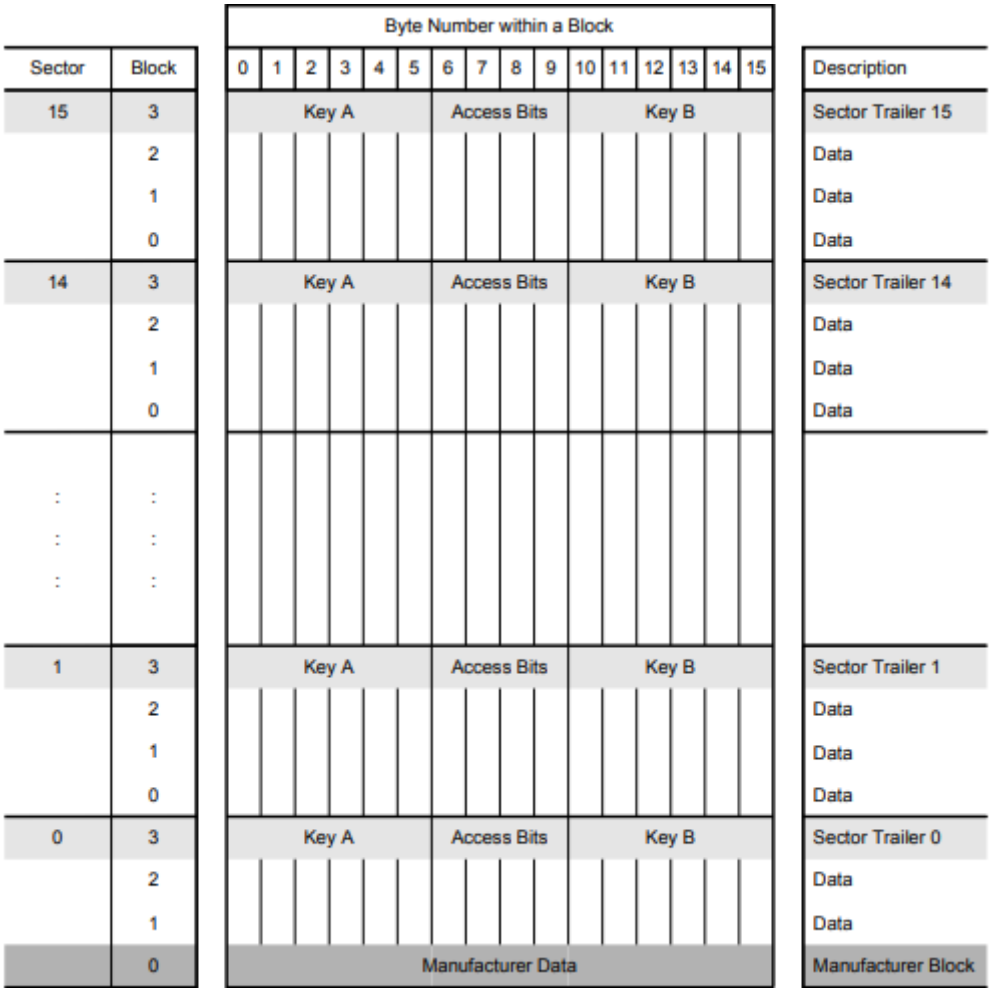

Figure 1. MIFARE Classic 1K Memory organization [6]

As Figure 1 shows, the two keys per sector are labelled A and B keys. The read and write authorization are dictated by these keys and programmed in by either the manufacturer or the system developer.

1.3.2 MIFARE Ultralight C

The following material is referenced from the NXP Semiconductors technical document about the MIFARE Ultralight C card [7]. The MIFARE Ultralight C card is a contactless card designed for limited use applications such as public transportation, event ticketing, and loyalty applications with an operating distance of up to 100mm and a swift transaction time of < 35 ms. It features a 192-byte EEPROM memory organized in 48 pages with 4 bytes each, as shown in Table 1. The document notes that, it has 3DES authentication for security, a unique 7-byte serial number (UID) for each device, a 4-byte user-programmable One Time Programmable (OTP) area, and read-only locking functions.

Table 1. MIFARE Ultralight C memory organization [7]

Page address		Byte number			
Decimal	Hex	0	1	2	3
0	00h	Serial number			
1	01h	Serial number			
2	02h	Serial number	internal	Lock bytes	Lock bytes
3	03h	OTP	OTP	OTP	OTP
4 to 39	04h to 27h	User memory	User memory	User memory	User memory
40	28h	Lock bytes	Lock bytes	-	-
41	29h	16-bit counter	16-bit counter	-	-
42	2Ah	Authentication configuration			
43	2Bh	Authentication configuration			
44 to 47	2Ch to 2Fh	Authentication key			

This thesis will mainly inspect pages 4 to 39 as these are the pages used to store user data and as such the Tartu bus card uses those pages for the application data.

2 Contactless cards hardware

In this bachelor's thesis, many different hardware components are used for reading, cloning, and writing over the card and sniffing the communication between the card and the Tartu Smart Bike reader. This section will give an overview of the hardware details, and the purpose of the hardware in relation to the thesis. It will present the data on the standalone ISIC card, Tartu bus card, and Tallinn Public Transportation card.

2.1 ACR1252U USB NFC reader

The Advanced Card Systems Ltd. ACR1252U Technical Specifications document [8] states that the ACR1252U is an NFC Forum certified NFC reader that runs on the 13.56 MHz contactless technology and supports ISO 14443 Type A and B cards, MIFARE, FeliCa, and ISO 18092-compliant NFC tags. The device has NFC card reading/writing capability, emulation, and peer-to-peer communication. It features USB 2.0 Full Speed Interface, read/write speeds up to 424 kbps, a card reading distance of up to 50 mm, and a PC/SC (Personal Computer/Smart Card) API. The document points out that the device is mainly used in e-Government, e-Banking and e-Payment, e-Healthcare, and various more areas.

Figure 2. The ACR1252U USB NFC Reader III

This thesis will use the ACR1252U USB NFC Reader III shown in Figure 2. in conjunction with Python 3, and the ACR1252U device's APDU (application protocol data unit) requests documented in the API guide [9]. The Author will create a script to read the contents of the standalone ISIC card, Tartu bus card, and Tallinn Public Transportation card.

2.2 Proxmark 3 RDV 4

The Proxmark 3 Introduction manual [10] states that the Proxmark 3 is an open-source device developed by Jonathan Westhues that enables sniffing, reading, and cloning low and high-frequency RFID tags operating at 125 kHz, 134 kHz, and 13.56 MHz. The manual points out that it can modify the UID on a MIFARE technology based Chinese Magic Card and read/write user data.

Figure 3. Proxmark 3 RDV 4.01 equipped with a high frequency antenna and its USB cable

This thesis will use the Proxmark 3 device, as shown in Figure 3, for creating the cloned cards and sniffing the commands sent between a card made to unlock the bikes at the Tartu Smart Bike Share card reader, which is in the middle of the handlebar stem of the bike. The device will aid in determining whether a full or partial clone of a card is required and if the card treats MIFARE Ultralight C and Classic 1K cards substantially differently. For example, only read a single block on the Classic 1K but use the data hashes on the Ultralight C.

In addition to the last section, the Proxmark 3 device will be used to crack MIFARE Classic 1K A and B keys needed to read and write the data blocks from the sectors. The Tallinn Public Transportation Card uses factory default keys [1] and doesn't require cracking, and the Tartu Bus Card doesn't require any authentication for read and write operations. The standalone ISIC card uses keys other than the default factory ones, so Proxmark 3's cracking function will be needed to find/crack the keys.

2.3 Chinese Magic Cards

A Lab401 article about Magic RFID cards [11] writes that when the MIFARE Classic 1K card first came out, the card was an advanced card compared to what was on the market by having an individual Unique ID, plurality of data sectors, access control lists, and keys. But the MIFARE Classic 1K cipher system and poor Pseudo-Random-Number-Generator were cracked. The first companies that started using the knowledge about the vulnerabilities of the MIFARE Classic 1K were Chinese companies, most notably FUDAN. The article writes that they began to create compatible chipsets that could forge the UID, write anywhere on the card without authentication codes and more. Since being the first to provide that sort of Magic Cards, the communities that used them started calling them Chinese Magic Cards.

Figure 4. MIFARE Classic 1K generation 1a Chinese Magic Card and Ultralight C generation 1 Chinese Magic Card

This thesis will use the Chinese Magic Cards shown in Figure 4 for the clones of the standalone ISIC card, Tartu bus card and Tallinn Public Transportation card.

The Lab401 article [11] points out that the Chinese Magic Cards used in this thesis can be detected as magic cards by an NFC card reader.

2.4 NFC cards

For the NFC cards used in this thesis, the Author has made a Python script [12] that uses the ACR1252U USB NFC Reader to send APDU (application protocol data unit) commands to the reader to read, write and authenticate the card sectors. This script will make the reading of the cards easy and extracts the necessary information to show the reader what is on the cards. In addition to the Python script, Proxmark 3 will be used to crack the MIFARE Classic 1K authentication keys.

2.4.1 Tallinn Public Transportation card

The Tallinn Public Transportation card is a MIFARE Classic 1K card [1]. The card can be bought for 2€ in most supermarkets, kiosks, and postal offices in Harjumaa and is used as a ticket for public transportation [13].

The Tallinn Public Transportation card is green in appearance with white and black text, shown in Figure 5. The relevant information on the card is on its backside where there is a bus card number along with a barcode.

