VERSITA

Annales UMCS Informatica AI XI, 3 (2011)
129–139; DOI: 10.2478/v10065-011-0001-z

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

The security of the multi–application public transport card

One mifare card exploration

Łukasz Papież*

Military University of Technology, Warsaw

Abstract

Mifare Classic widely used as a public transport card based on the weak cipher Crypto-1 broken three years ago with a number of serious attacks published by researchers from the Dutch University of Nijmegen and still another was developed at University College of London. The report entitled Cloning Reactivation published in the Polish Computerworld magazine presented the security of Warsaw City Card at that time. It also announced that starting from 2010 the security of the Warsaw system would undergo an upgrade with the usage of 3DES algorithm. While in London all new Oyster cards emitted since 2010 are more secureDesFire cards, the security of the Warsaw card stays nearly the same.

1. Introduction

Over the last decades public transport ticketing infrastructure was based simply on paper tickets and validators punching tickets. Due to the growing number of fake paper tickets similar to the original ones, the ticketing systems were upgraded to contactless cards which in principle are harder to copy (clone). This technology requires the new whole ticketing infrastructure.

*E-mail address: lukaszpapiez@gmail.com

The key ingredient in modernizing the fare collection by public transport providers, and a new way of issuing tickets and controlling their usage, is an electronic public transport card. The idea was to use a contactless smartcard as a ticket with a possibility to top up the same card without necessity to buy a new piece of plastic for the next month ticket. The new ticketing system consists of contactless smart cards and readers. The future idea was to change one-application transport card with a multi-application one. It allows the user to employ just one card e. g. for public transport, buying tickets for the cinema, visiting a museum etc.

Nowadays the public transport system based on the multi-application card is widely used in a number of Polish cities. The card usually is based on Mifare Classic technology. The more widespread use of smart cards creates concerns about their security. The principal security concerns can be divided into several categories: hardware and software security issues, the cryptographic algorithm issues, key management issues, random number generation issues, etc. Over the last four years a number of vulnerabilities and attacks were found in each of these categories. The Mifare Classic cipher Crypto-1 and the chip in which it is used were thoroughly studied. In many cases different vulnerabilities can be combined in one single attack and can make that the security of certain card systems will be close to zero.

In this article we try to obtain the state of security of contactless smart cards based on the Mifare Classic chip especially the one used as a public transport card in Warsaw.

2. Mifare classic memory organization

According to [**1**] EEPROM memory of Mifare Classic 1K is divided into 16 sectors of 4 data blocks each. The basic block has 16 bytes.

Block 0 of sector 0 is called the Manufacturer Block. It contains special data which cannot be modified because of the lock at the production process. The first four bytes store the Unique IDentifier of the card (UID) on 32 bits. The following 1 byte contains the bit count check (BCC) and the remaining bytes store manufacturer data.

The last block of each sector is called Sector Trailer. It contains the secret key A, Access Condition Bits for a sector and the optional secret key B.

Any memory operation i.e. Read, Write etc. needs the first authentication process done for the proper sector according to its Access Bits. The authentication process has several steps.

