NFC/RFID token security.
Case study: Estonian public transportation cards

Yauhen Yakimenka
Supervised by Jan Willemson

December 17, 2016

Abstract

This report, prepared for the course *Research Seminar in Cryptography*, talks about security of NFC/RFID cards. It first describes the most widely-used type of cards, MIFARE Classic, and then considers a real-life application, namely Estonian public transportation cards. The communication between a real card reader installed in Tartu bus and a Tallinn public transportation card is eavesdropped and analysed on high level.

1 Introduction

Radio frequency identification (RFID) cards is a pervasive technology nowadays. More and more systems adopted this technology as replacement for barcodes, magnetic stripe cards and paper tickets for a variety of applications. Contactless cards consist of a small piece of memory that can be accessed wirelessly. Some of them can also have some computing capabilities.

RFID cards are of two main classes, low-frequency and high-frequency. The former operates on 125kHz frequency range and do not any cryptographic capabilities. In fact, they can be seen as radio-frequency analogue of bar codes: a card simply transmits pre-coded information.

Cards operating on a 13.56GHz frequency range (also known as *Near-Field Communication (NFC)* cards) can be made to implement more sophisticated protocols. We further talk about one particular type of such cards, MIFARE Classic.

2 MIFARE Classic cards

MIFARE Classic is perhaps the most widely used type of high-frequency cards. They were introduce in 1995 by NXP (formerly Philips).

The logical structure of MIFARE Classic is shown in Figure 1.[1] The card is in principle a memory card with few extra functionalities (read, write, increment and decrement). To perform an operation on a specific block, the reader must first authenticate for the sector containing that block. The access conditions of that sector determine whether key A or B must be used.

[1] In fact, there are two different types of MIFARE Classic cards, 1k and 4k. However they are exactly the same in the first sectors.

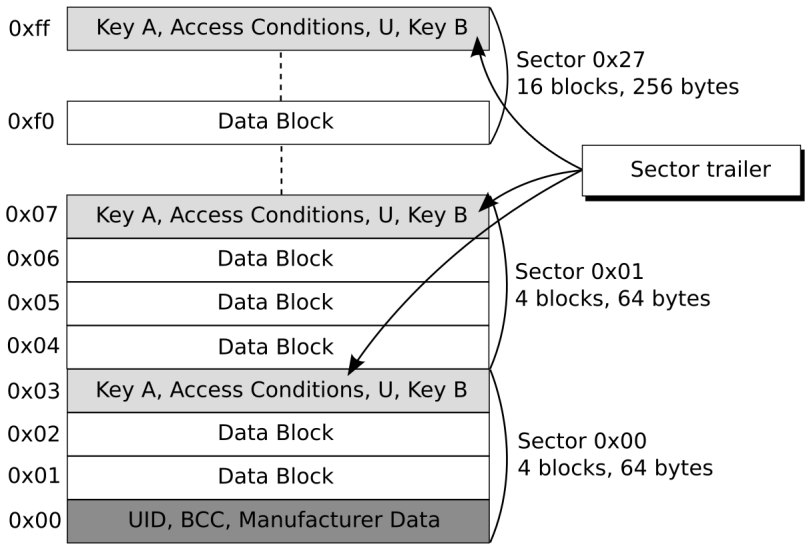

Figure 1: MIFARE Classic logical structure. Figure is taken from [2].

Note that block 0 of sector 0 contains special data. The first 4 data bytes contain the unique identifier (UID) followed by its 1-byte bit count check (BCC) which is simply XOR of all UID bytes. The remaining bytes are used to store manufacturer data. This block is read only. The keys A and B are stored in the trailer (i.e. the last block of a sector) and usually are not available for reading.

The card can perform only symmetric cryptographic primitives, and its pseudarandom number generator is stateless (hence vulnerable to replay attacks). However, the proprietary CRYPTO1 protocol was strongly based on security by obscurity and, after being reverse-engineered, proven insecure [2]. There were some new versions of MIFARE cards but because of their closed nature, their security is still questionable.

3 Estonian public transportation cards

Tallinn, Tartu, as well as some other municipalities in Estonia, use public transportation payment systems based on usage of NFC cards [1, p. 48].

Figure 2: Tallinn public transportation card (MIFARE Classic)

The cards of different models are in use. For instance, Tallinn cards (collo-

quially known as “green” cards) are built on MIFARE Classic technology; and Tartu cards are MIFARE Ultralight C cards. However, readers in buses of both cities accept each other’s cards as valid payment method. Therefore we consider the weakest solution, i.e. MIFARE Classic card.