Figure 5. Tallinn Public Transportation card front and back view

Using the Python script created for this thesis [12], it can be seen that the Tallinn Public Transportation card uses the first 7 sectors out of 16. It contains the card's UID, PAN (Payment card number), NDEF (NFC Data Exchange Format) records with one being a well-known Signature type and the other being an External type record "pilet.ee:ekaart:2".

```
************ CARD INFO **************

Card type: Tallinn Public Transportation Card
External type record: pilet.ee:ekaart:2
Card ATR: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6A
Card UID: 90 B3 16 BA
Card Number: 90025294061
Card PAN: 3086490090025294061
Cert: http://pilet.ee/crt/30864900-0001.crt

sector 0 key: a0 a1 a2 a3 a4 a5
sector 1 key: d3 f7 d3 f7 d3 f7
sector 2 key: d3 f7 d3 f7 d3 f7
sector 3 key: d3 f7 d3 f7 d3 f7
sector 4 key: d3 f7 d3 f7 d3 f7
sector 5 key: d3 f7 d3 f7 d3 f7
sector 6 key: d3 f7 d3 f7 d3 f7
sector 7 key: ff ff ff ff ff ff
sector 8 key: ff ff ff ff ff ff
sector 9 key: ff ff ff ff ff ff
sector 10 key: ff ff ff ff ff ff
sector 11 key: ff ff ff ff ff ff
sector 12 key: ff ff ff ff ff ff
sector 13 key: ff ff ff ff ff ff
sector 14 key: ff ff ff ff ff ff
sector 15 key: ff ff ff ff ff ff
```

Figure 6. Tallinn Public Transportation card data

Viewing the Python script output shown in Figure 6, the card's UID in hex code is 90 B3 16 BA and PAN number is 3086490090025294061, the bus card number 90025294061 is included in the PAN number which is also on the back of the Tallinn Public Transportation card. In Figure 5, the default keys that were used to authenticate the sectors are also shown. The full data dump of the card can be viewed in Appendix I, Table 2.

2.4.2 Standalone ISIC card

The standalone ISIC card is a MIFARE Classic 1K card, as stated before in this thesis. The ISIC website states [14] that the standalone ISIC card can be ordered through MinuKool for the price of 7.5€ with a validity time start of 1. August and end of the following calendar year. For a person to be eligible for the card, he/she must be at least 16 years of age and a student

studying in a higher education curriculum at a university, professional college, vocational education institution in Estonia or abroad. The website states the standalone ISIC card has a chip with the same functionality as the Tallinn Public Transportation card.

The standalone ISIC card is green and white in appearance, as shown in Figure 7, with white, dark green, and black text. On the front side, the card features an ISIC card number, the student's school name, the student's name, the student's date of birth, the student's ID number and the card's expiration date. It also comes with the portrait of the student and the logo of the school. On the backside of the card, it has the issuer, its address, signature area, RFID logo and miscellaneous info about the card.

Figure 7. The standalone ISIC card front and back view

As was stated in the 2.2 Proxmark 3 RDV 4 section, the standalone ISIC card does not use factory default keys on all its sectors, and the Proxmark 3 will be needed for the cracking of the keys. Using the Python program created for this thesis it is seen that sectors 0 to 6 use the same keys as the Tallinn Public Transportation card does, but sectors 7 to 15 have unknown keys. For the cracking of the keys, the RfidResearchGroup has a Proxmark 3 GitHub software repository with the functions needed [15].

The MIFARE Command cheat sheet [16] in the Proxmark 3 GitHub repository mentioned before, shows the command to commence the attack to crack the keys. The author will use the "staticnested" command as the Proxmark 3 device recommends it for this card. The command requires that the type of the card being attacked is known, in our case it is a Classic 1K, and a block number with its known authentication key, which is block 0 with a key of "a0a1a2a3a4a5". The author referring to the documentation in the GitHub repository used the full command of "hf mf staticnested –1k –blk 0 -a -k a0a1a2a3a4a5". As shown in Figure 8 the

cracking of the keys was successful and all of the ISIC A keys required to read the sectors are now known. The keys can be seen in Appendix II, Table 5.

```
[usb] pm3 --> hf mf staticnested --1k --blk 0 -a -k A0A1A2A3A4A5
[#] 1 static nonce 01200145
[=] RDV4 with flashmemory supported detected.
[+] Testing known keys. Sector count 16
..[=] Chunk: 4.9s | found 7/32 keys (24)
[+] Time to check 23 known keys: 5 seconds

[+] enter static nested key recovery
[+] Found 40207 key candidates
[\]     0/40207 keys | 135.2 keys/sec | worst case  297.4 seconds[#] Card didn't answer to select
[\]  4000/40207 keys | 135.1 keys/sec | worst case  268.0 seconds[#] Card didn't answer to select
[|]  5000/40207 keys | 135.0 keys/sec | worst case  260.7 seconds[#] Card didn't answer to select
[/]  6000/40207 keys | 134.9 keys/sec | worst case  253.6 seconds[#] Card didn't answer to select
[#] Card didn't answer to select
[-]  7000/40207 keys | 134.8 keys/sec | worst case  246.3 seconds[#] Card didn't answer to select
[|]  9000/40207 keys | 134.3 keys/sec | worst case  232.4 seconds[#] Card didn't answer to select
[#] Card didn't answer to select
[-] 11000/40207 keys | 129.8 keys/sec | worst case  225.0 seconds[#] Card didn't answer to select
[-] 15000/40207 keys | 130.3 keys/sec | worst case  193.5 seconds[#] Card didn't answer to select
[\] 16000/40207 keys | 130.3 keys/sec | worst case  185.7 seconds[#] Card didn't answer to select
[/] 18000/40207 keys | 130.3 keys/sec | worst case  170.5 seconds[#] Card didn't answer to select
[/] 22000/40207 keys | 130.5 keys/sec | worst case  139.5 seconds[#] Card didn't answer to select
[\] 24000/40207 keys | 130.6 keys/sec | worst case  124.1 seconds[#] Card didn't answer to select
[/] 30000/40207 keys | 130.7 keys/sec | worst case   78.1 seconds[#] Card didn't answer to select
[\] 32000/40207 keys | 130.8 keys/sec | worst case   62.8 seconds[#] Card didn't answer to select
[/] 34000/40207 keys | 130.8 keys/sec | worst case   47.5 seconds[#] Card didn't answer to select
[-] 35000/40207 keys | 130.8 keys/sec | worst case   39.8 seconds
[+] target block: 28 key type: A  -- found valid key [ 68 7A 02 EC E0 8C ]
[=] Chunk: 0.5s | found 1/32 keys (1)
[+] Found 45322 key candidates
[/]  6000/45322 keys | 131.5 keys/sec | worst case  299.1 seconds[#] Card didn't answer to select
[|]  9000/45322 keys | 131.4 keys/sec | worst case  276.4 seconds[#] Card didn't answer to select
[|] 13000/45322 keys | 129.7 keys/sec | worst case  249.2 seconds[#] Card didn't answer to select
[/] 14000/45322 keys | 129.8 keys/sec | worst case  241.2 seconds[#] Card didn't answer to select
[|] 25000/45322 keys | 130.5 keys/sec | worst case  155.7 seconds[#] Card didn't answer to select
[#] Card didn't answer to select
[/] 26000/45322 keys | 130.5 keys/sec | worst case  148.0 seconds[#] Card didn't answer to select
[\] 28000/45322 keys | 130.6 keys/sec | worst case  132.7 seconds[#] Card didn't answer to select
[/] 30000/45322 keys | 130.5 keys/sec | worst case  117.4 seconds[#] Card didn't answer to select
[-] 31000/45322 keys | 129.8 keys/sec | worst case  110.3 seconds[#] Card didn't answer to select
[/] 34000/45322 keys | 130.0 keys/sec | worst case   87.1 seconds
[+] target block: 32 key type: A  -- found valid key [ E9 B0 32 80 46 CB ]
[=] Chunk: 0.5s | found 1/32 keys (1)
[+] Found 45262 key candidates
[#] Card didn't answer to select
[-]  8000/45262 keys | 129.0 keys/sec | worst case  288.9 seconds[#] Card didn't answer to select
[/] 11000/45262 keys | 129.6 keys/sec | worst case  264.4 seconds[#] Card didn't answer to select
[-] 12000/45262 keys | 129.7 keys/sec | worst case  256.5 seconds[#] Card didn't answer to select
[\] 13000/45262 keys | 129.7 keys/sec | worst case  248.7 seconds[#] Card didn't answer to select
[|] 14000/45262 keys | 129.9 keys/sec | worst case  240.7 seconds[#] Card didn't answer to select
[/] 15000/45262 keys | 130.0 keys/sec | worst case  232.9 seconds[#] Card didn't answer to select
[#] Card didn't answer to select
[#] Card didn't answer to select
[|] 18000/45262 keys | 130.1 keys/sec | worst case  209.5 seconds[#] Card didn't answer to select
[#] Card didn't answer to select
[|] 22000/45262 keys | 130.3 keys/sec | worst case  178.6 seconds[#] Card didn't answer to select
[#] Card didn't answer to select
[|] 30000/45262 keys | 130.5 keys/sec | worst case  116.9 seconds[#] Card didn't answer to select
[/] 31000/45262 keys | 130.6 keys/sec | worst case  109.2 seconds[#] Card didn't answer to select
[/] 35000/45262 keys | 130.7 keys/sec | worst case   78.5 seconds
[+] target block: 36 key type: A  -- found valid key [ 57 DA 46 F8 10 EA ]
[=] Chunk: 0.5s | found 1/32 keys (1)
```

Figure 8. Standalone ISIC card key cracking with a nested attack

Using the author's created Python script [12], we can look at all the data on the standalone ISIC card after adding the authentication keys to the script. Comparing the card to the Tallinn Public Transportation card, the first 7 sectors, sectors 0 to 6, are the same format. This means that it

uses the same external NDEF record of “pilet.ee:kaart:2” and well-known Signature record on sectors 0 to 6.

```
************ CARD INFO **************

Card type: ISIC CARD
External type record: pilet.ee:ekaart:2
Card ATR: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6A
Card UID: 04 AF E6 42 00 00 01
Card Number: 80111827080
Card PAN: 3086490080111827080
Cert: http://pilet.ee/crt/30864900-0001.crt
ISIC number: S372000259187N
User and DoB: PEETERG VAHE 16091998
User ID: 39809160845
School: Tartu .likool
Expiration date: 31/12/2021
Keys used for authentication:

sector 0 key: a0 a1 a2 a3 a4 a5
sector 1 key: d3 f7 d3 f7 d3 f7
sector 2 key: d3 f7 d3 f7 d3 f7
sector 3 key: d3 f7 d3 f7 d3 f7
sector 4 key: d3 f7 d3 f7 d3 f7
sector 5 key: d3 f7 d3 f7 d3 f7
sector 6 key: d3 f7 d3 f7 d3 f7
sector 7 key: 68 7a 02 ec e0 8c
sector 8 key: e9 b0 32 80 46 cb
sector 9 key: 57 da 46 f8 10 ea
sector 10 key: 8c 51 16 ae 70 b6
sector 11 key: 4c 5b 7f ef 08 f2
sector 12 key: d5 5d 40 1f 9d f7
sector 13 key: 7c e0 86 02 c8 4c
sector 14 key: d9 e5 76 07 cb 4f
sector 15 key: dc fe cb 8f 7f da
```

Figure 9. Standalone ISIC card data

The standalone ISIC card uses sectors 7 to 10 to hold the information about the user that is displayed on the front of the card. Sector 7 block 28 has the ISIC card number, sector 8 block 32 has the first name of the student, block 33 the last name and block 34 the date of birth in the

format of “dd-mm-yyyy”, sector 9 block 36 has the ID number of the student and block 37 has the name of the university, sector 10 block 40 has the expiration date of the card. The data as mentioned above can be seen in Figure 7 and Figure 9. The full data dump of the card can be viewed in Appendix I, Table 3.

2.4.3 Tartu bus card

The Tartu bus card is a MIFARE Ultralight C card, as stated before in the thesis. The city of Tartu’s website states that the Tartu bus card [17] can be bought for 2€ at R-Kiosks, shopping centres, and other stores. The website states that it first has to be registered, and then the owner has to either load funds on it or buy a ticket to use it on the bus on-board validators.

The Tartu bus card is red and white in appearance with the Tartu Town Hall and the sculpture “Kissing Students” on the front, as shown in Figure 10. On the backside of the card it has instructions in white text explaining how to use it, a place to write the name of the user, a control code, and the bus card number along with a barcode. It also features RFID icons on the front and backside of the card.

Figure 10. Tartu bus card front and back view

Since the user data pages of the Tartu bus card aren’t locked, the reading of the card is trivial with the Author created Python script [12]. The card uses all its data pages. Similar to the Tallinn Public Transportation card and standalone ISIC card, it contains an external type record, PAN number and a well-known Signature record type.

```
************ CARD INFO **************

Card type: Tartu bus card
External type record: pilet.ee:ekaart:3
Card ATR: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 3A 00 00 00 00 51
Card UID: 04 61 41 EA CE 61 80
PAN: 3086490099501714963
Card Number: 99501714963
```

Figure 11. Tartu bus card data

Viewing the Python script output in Figure 11, we can see that the card is a Tartu bus card, with a UID in hex code of 04 61 41 EA CE 61 80, a PAN number of 3086490099501714963, which also contains the card number 99501714963 that is also shown on the back of the Tartu bus card as shown in Figure 10. The Tartu bus card has a different external record to the Tallinn Public Transportation card and standalone ISIC card, with it being “pilet.ee:ekaart:3” instead of “pilet.ee:ekaart:2”.

3 Cloning the NFC cards

The Proxmark 3 RDV 4 device with a GitHub repository made for the Proxmark 3 created by RfidResearchGroup [15] was used to clone the Tartu bus card, Tallinn Public Transportation card, and standalone ISIC card onto Chinese Magic Cards. This chapter will go over the exact cloning process of the MIFARE Classic 1K and MIFARE Ultralight C. It will also provide alternative ways to create clones using cheaper methods than the Proxmark 3.

3.1 Proxmark 3

The Proxmark 3 RDV 4 device is a highly versatile device that can clone the cards used in this thesis onto Chinese Magic Cards and change their UID-s. However, due to being an advanced tool, the Proxmark 3 device costs approximately 339€ [18] and can be considered too pricey by some users.

3.1.1 MIFARE Classic 1K

The prerequisite to creating a full clone from a MIFARE Classic 1K card is knowing at least one of the A or B authentication keys, having a UID changeable magic card, and a Proxmark 3 device. The commands that will be used are referenced from the MIFARE Command cheat sheet [16]. The process goes as follows:

- Create a binary key dump file from the original card

  A key cracking function will be used to create a binary key dump file. After the function has cracked all the keys on the card, it will dump them in a binary file. In this case, since all of the B keys were not known on either card, the Author used the cracking function for the ISIC card with a command of “hf mf autopwn --1k -s 0 -a -k A0A1A2A3A4A5 -f ISICakeys.dic”. For the Tallinn Public Transportation card, the Author used a command of “hf mf nested --1k --blk 0 -a -k A0A1A2A3A4A5 –dump”. The dump of the A and B keys can be seen in Appendix II, Table 5 and Table 6.

  The ISIC card command specifies that it’s a high-frequency MIFARE Classic 1K card and starts the automatic key recovery process at block 0 with an A key of A0A1A2A3A4A5 and a dictionary containing the keys that were discovered in the 2.4.2 section of this thesis. The Tallinn Public Transportation card command specifies that it’s a high-frequency

MIFARE Classic 1K card and to use a nested attack starting from block 0 using an A key of A0A1A2A3A4A5 and to dump the output to a binary file.

- Dump the data of the original card to a binary file

To create a data dump file, the cheat sheet has the command “dump”. The Author for creating the dump file for the ISIC card used the following command: “hf mf dump --1k --keys hf-mf-04AFE642000001-key-1.bin” and for the Tallinn Public Transportation card the command: “hf mf dump --1k --keys hf-mf-90B316BA-key-1.bin”.

Both of the commands specify that the card is a high-frequency MIFARE Classic 1k card and to use the following keys.

- Write the keys and data onto a magic card

To write the keys and data onto a magic card, the cheat sheet has a command named “restore”. The Author, for writing the ISIC card data onto a magic card used the following command: “hf mf restore --uid 04AFE642000001 -k hf-mf-04AFE642000001-key-1.bin”. For the Tallinn Public Transportation card “hf mf restore --uid 90B316BA -k hf-mf-90B316BA-key-1.bin”.

Both of the commands specify that the card is a high-frequency MIFARE Classic card and to restore the data with the following UID and keys to a tag.

- Set the correct UID on the magic card

The last process restores all the data blocks and authentication keys but doesn’t overwrite the UID. For that, there is a command of “csetuid”. To set the new UID on the magic card that is a clone of the ISIC card, the following command: was used “hf mf csetuid -u 04AFE642000001” and for the Tallinn Public Transportation card clone: “hf mf csetuid -u 90B316BA”.

Both commands specify to set the UID of the MIFARE Classic magic card to the one specified in the command.

We now have full working clones of both the standalone ISIC card and Tallinn Public Transportation card.

3.1.2 MIFARE Ultralight C

The prerequisites to fully clone a MIFARE Ultralight C card is a Proxmark 3 RDV 4 device and a UID changeable magic card. The commands that were used are referenced from the MIFARE Command cheat sheet [16]. The process goes as follows:

- Create a binary data dump file from the original card

To create a data dump file, the cheat sheet has the command “dump”. The Author for creating the dump file for the Tartu bus card used the following command: “hf mfu dump”. The output of the command will then show the name of the binary file that was created.

The command specifies that it is a MIFARE Ultralight card and to dump all of its pages into a binary file.

- Write the data onto the magic card

To write the data onto a magic card, the cheat sheet has a command named “restore”. The Author, for writing the Bus card data onto a magic card used the following command: “hf mfu restore -f hf-mfu-046141EACE6180-dump.bin -s”.

The command specifies that it is an MIFARE Ultralight magic card and writes the data of the specified file along with the original card’s UID onto the magic card.

- Write the 3DES key onto the magic card

To write the 3DES key of the original card to the magic card, we first must know the key. For that, the cheat sheet mentions to use the command “hf mfu info” on the original card and in its output the key will be shown, in our case the key output is the following: “3des key: 49454D4B41455242214E4143554F5946”.

The cheat sheet has a command named “setpwd” to set the 3DES key. The Author uses the following command to set the 3DES key onto the magic card: “hf mfu setpwd –key 49454D4B41455242214E4143554F5946”

The command specifies that it is a MIFARE Ultralight magic card and to write the given 3DES key onto the card.

With this we now have a full working clone of the MIFARE Ultralight C Tartu bus card.

3.2 Alternative ways to create clones

Although the Proxmark 3 is an excellent tool, some consider it too expensive. This section will introduce alternative ways to clone the cards used in this thesis at more affordable price points.