Both cards communicate with a reader in a very basic manner. Each card stores its unique identifier (UID) and a signature that are sent to the reader during the communication process. The signature is secure (RSA1024 in Tallinn and ECDSA on curve P-192 in Tartu [1]) but it is pre-generated by external tools when the card is being initialised. This is because the card itself cannot perform asymmetric cryptographic primitives. Such an approach prevent trespassers from creating their own cards with new UIDs. On the other hand, the security against cloning attack in fact relies only on hardness of copying the card.

4 Proxmark III

Proxmark III hardware kit (see Figure 3) was originally developed by Jonathan Westhues but was later half-abandoned by the creator. However it has a very lively community which develops it further. Proxmark III can act both as a reader and a card/tag and also is able to eavesdrop communication between a card and a reader. Nowadays the kit can be bought from Rysc Corp.

Figure 3: Proxmark III hardware kit. It also includes high- and low-frequency antennas, as well as examples of cards of different standards.

Though the kit is fully programmable, its client has an extensive implemented functionality and we did not need more. The client software works on Ubuntu out of the box.

4.1 Standard keys

There exists a pre-defined set of keys A and B[2]. Proxmark III can automatically test these keys on a particular card. After applying this procedure on Tallinn "green" card it turned out that all the keys used are from this classical set:

```
Detailed keys recovered (from "hf mf chk * ?")
sector | key A        | key B
-------|--------------|--------------
0      | a0a1a2a3a4a5 |
1      | d3f7d3f7d3f7 |
2      | d3f7d3f7d3f7 |
3      | d3f7d3f7d3f7 |
4      | d3f7d3f7d3f7 |
5      | d3f7d3f7d3f7 |
6      | d3f7d3f7d3f7 |
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
```

Therefore the contents of the card are, in fact, stored unprotected.

4.2 Communication eavesdropping

We used Proxmark III to eavesdrop the communication between reader in public bus in Tartu and Tallinn card used for travel payment. Here `Rdr` denotes messages sent by the reader, and `Tag` – by the card.

First the reader and the card perform anti-collision operations:

```
| Src | Data (! denotes parity error)        | CRC | Annotation         |
|-----|--------------------------------------|-----|--------------------|
| Rdr | 52                                   |     | WUPA
| Tag | 04  00                               |     |
| Rdr | 93  20                               |     | ANTICOLL
| Tag | cb  12  8d  58  0c                   |     |
| Rdr | 93  70  cb  12  8d  58  0c  18  0b   |     | SELECT_UID
| Tag | 08  b6  dd                           |     |
```

During this phase the card tells its UID (here `cb 12 8d 58`) and type (`08 b6 dd` stands for "MIFARE Classic 1k").

Next, there is an authentication phase:

```
| Src | Data (! denotes parity error)        | CRC | Annotation         |
|-----|--------------------------------------|-----|--------------------|
| Rdr | 60  03  6e  49                       |     | AUTH-A(3)
```

[2]To be precise, 10 keys

```
| Tag | e8  a4  68  dc                        |     |
| Rdr | bb  63! 3e  e7  99! 06! 0c  0c        | !crc| ?
| Tag | d1! ea  48! 35!                       |     |
```

The first command here is request for authorisation with key A (command `60`) for the block number 3 (second byte, i.e. `03`). Second, third and forth lines here stands for exchange of nonces, i.e. challenge-responce sequence (ks_1, ks_2 and ks_3 represent the keystream):

- in the second line the card sends to the reader a challenge nonce n_T;
- in the third line the reader sends $n_R \oplus ks_1$ and $a_R \oplus ks_2$, where n_R is its own challenge nonce a_R is the answer to the card's challenge;
- in the forth line the card sends its reply $a_T \oplus ks_3$ to the reader's challenge.

Starting from the third line, all communication is encrypted. We omit the further parts of the eavesdropped communication.

The exclamation marks denote parity errors. However, since the parity data is also encrypted, usually it simply means that the data is encrypted.

5 Conclusion

This report present a small overview of NFC/RFID cards with one particular real-life application, Estonian public transportation system. While the cards have proven to be insecure, they are still widely used in non-critical applications.

References

[1] Cybernetica. Cryptographic algorithms lifecycle report 2016.

[2] Flavio D Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter Van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs. Dismantling MIFARE classic. In *European Symposium on Research in Computer Security*, pages 97–114. Springer, 2008.