3.2.1 Author created Python script

The Author created a Python script [12] that makes it possible to also create partial clones from the cards used in this thesis with an ACR1252U, which costs approximately 38€ [19]. The script has been optimized to know what card is being used, so all the cards follow the same process in the cloning. The process goes as follows:

- Create a hex dump from the original card

  To create a hex dump, the user must run the script with the "-dump" parameter. Meaning the command would look like this: "python3 kaardiviisard.py -dump <filename>", this creates a text file that holds all the card's hex data as shown in Figure 12.

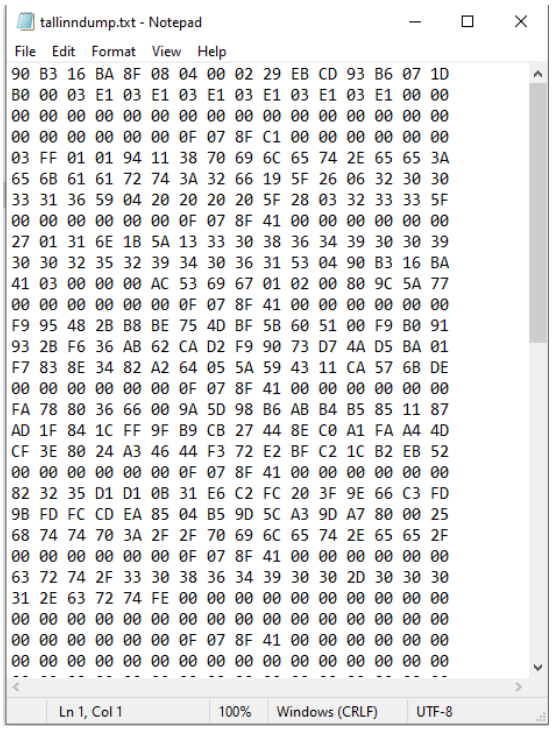

Figure 12. Tallinn Public Transportation card hex dump

- Write the hex dump onto an empty Chinese Magic Card

To write the data onto a Chinese Magic Card, the card has to be empty and the A and B authentication keys have to be the default “FF FF FF FF FF FF” for the MIFARE Classic 1K, for the MIFARE Ultralight C there are no requirements. Use the “-clone” parameter to write the dump created earlier onto a magic card: “python3 kaardiviisard.py -clone <filename>.

With this you now have created partial clone of a card. To set the correct UID onto the card and make it a full clone, it is possible to use the libnfc library with an ACR122U device [20]. The libnfc library currently supports setting the correct UID-s of Ultralight C magic cards and MIFARE Classic 1K 4-byte magic cards. This means that as this thesis is being written, it is possible to create full clones from the Tallinn Public Transportation card and Tartu bus card while the 7-byte UID ISIC card is unsupported.

3.2.2 ChameleonMini

Lab401 writes [21] that the Proxgrind ChameleonMini RevG, which costs 119€, is a powerful and portable RFID emulation multi-tool. It can emulate, read, sniff and fuzz and is compatible with MIFARE Classic 1K 4-byte and 7-byte, and Ultralight C if using the latest firmware [22]. This means that it is possible to emulate the standalone ISIC card, Tallinn Public Transportation card and Tartu bus card with this device without the need for Chinese Magic Cards. The webpage writes that the magic capabilities can be turned on and off, which is useful since some NFC card readers detect magic cards and deny access.

The Proxgrind ChameleonMini RevG is an excellent device for doing operations with MIFARE cards while being a third of the Proxmark 3 price. This device is suitable if the ACR1252U capabilities are not enough for the user, but the Proxmark 3 is too expensive.

4 Testing the cloned NFC cards

This section will first give an overview of how to register a card to a Tartu Smart Bike Share system and briefly show how the card's validity is usually checked. Secondly, it will show the Proxmark 3 sniffing outputs and whether each clone works in unlocking the bike from the bike stand.

4.1 Registering a valid NFC card to Tartu Smart Bike Share

The Tartu city website [23] details how to rent a bike from the Tartu Smart Bike Share service with a Tartu bus season ticket. To rent a bike, the user needs to have a valid 10, 30, or 90-day ticket and a bike share account that is accessible from the ratas.tartu.ee website or Tartu Smart Bike mobile application. The bus card needs to have been used at least once on a validator in a bus or at the Tartu Information Centre. In addition to that the card needs to have been personalised in the tartu.pilet.ee webpage with your ID number and the ID document number.

If all the prerequisites have been fulfilled, the user must register the bus card number on the ratas.tartu.ee webpage along with the ID number the bus card was personalised with and validate it as show in Figure 13.

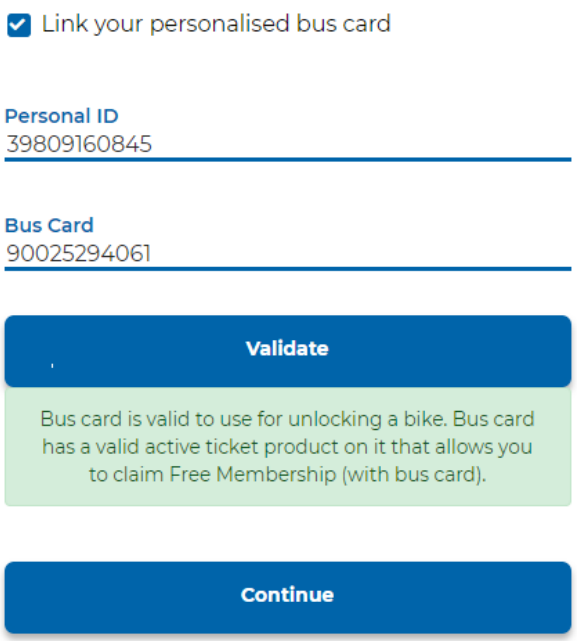

Figure 13. Registering the Author's Tallinn Public Transportation card

If everything was done correctly a green box appears that states that the ticket is valid, and the card can be now used to unlock a bike as show in Figure 13.

Martin Paljak [1] writes that the bus validators check if a card is valid by taking the Signature and “pilet.ee:ekaart:2” or “pilet.ee:ekaart:3” NDEF records that are stored on the cards and verifies the signature of that data. This makes sure that the data has not been tampered with and the data on the card is correct. Later the Author will determine if the same system for validation is in use with the Tartu Smart Bike Share service.

4.2 Sniffing the communications

To sniff the communication between an NFC card and a reader, the Proxmark 3 device ideally needs to be between the card and the reader. In this thesis, the reader is between the handlebars of the bike, so the Proxmark 3 device is placed in between the handlebars, onto the reader and the card will be placed on top of it. The Proxmark 3 command dump [24] indicates that to enable sniffing ISO 14443-a traffic, it is needed to use the command “hf 14a sniff” and to display that traffic with a command after the sniffing has been completed: “hf 14a list”.

4.3 Detecting magic cards

As Lab401’s article [11] stated, the Chinese Magic Cards used in this thesis can be identified as magic cards, usually with the query code of “0x43 / 0x40”. The Proxmark 3 device can identify magic cards by taking a function from the Proxmark 3 command dump [24], “hf search”. This function searches for high-frequency tags and displays basic information about them, including if the card is a magic card.

For testing, the Author will use four types of magic cards: MIFARE Ultralight C magic card, two types of MIFARE Classic 1K magic card, and a Lab401 magic card. All these tags are shown in Figure 14.

Figure 14. Top left is MIFARE Classic 1K magic card, top right is MIFARE Ultralight C magic card, bottom left is MIFARE Classic 1K magic tag, bottom right is Lab401 magic card

```
[usb] pm3 --> hf search
[|] Searching for ISO14443-A tag...
[+]  UID: 01 02 03 04
[+] ATQA: 00 04
[+]  SAK: 08 [2]
[+] Possible types:
[+]    MIFARE Classic 1K
[=] proprietary non iso14443-4 card found, RATS not supported
[+] Magic capabilities : Gen 1a
[#] 1 static nonce 01200145
[+] Static nonce: yes
[#] Auth error
[?] Hint: try `hf mf` commands


[+] Valid ISO14443-A tag found
```

Figure 15. The Proxmark 3 device analysing MIFARE Classic 1K magic card displayed in top left of Figure 14

The method described in the beginning of this section is shown in Figure 15, where all the cards are analysed and displayed whether they are magic cards. The Proxmark 3 detected all the cards in Figure 14 as magic cards.

4.4 Tallinn Public Transportation card

Sniffing the communication between the bike's NFC card reader and the Tallinn Public Transportation card shows that blocks 1 to 26 were read and authenticated by the NFC reader. The full sniffing data can be seen in Appendix II, Table 7. As it was discovered in section 2.4.1 of this thesis, the Tallinn Public Transportation card uses the first 7 sectors, or first 25 blocks to hold its data. This means that the system is not extracting only the card number but all the data present on the card.

Using a full clone of the Tallinn Public Transportation card on the bike's NFC card reader unlocks the bike and displays a message that states that the bike has been unlocked and to have a nice ride, as shown in Figure 16.

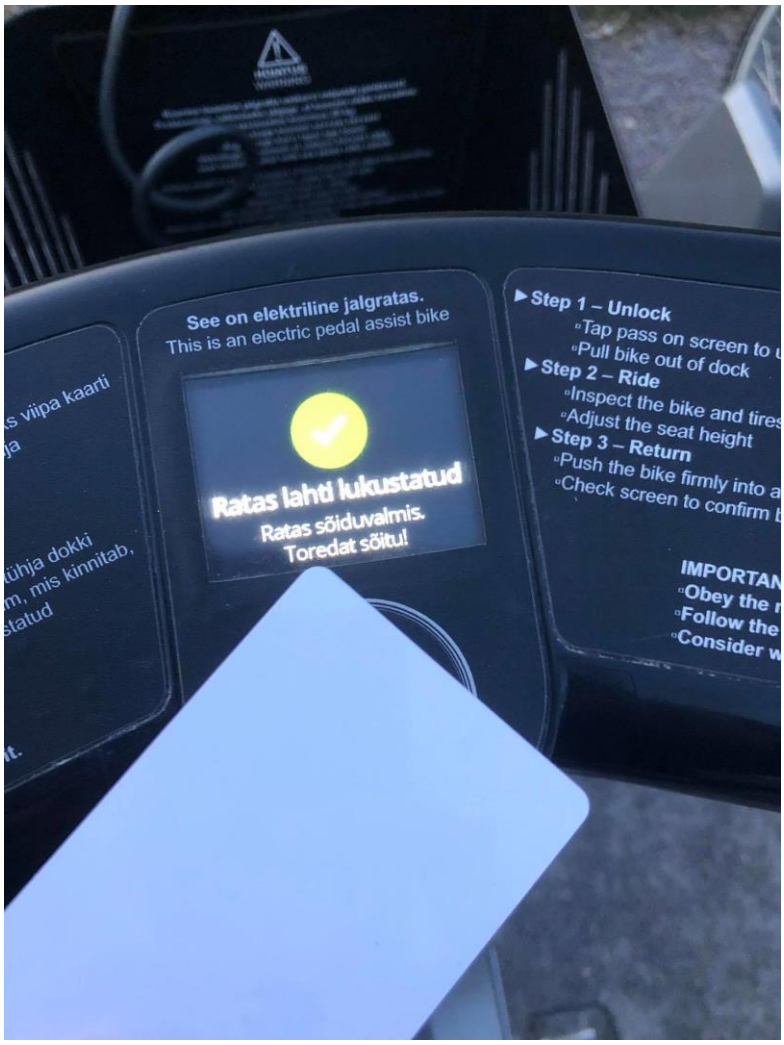

Figure 16. Using a full clone of the Tallinn Public Transportation card on a Tartu Smart Bike Share bike's NFC card reader

This means that a with a full clone it is possible to unlock a bike from the Tartu Smart Bike Share.

To test if a partial clone works, the Author uses two methods to modify the full clone:

- Changing only the UID of the card

  Changing the UID to anything other than the initial UID results in the NFC card reader displaying a message that it cannot read the card and contact customer service. This results in the bike staying locked.

- Changing only a block inside the Signature NDEF record.

  Changing block 13, which is part of the Signature NDEF record, to anything other than the original value results in the NFC card reader displaying an error message shown in Figure 17 and results in the bike staying locked.

Figure 17. Using a partial clone of the Tallinn Public Transportation card

This means that the card is using the same validation method as is described in the end of section 4.1 and a full clone is required to unlock the bikes.

Furthermore, the bike's NFC card reader was unable to detect the card as a magic card.

4.5 Standalone ISIC card

Sniffing the communication between the bike's NFC card reader and the standalone ISIC card shows that the reader reads bytes 51 to 187. This shows that only the pilet.ee:ekaart:2 external NDEF record is read, and the Signature NDEF record seems to be left out. The full sniffing data can be seen in Appendix III, Table 8.

Using a full clone of the standalone ISIC card on the NFC card reader unlocks the bike and, similarly to the Tallinn Public Transportation card, displays a message that states that the bike has been unlocked and to have a nice ride, as shown in Figure 16. This means that it is possible to unlock a bike from the Tartu Smart Bike Share with a full clone of the standalone ISIC card.

To test if a partial clone works, the Author uses three methods to modify the full clone:

- Changing only the ISIC card number

  To use an ISIC card in the Tartu Smart Bike Share system, the user needs to register the ISIC number into the system. The ISIC number is stored in block 28 on the card, as was stated in section 2.4.2 of this thesis. Changing block 28 to a value other than the original, the bike can be unlocked. This means that the ISIC card number on block 28 is not used to check the card's validity.

- Changing only the UID of the standalone ISIC card

  Changing the UID to anything other than the initial UID results in the NFC card reader displaying a message that it cannot read the card and contact customer service. This results in the bike staying locked.

- Changing only a block inside the Signature NDEF record of the standalone ISIC card

  Changing the block 13, which is part of the Signature NDEF record, to anything other than the original value results in the NFC card reader displaying an error message shown in Figure 17 and results in the bike staying locked.

Testing brought out that the ISIC number is not being used to check the validity of the card. To register an ISIC card for public transportation, the user has to enter their ID number to

tartu.pilet.ee system in the ISIC card section. This means that the ISIC number is not used there as well. The Author thinks that the card number from the ISIC card is taken, then checked from a database for a person's ID number correlating to that card number. Finally, the ID number is used to match a working public transportation card with a valid ticket.

Although the sniffing data showed that the bike's NFC card reader only reads the pilet.ee:ekaart:2 external NDEF record and not the Signature NDEF record, testing a partial clone with a modified Signature NDEF record indicates otherwise. Testing shows that both NDEF records are extracted from the card and to check the validity of the card and the card's UID. This means that the standalone ISIC card uses the same validation method as is described at the end of section 4.1, and a full clone is required to unlock the bikes.

Furthermore, the bike's NFC card reader was unable to detect the card as a magic card.

4.6 Tartu bus card

Sniffing the communication between the bike's NFC card reader and the Tartu bus card shows that the NFC card reader reads all the data pages on the card. The full sniffing data can be seen in Appendix III, Table 9. As it was written in section 2.4.3 of this thesis, all the Tartu bus card data pages are being used to hold the NDEF records. This means that the system is not extracting only the card number but all the data present on the card.

Using a full clone of the Tartu bus card on the NFC card reader unlocks the bike and, similarly to the Tallinn Public Transportation card, displays a message that states that the bike has been unlocked and to have a nice ride, as shown in Figure 16. This means that it is possible to unlock a bike from the Tartu Smart Bike Share with a full clone.

To test if a partial clone works, the Author uses two methods to modify the full clone:

- Changing only the UID of the Tartu bus card

  Changing the UID to anything other than the initial UID results in the NFC card reader displaying a message that it cannot read the card and contact customer service. This results in the bike staying locked.

- Changing only a page inside the Signature NDEF record of the Tartu bus card

Changing page 32, which is part of the Signature NDEF record, to anything other than the original value has the same results as the Tallinn Public Transportation card partial clone. The NFC card reader displays an error message shown in Figure 17 and results in the bike staying locked.

This means that the card uses the same validation method as is described at the end of section 4.1, and a full clone is required to unlock the bikes.

Furthermore, the bike's NFC card reader was unable to detect the card as a magic card.

5 Bugs found in the Tartu Smart Bike Share system

During the testing of the regular and cloned cards, the Author of this thesis encountered two bugs that allowed him to unlock more than one bike simultaneously. Usually, one account can have only one bike unlocked, and if a user tries to unlock a second bike while having one bike already unlocked, the system will display that the user has exceeded the bike limit.

The first bug requires a user to have the original card and a clone to unlock the bikes from the Tartu Smart Bike Share system. It is possible to unlock at least two bikes simultaneously by following these steps:

1. Put the original card onto the NFC card reader of a bike
2. Wait about one or half a second and put the cloned card onto the NFC reader of another bike
3. Pull both bikes at the same time
4. Both bikes are unlocked

This method is very unreliable, which worked for the Author 1 in 50 times on average. The Author displays the unlocked bikes in Figure 18.

Figure 18. Two Tartu Smart Bike Share bikes unlocked at the same time with an original and a cloned card

The second bug uses the Tartu Smart Bike Share mobile application and a card that can unlock bikes. It is possible to unlock two bikes simultaneously by following these steps:

1. Open the app on your phone and write the bike number you wish to unlock, and press unlock
2. Right after that, put the card on another bike's reader, unlocking it and disrupting the phone apps unlocking process (It should pop up in a red box saying the bike limit has been reached)
3. Put the bike you unlocked with the card back into the dock
4. Press the application's unlock button again (the number of the bike should still be there) and wait 5-6 seconds
5. After 6 seconds, put the card onto the bike's reader (not the one you are unlocking with your app)
6. The red box that appeared in step 2 does not appear, and both bikes display that they are unlocked and to undock them
7. You now have two bikes unlocked with a single membership

This method is more reliable than the first one and for the Author on average it worked about 1 in 5 times. Although a cloned card is not required to be used in this process, the Author feels that this bug should be included in this thesis.

5.1 Possible attacks

The bugs found in the Tartu Smart Bike Share system have made regular users vulnerable for attacks if an attacker has a cloned card of a regular user. The Author brings out two possible attacks:

The attacker uses either of the bugs mentioned in the last section and unlocks a bike with a clone of the regular user’s card at the exact moment the regular user does. The regular user probably does not notice that a bike is being used, and the attacker has a bike to use without the regular user suspecting anything.

The attacker uses the cloned card at any point in time. This means that if a regular user tries to unlock a bike, and the attacker is already using a bike unlocked with the regular user’s cloned card, the regular user is effectively blocked from their account. This is because the maximum number of concurrently unlocked bikes is one.

6 Conclusion

This thesis set out to test whether it is possible to clone the contents of the standalone ISIC card, Tallinn Public Transportation card and Tartu bus card onto Chinese Magic Cards, and use those cards to unlock bikes from the Tartu Smart Bike Share system. Firstly, an overview of the RFID and NFC technologies was given, followed by the technical aspects of MIFARE Classic 1K and MIFARE Ultralight C. Secondly, the hardware used in the thesis was described and what its purpose is, followed by showing the reader the contents of each card. Thirdly, the reader was given step-by-step instructions on how to make full clones of each card. Fourthly, the full and partial clones of the standalone ISIC card, Tallinn Public Transportation card, and Tartu bus card were tested on the Tartu Smart Bike Share system and determined the kind of clone needed to unlock the bikes. Finally, the bugs that were found while testing the clones were presented by step-by-step instructions.

The thesis concludes that a full clone of each card is required to unlock the bikes from the Tartu Smart Bike Share. The validation system used by the Tartu Smart Bike Share is similar to those used in buses, meaning that the data signature is checked and made sure that the data has not been tampered with, which means that a partial clone is not sufficient.

To improve the Tartu Smart Bike Share system, the Author advises to use a queue for handling requests and explicitly denying magic cards access to the system.

References

[1] M. Paljak. Tallinna ühiskaardi miniuuring. 2015. https://martinpaljak.github.io/yhiskaart/ (7.12.2020)

[2] R. Want. An introduction to RFID technology. *IEEE Pervasive Computing*, vol 5, no. 1. 2006. https://ieeexplore.ieee.org/document/1593568 (06.04.2021)

[3] R. Want. Near field communication. *IEEE Pervasive Computing*, vol 10, no. 3. 2011. https://ieeexplore.ieee.org/document/5958681 (06.04.2021)

[4] S. Tedijni, E. Perret. Radio-Frequency Identification Systems and Advances in Tag Design. *URSI Radio Science Bulletin*, vol. 2009, no. 331. 2009. https://ieeexplore.ieee.org/document/7909297 (06.04.2021)

[5] MIFARE – The Brand of Contactless IC Products. https://www.mifare.net/en/about-mifare/ (06.04.2021)

[6] NXP Semiconductors. MIFARE Classic EV1 1K – Mainstream contactless smart card IC for fast and easy solution development. 2018. https://www.nxp.com/docs/en/data-sheet/MF1S50YYX_V1.pdf (06.04.2021)

[7] NXP Semiconductors. MIFARE Ultralight C – Contactless ticket IC. 2019. https://www.nxp.com/docs/en/data-sheet/MF0ICU2.pdf (06.04.2021)

[8] Advanced Card Systems Ltd. ACR1252U USB NFC Forum Certified Reader Technical Specifications V1.06. https://www.acs.com.hk/download-manual/6401/TSP-ACR1252U-1.06.pdf (11.04.2021)

[9] Advanced Card Systems Ltd. ACR1252U USB NFC Forum Certified Reader Application Programming Interface V1.16. https://www.acs.com.hk/download-manual/6402/API-ACR1252U-1.16.pdf (19.04.21)

[10] Wilson. Proxmark III User Guide. 2015. https://lab401.com/docs/Proxmark%20III%20User%20Guide.pdf (13.04.2021)

[11] Know your magic cards. https://lab401.com/blogs/academy/know-your-magic-cards (23.04.2021)

[12] P. Vahe. kaardiviisard. https://github.com/GUCCIFER/kaardiviisard (19.04.2021)

[13] Pilet.ee info about the Ühiskaart. https://pilet.ee/viipe/uhiskaart/persod/perso/?lang=ee (18.04.2021)

[14] ISIC Student ticket as a regular card. https://www.isic.ee/cards/isic-uliopilaspilet/ (19.04.21)

[15] RfidResearchGroup. proxmark3. https://github.com/RfidResearchGroup/proxmark3 (19.04.2021)

[16] RfidResearchGroup. MIFARE Command cheat sheet. https://github.com/RfidResearchGroup/proxmark3/blob/master/doc/cheatsheet.md#MIFARE (20.04.2021)

[17] Tartu bus card information. https://www.tartu.ee/en/tartu-bus-card (20.04.2021)

[18] Lab401. Proxmark 3 RDV4.01. https://lab401.com/products/proxmark-3-rdv4 (03.05.2021)

[19] Smartcard Focus. ACR1252U. https://www.smartcardfocus.com/shop/ilp/id~841/acr1252u/p/index.shtml (03.05.2021)

[20] nfc-tools. libnfc. https://github.com/nfc-tools/libnfc (03.05.2021)

[21] Lab401. Proxgrind ChameleonMini RevG. https://lab401.com/products/proxgrind-chameleon-mini-revg (03.05.2021)

[22] RfidResearchGroup. ChameleonMini. https://github.com/RfidResearchGroup/ChameleonMini (03.05.2021)

[23] Tartu Smart Bike Share. https://tartu.ee/en/bikeshare (21.04.2021)

[24] RfidResearchGroup. Proxmark3 command dump. https://github.com/RfidResearchGroup/proxmark3/blob/master/doc/commands.md (24.04.2021)

Appendix

I. Memory dumps of MIFARE cards

Table 2. Memory dump of the Tallinn Public Transportation card

Block	Data Bytes	ASCII
Sector 0		
Block 00	90 B3 16 BA 8F 08 04 00 02 29 EB CD 93 B6 07 1D	.........).....
Block 01	B0 00 03 E1 03 E1 03 E1 03 E1 03 E1 03 E1 00 00	.....
Block 02	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 03	00 00 00 00 00 00 0F 07 8F C1 00 00 00 00 00 00	.....
Sector 1		
Block 04	03 FF 01 01 94 11 38 70 69 6C 65 74 2E 65 65 3A	......8pilet.ee:
Block 05	65 6B 61 61 72 74 3A 32 66 19 5F 26 06 32 30 30	ekaart:2f._&.200
Block 06	33 31 36 59 04 20 20 20 20 5F 28 03 32 33 33 5F	316Y....._(.233_
Block 07	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.........A.....
Sector 2		
Block 08	27 01 31 6E 1B 5A 13 33 30 38 36 34 39 30 30 39	'.1n.Z.308649009
Block 09	30 30 32 35 32 39 34 30 36 31 53 04 90 B3 16 BA	0025294061S.....
Block 10	41 03 00 00 00 AC 53 69 67 01 02 00 80 9C 5A 77	A.....Sig.....Zw
Block 11	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.........A.....
Sector 3		
Block 12	F9 95 48 2B B8 BE 75 4D BF 5B 60 51 00 F9 B0 91	..H+..uM.[`Q....
Block 13	93 2B F6 36 AB 62 CA D2 F9 90 73 D7 4A D5 BA 01	.+.6.b....s.J...
Block 14	F7 83 8E 34 82 A2 64 05 5A 59 43 11 CA 57 6B DE	...4..d.ZYC..Wk.
Block 15	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.........A.....
Sector 4		
Block 16	FA 78 80 36 66 00 9A 5D 98 B6 AB B4 B5 85 11 87	.x.6f..]........
Block 17	AD 1F 84 1C FF 9F B9 CB 27 44 8E C0 A1 FA A4 4D	........'D.....M

Block 18	CF 3E 80 24 A3 46 44 F3 72 E2 BF C2 1C B2 EB 52	.>.$.FD.r.....R
Block 19	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.........A.....
Sector 5		
Block 20	82 32 35 D1 D1 0B 31 E6 C2 FC 20 3F 9E 66 C3 FD	.25...1....?.f..
Block 21	9B FD FC CD EA 85 04 B5 9D 5C A3 9D A7 80 00 25	.........\.....%
Block 22	68 74 74 70 3A 2F 2F 70 69 6C 65 74 2E 65 65 2F	http://pilet.ee/
Block 23	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.........A.....
Sector 6		
Block 24	63 72 74 2F 33 30 38 36 34 39 30 30 2D 30 30 30	crt/30864900-000
Block 25	31 2E 63 72 74 FE 00 00 00 00 00 00 00 00 00 00	1.crt..........
Block 26	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 27	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.........A.....
Sector 7		
Block 28	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 29	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 30	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 31	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF	.........i.....
Sector 8		
Block 32	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 33	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 34	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 35	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF	.........i.....
Sector 9		
Block 36	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 37	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 38	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 39	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF	.........i.....
Sector 10		

Block 40	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 41	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 42	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 43	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF	.....i.....
Sector 11		
Block 44	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 45	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 46	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 47	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF	.....i.....
Sector 12		
Block 48	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 49	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 50	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 51	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF	.....i.....
Sector 13		
Block 52	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 53	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 54	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 55	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF	.....i.....
Sector 14		
Block 56	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 57	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 58	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 59	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF	.....i.....
Sector 15		
Block 60	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 61	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 62	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....

Block 63	00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF	.....i.....

Table 3. Memory dump of the standalone ISIC card

Block	**Data Bytes**	**ASCII**
Sector 0		
Block 00	04 AF E6 42 00 00 01 88 44 00 C2 00 00 00 00 00	...B....D.....
Block 01	B0 00 03 E1 03 E1 03 E1 03 E1 03 E1 03 E1 00 00	.....
Block 02	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 03	00 00 00 00 00 00 0F 07 8F C1 00 00 00 00 00 00	.....
Sector 1		
Block 04	03 FF 01 04 94 11 3B 70 69 6C 65 74 2E 65 65 3A	......;pilet.ee:
Block 05	65 6B 61 61 72 74 3A 32 66 19 5F 26 06 32 31 30	ekaart:2f._&.210
Block 06	32 30 38 59 04 20 20 20 20 5F 28 03 32 33 33 5F	208Y....._(.233_
Block 07	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.........A.....
Sector 2		
Block 08	27 01 31 6E 1E 5A 13 33 30 38 36 34 39 30 30 38	'.1n.Z.308649008
Block 09	30 31 31 31 38 32 37 30 38 30 53 07 04 AF E6 42	0111827080S....B
Block 10	00 00 01 41 03 00 00 00 AC 53 69 67 01 02 00 80	...A.....Sig....
Block 11	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.........A.....
Sector 3		
Block 12	55 30 C3 55 0F DB F7 A7 9B F1 CC 1B AC 2D 03 F8	U0.U.........-..
Block 13	3E 08 CE 54 67 2C 9B 0A D7 F6 1D F7 41 A2 70 91	>..Tg,......A.p.
Block 14	61 21 B0 58 98 C8 95 8A FF B9 30 F5 F7 9B 10 7F	a!.X......0.....
Block 15	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.........A.....
Sector 4		
Block 16	EB DD 32 09 BC 7F 57 74 E1 EC DD BB 01 AA 66 52	..2...Wt......fR
Block 17	7D C5 8C FA 8D BF 9E EE 10 C4 61 C5 58 DD 59 A2	}.........a.X.Y.
Block 18	01 B6 2F 14 C2 56 7C 93 8D A8 26 93 2E 20 5D 74	../..V|...&...]t

Block 19	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.....A.....
Sector 5		
Block 20	F4 67 2D 6B AE 2E E9 35 72 2E 77 E8 1A 90 D0 C5	.g-k...5r.w.....
Block 21	63 79 6A 18 53 BC DD 4B C9 DE E3 FB 72 A3 C3 F8	cyj.S..K....r...
Block 22	80 00 25 68 74 74 70 3A 2F 2F 70 69 6C 65 74 2E	..%http://pilet.
Block 23	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.....A.....
Sector 6		
Block 24	65 65 2F 63 72 74 2F 33 30 38 36 34 39 30 30 2D	ee/crt/30864900-
Block 25	30 30 30 31 2E 63 72 74 FE 00 00 00 00 00 00 00	0001.crt........
Block 26	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 27	00 00 00 00 00 00 0F 07 8F 41 00 00 00 00 00 00	.....A.....
Sector 7		
Block 28	53 33 37 32 30 30 30 32 35 39 31 38 37 4E 00 00	S372000259187N..
Block 29	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 30	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 31	00 00 00 00 00 00 0F 07 8F 00 00 00 00 00 00 00	....
Sector 8		
Block 32	50 45 45 54 45 52 47 00 00 00 00 00 00 00 00 00	PEETERG.........
Block 33	56 41 48 45 04 00 00 00 00 00 00 00 00 00 00 00	VAHE............
Block 34	31 36 30 39 31 39 39 38 00 00 00 00 00 00 00 00	16091998........
Block 35	00 00 00 00 00 00 0F 07 8F 00 00 00 00 00 00 00	....
Sector 9		
Block 36	33 39 38 30 39 31 36 30 38 34 35 00 00 00 00 00	39809160845.....
Block 37	54 61 72 74 75 20 DC 6C 69 6B 6F 6F 6C 00 00 00	Tartu..likool...
Block 38	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
Block 39	00 00 00 00 00 00 0F 07 8F 00 00 00 00 00 00 00	....
Sector 10		
Block 40	33 31 2F 31 32 2F 32 30 32 31 00 00 00 00 00 00	31/12/2021......

Block 41	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 42	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 43	00 00 00 00 00 00 0F 07 8F 00 00 00 00 00 00 00	.....
Sector 11		
Block 44	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 45	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 46	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 47	00 00 00 00 00 00 0F 07 8F 00 00 00 00 00 00 00	.....
Sector 12		
Block 48	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 49	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 50	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 51	00 00 00 00 00 00 0F 07 8F 00 00 00 00 00 00 00	.....
Sector 13		
Block 52	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 53	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 54	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 55	00 00 00 00 00 00 0F 07 8F 00 00 00 00 00 00 00	.....
Sector 14		
Block 56	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 57	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 58	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 59	00 00 00 00 00 00 0F 07 8F 00 00 00 00 00 00 00	.....
Sector 15		
Block 60	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 61	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 62	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
Block 63	00 00 00 00 00 00 0F 07 8F 00 00 00 00 00 00 00	.....

Table 4. Memory dump of the Tartu bus card

Pages	Data Bytes	ASCII
Page 04-07	03 8A 94 11 35 70 69 6C 65 74 2E 65 65 3A 65 6B	....5pilet.ee:ek
Pages 08-11	61 61 72 74 3A 33 66 0F 5F 26 06 31 39 31 30 32	aart:3f._&.19102
Pages 12-15	32 59 04 20 20 20 20 6E 22 5A 13 33 30 38 36 34	2Y.....n"Z.30864
Page 16-19	39 30 30 39 39 35 30 31 37 31 34 39 36 33 53 07	90099501714963S.
Pages 20-23	04 61 41 EA CE 61 80 54 02 00 01 51 03 3B 53 69	.aA..a.T...Q.;Si
Pages 24-27	67 01 04 00 36 30 34 02 18 05 40 7A 5C 98 1C BE	g...604...@z\...
Pages 28-31	6D DB F1 1A 1B CE F3 A7 9E 70 57 BD 73 32 6C D5	m........pW.s2l.
Pages 32-35	68 02 18 55 C1 19 2A 40 3C A5 F4 5F B3 AE 65 EF	h..U..*@<.._..e.
Pages 36-39	BD F7 3E 50 72 F2 6D 0C 18 25 6F 00 00 00 00 00	..>Pr.m..%o.....

II. MIFARE Authentication keys

Table 5. Standalone ISIC card A and B authentication keys

Sector	Key A	Key B	Sector	Key A	Key B
0	A0A1A2A3A4A5	916A7E8E8164	**1**	D3F7D3F7D3F7	076EA79721C2
2	D3F7D3F7D3F7	6234452BB66E	**3**	D3F7D3F7D3F7	F37DE4199E86
4	D3F7D3F7D3F7	9341B8F62AB4	**5**	D3F7D3F7D3F7	5FC2C76184DF
6	D3F7D3F7D3F7	D7E19BC535F8	**7**	687A02ECE08C	D3015704D9B3
8	E9B0328046CB	192A2F866DD3	**9**	57DA46F810EA	494562A17C5D
10	8C5116AE70B6	026B6458CEA4	**11**	4C5B7FEF08F2	B3976B16B504
12	D55D401F9DF7	B3B09E735837	**13**	7CE08602C84C	C773830D452A
14	D9E57607CB4F	9283D301F7FD	**15**	DCFECB8F7FDA	8946E6B902E0

Table 6. Tallinn Public Transportation card A and B authentication keys

Sector	Key A	Key B	Sector	Key A	Key B
0	A0A1A2A3A4A5	83275168F885	**1**	D3F7D3F7D3F7	2703A1BACC5B
2	D3F7D3F7D3F7	123DDFDDD897	**3**	D3F7D3F7D3F7	1838536C0626
4	D3F7D3F7D3F7	EEC951EF32EE	**5**	D3F7D3F7D3F7	D7648552806A
6	D3F7D3F7D3F7	0B457CE65C1E	**7**	FFFFFFFFFFFF	FFFFFFFFFFFF
8	FFFFFFFFFFFF	FFFFFFFFFFFF	**9**	FFFFFFFFFFFF	FFFFFFFFFFFF
10	FFFFFFFFFFFF	FFFFFFFFFFFF	**11**	FFFFFFFFFFFF	FFFFFFFFFFFF
12	FFFFFFFFFFFF	FFFFFFFFFFFF	**13**	FFFFFFFFFFFF	FFFFFFFFFFFF
14	FFFFFFFFFFFF	FFFFFFFFFFFF	**15**	FFFFFFFFFFFF	FFFFFFFFFFFF

III. Proxmark 3 sniffing output

Table 7. Tallinn Public Transportation card sniffing output

Source	Data (! denotes parity error)	CRC	Annotation
Rdr	52(7)		WUPA
Rdr	f9 7f		
Rdr	93 20		ANTICOLL
Rdr	27! 04		
Rdr	93 70 90 b3 16 ba 8f 75 4b	ok	SELECT_UID
Rdr	f3 00!		
Rdr	00(1)		
Rdr	60 04 d1 3d	ok	AUTH-A(4)
Rdr	03(3)		
Rdr	02(4)		
Rdr	01(1)		
Rdr	30! 7e d3 0c! 41 ed! 7a 96!	!crc	READBLOCK(126)
Rdr	03! 03!		
Rdr	01(3)		
Rdr	00(1)		
Rdr	57 df! 5f 91	!crc	
Rdr	07(5)		
Rdr	9f! 0c 00!	!crc	
Rdr	46 00!		
Rdr	0f(5)		
Rdr	80 c0! 32! cb	!crc	
Rdr	01(2)		
Rdr	f3! e1! f7! 84!	!crc	
Rdr	03(4)		
Rdr	0f(5)		
Rdr	00(1)		
Rdr	0c(5)		
Rdr	01(2)		
Rdr	26(7)		REQA
Rdr	0e		
Rdr	01! 01		
Rdr	91 bd 06 45!	!crc	
Rdr	07(4)		
Rdr	00(2)		
Rdr	2c! 67 d8! 62 4c 28! 1b ea!	!crc	
Rdr	32! 00!		
Rdr	fa c8! 23 9a	!crc	
Rdr	06(6)		
Rdr	01(3)		
Rdr	04(6)		
Rdr	c8 00!		
Rdr	32		

Rdr	03(4)		
Rdr	90 00!		
Rdr	5a 87! 54! c7	!crc	
Rdr	00(1)		
Rdr	43 99 4c! f8	!crc	MAGIC WUPC2
Rdr	00(4)		
Rdr	08(6)		
Rdr	0f(5)		
Rdr	11 c4 00!	!crc	
Rdr	00(2)		
Rdr	01! 20 9e! 01	!crc	
Rdr	18 c8! 55! da!	!crc	
Rdr	04(4)		
Rdr	63! 01		
Rdr	13 e4 96! 20! 17 44! f0! d8!	!crc	
Rdr	8c! 04		
Rdr	e0 00!		RATS
Rdr	c7 f5! 37! 2d!	!crc	
Rdr	ce! 23 63!	!crc	
Rdr	01(3)		
Rdr	84!		
Rdr	72! 49		
Rdr	71!		
Rdr	3f(6)		
Rdr	5b! f3! 97 d4!	!crc	
Rdr	b6 ff b1 70	!crc	
Rdr	00(2)		
Rdr	00(2)		
Rdr	00(1)		
Rdr	87! 92 38! 00!	!crc	
Rdr	12(7)		
Rdr	03(3)		
Rdr	33! 01		
Rdr	b0 91 72 17!	!crc	TRANSFER(145)
Rdr	00(1)		
Rdr	00(2)		
Rdr	02(4)		
Rdr	60!		EV1 VERSION
Rdr	09! 49 4f! fc 9b 1d 71 a5!	!crc	
Rdr	01(2)		
Rdr	7e!		
Rdr	dd! 4b! 5b e4	!crc	
Rdr	07(4)		
Rdr	00(1)		
Rdr	01(3)		
Rdr	31(7)		
Rdr	e4 c9! 00!	!crc	
Rdr	06(4)		

Rdr	00(1)		
Rdr	00(2)		
Rdr	03(4)		
Rdr	30! 32 86! 03!	!crc	READBLOCK(50)
Rdr	f4! d3 03 f5	!crc	
Rdr	f4 83 62! a2	!crc	
Rdr	02(3)		
Rdr	e7! 0e		
Rdr	e2 e3 00!	!crc	
Rdr	01(2)		
Rdr	ff 01		
Rdr	00(1)		
Rdr	e3 00!		
Rdr	c8! 00!		
Rdr	f1 df cd! 5f!	!crc	
Rdr	01(5)		
Rdr	02(4)		
Rdr	b7! d7 3a 2c! 75 c9! eb! e5!	!crc	
Rdr	00(1)		
Rdr	01(2)		
Rdr	6a! f5! 12! b2!	!crc	
Rdr	03(3)		
Rdr	00(1)		
Rdr	02 e3! 08	!crc	
Rdr	07(4)		
Rdr	07(4)		
Rdr	a7! bb 31! 0b!	!crc	
Rdr	07(4)		

Table 8. ISIC card sniffing output

Source	**Data (! denotes parity error)**	**CRC**	**Annotation**
Rdr	52(7)		WUPA
Tag	44 00		
Rdr	93 20		ANTICOLL
Tag	88 04 af e6 c5		
Rdr	93 70 88 04 af e6 c5 fd 12	ok	SELECT_UID
Tag	04 da 17		
Rdr	95 20		ANTICOLL-2
Tag	42 00 00 01 43		
Rdr	95 70 42 00 00 01 43 bc 3f	ok	SELECT_UID-2
Tag	08 b6 dd		
Rdr	60 04 d1 3d	ok	AUTH-A(4)
Tag	01 20 01 45		
Rdr	f7! 12! 03 e4! a5! 1c! 23 e3	!crc	

Tag	bb 1d! ce! 03		
Rdr	3a! 33 bb 01	!crc	READ RANGE (51-187)
Tag	18! 54 7c 3e! 4f 27! 74! ea! 2e 0e 2a 05! 31! 2b! ca! 13! e1! d2!	!crc	
Rdr	98! a1 be 9b	!crc	
Tag	5c 07 cb! f6 85! d4! c7! 66 23! e5! 48! 75! 81! 9b 03! 46! d4! 45	!crc	
Rdr	20! 1c! 46 d5	!crc	
Tag	84 b3 6d a9 d4 4d b3! 75! 97 e3! 6c! 86 d9! ac 25 16 9f f5!	!crc	
Rdr	5b! c0 28 bf!	!crc	
Tag	a8 6a! 98 76		
Rdr	01! 92! 56 79! a8! ea 10 72!	!crc	
Tag	a8! 1c! c0 bc!		
Rdr	d9 e1 45! 45	!crc	
Tag	66 b9 d9! ef! 95! b1 dd! 19! 2e! 23! 91! 6e! 62 8d! 6f 42 ae! 01!	!crc	
Rdr	b3! 21 b5 e2!	!crc	
Tag	71 71! eb! a5! 1c 16! 42 1b! 45! 51! f9 1f! 21 4c! 14 dd! f8 24!	!crc	
Rdr	9c! ff b4! 2d	!crc	
Tag	cd 65 a7 b1! 00 ec e7! d3 8d! 92 7a! ad! 4c! b5! a5! 1d! f4! 1e	!crc	
Rdr	9d! 4c! 93! d4!	!crc	
Tag	e5 05 3f f7		
Rdr	6a 86! 17 e8 c0 3f c1! 98	!crc	
Tag	e1 6b 78 36		
Rdr	59 40! a8! 79	!crc	
Tag	19! d6 98! 95! e8! 28! fa! 71 d2! 71 6f dd! 53 b9! 85 25! 9c 6b	!crc	
Rdr	6f 4b! bb 9d!	!crc	
Tag	c7 ce 30 9d! a4! 4b c0! ae 74 b6 62! c7! 8e ea d4! 7b c2! 68!	!crc	
Rdr	6e! ae! 97 83!	!crc	
Tag	53! 07! 82! 6e! 13 8e! e3 07 d3! bf! b9! 5f 8b a8! cf! 31! 1e! 37	!crc	
Rdr	21! 85 6e ce!	!crc	
Tag	ee ac! d1 2a		
Rdr	b8! 20! 20 27 39! ea! 5a c0!	!crc	
Tag	15 87 95! 32!		
Rdr	bb! 5d! 48! 9c	!crc	
Tag	a3 47 ec 03 26 07 61! a3! 12 ae 19! 69 f7 a6! e1! ad! a7 2a	!crc	
Rdr	42 1b! f3 b5!	!crc	
Tag	7d! 3c 8a ea f6! c6 83 c0! 8d 35 ec! 1c! 68! 40! 31 11 17 ad	!crc	
Rdr	7e! cb de a4	!crc	

Tag	eb 1f 7b! d4! 2d 69! 77 b8! 7c 97! ad 71 fa b6 27 a3! a9! c8	!crc	
Rdr	7f c3 fa! 54	!crc	
Tag	08 40 40 c6		
Rdr	20 e4! a5 1c ac e3 02! 16!	!crc	
Tag	d6 ff! cd 20		
Rdr	87! b1 79! 09!	!crc	
Tag	98 d7 98 46 86! 11 a1 a9! 09! 65! 22! 31 75! fb! 8a 1f! 47 36	!crc	
Rdr	63 3d 98! 23!	!crc	
Tag	ee e9! f1 5d 8e! fd 3d 68 e4! 5b 42! dd 69! 08 a4 d3 39! 93!	!crc	
Rdr	52(7)		WUPA

Table 9. Tartu bus card sniffing output

Source	Data (! denotes parity error)	CRC	Annotation
Rdr	00(2)		WUPA
Tag	3f(5)		
Tag	93 20		
Rdr	00(3)		ANTICOLL
Tag	93 70 88 04 61 41 ac 4c 14		
Rdr	00(2)	ok	SELECT_UID
Tag	95 20		
Rdr	00(2)		ANTICOLL-2
Tag	95 70 ea ce 61 80 c5 78 31		
Rdr	00!	ok	SELECT_UID-2
Tag	30 04 26 ee		
Rdr	03!	ok	READBLOCK(4)
Tag	01(0)		
Tag	01(0)		
Tag	01(0)		
Tag	07(2)		
Tag	30 08 4a 24		
Rdr	01(5)	ok	READBLOCK(8)
Tag	00(1)		
Tag	00(2)		
Tag	01(0)		
Tag	30 0c 6e 62		
Rdr	00(1)	ok	READBLOCK(12)
Tag	0f(3)		
Tag	30 10 83 b8		
Rdr	01(3)	ok	READBLOCK(16)
Tag	00(6)		
Tag	00(1)		
Tag	30 14 a7 fe		

Rdr	00(2)	ok	READBLOCK(20)
Tag	00(2)		
Tag	01(0)		
Tag	01(0)		
Tag	00(1)		
Tag	1f(4)		
Tag	30 18 cb 34		
Rdr	07(5)	ok	READBLOCK(24)
Tag	07(2)		
Tag	30 1c ef 72		
Rdr	01(2)	ok	READBLOCK(28)
Tag	07(2)		
Tag	00(2)		
Tag	01(0)		
Tag	00(2)		
Tag	30 20 00 89		
Rdr	00(3)	ok	READBLOCK(32)
Tag	03(1)		
Tag	0f(3)		
Tag	7f(6)		
Tag	01(0)		
Tag	00(2)		
Tag	03(3)		
Tag	30 24 24 cf		
Rdr	01(0)	ok	READBLOCK(36)
Tag	00(2)		
Tag	30 28 48 05		
Rdr	52(7)	ok	READBLOCK(40)
Rdr	00(2)		WUPA

IV. License

Non-exclusive licence to reproduce thesis and make thesis public

I, Peeter Vahe,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,
   Tartu Smart Bike Share Access Cards Authentication Analysis
   Supervised by Danielle Melissa Morgan

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Peeter Vahe
04/05/2021