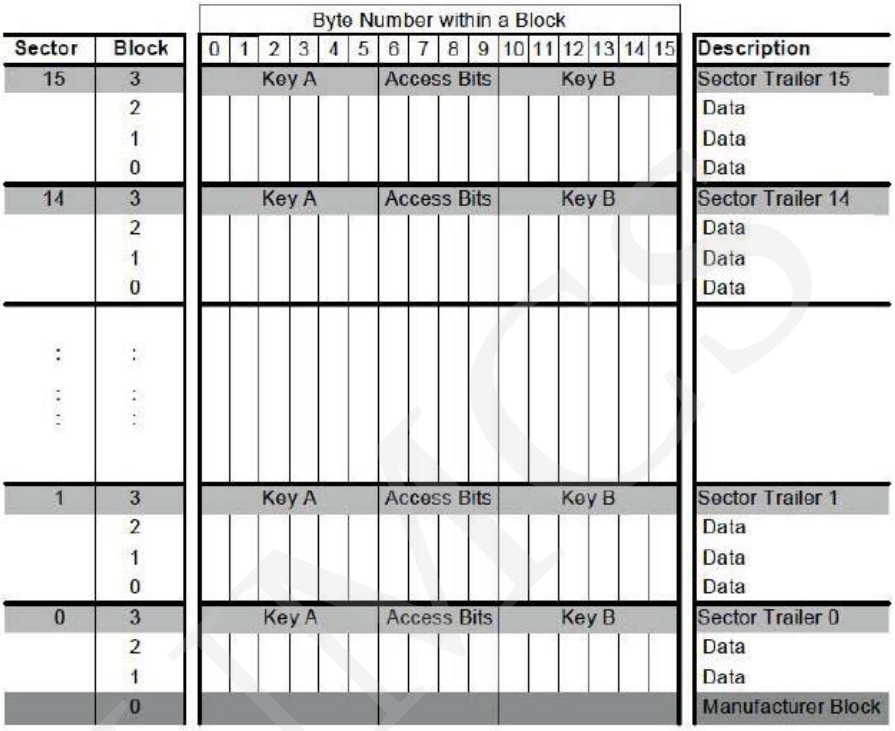

Fig. 1. Mifare classic memory.

When the tag enters the electromagnetic field of the reader and powers up, it immediately starts the anti-collision protocol by sending its uid and next as specified in ISO14443-A [**2**] the tag is selected.

2.1. The three-step authentication

- The reader sends an authentication request for a specific block. Next, the tag picks a random challenge nonce on 32 bits N_T and sends it to the reader in the clear.
- Then the reader sends its own challenge nonce N_R together with the answer a_R to the challenge of the tag.
- The tag finishes authentication by replying a_T to the challenge of the reader.

Starting with N_R, all communication is encrypted which, means that N_R, a_T, and a_T are XOR-ed with the keystreams $ks1, ks2, ks3$.

3. Sector access conditions

Access condition bits are stored in the last block of each sector called Sector Trailer. The bits are stored once inverted and once non-inverted. The bits, according to Fig. 2, Fig. 3, Fig. 4 denote the access condition for the proper block of memory. The main task of the bits is to control access and modification of the blocks of data using the appropriate key A or B. The format of access bits is specified by the internal chip logic. Changing the format, incompatible with the logic results in permanent lock of the sector.

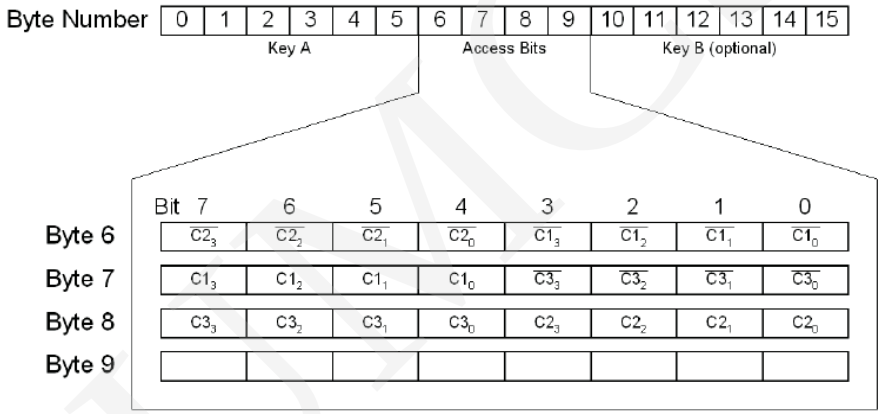

Fig. 2. Access condition.

From the scheme taken from Figure 2 we can obtain the values of C1, C2, C3 from the access bits. Hence it is possible by using Figures 3 and 4 to define the card operation which can be performed on the sector trailer block and data blocks with a given key A and B.

The tables below include the interpretation of access condition:

4. The state of public transport card in Warsaw

Three years ago, scientists considering the Mifare Classic Card began a detailed discussion and analysis of security of various systems which use this card.

There were published a lot of reports about new weakness of each broken card, especially transport cards, based on the Mifare Classic chips, as used in London, Amsterdam, Warsaw etc. In one of the reports from Computerworld Magazine (June 9th, 2009) - Cloning Reactivation [**3**], we could find some information about the state of security on the Warsaw City Card at that time. The most interesting was:

"The weakest tested card was Warsaw City Card"

Table 1. Access conditions for the sector trailer.

Access bits			Access condition for					
			KEY A		Access bits		KEY B	
C1	C2	C3	read	write	read	write	read	write
0	0	0	never	key A	key A	never	key A	key A
0	1	0	never	never	key A	never	key A	never
1	0	0	never	key B	key A or B	never	never	key B
1	1	0	never	never	key A or B	never	never	never
0	0	1	never	key A	key A	key A	key A	key A
0	1	1	never	key B	key A or B	key B	never	key B
1	0	1	never	never	key A or B	key B	never	never
1	1	1	never	never	key A or B	never	never	never

Table 2. Access conditions for data blocks.

Access bits			Access condition for					
			KEY A		Access bits		KEY B	
C1	C2	C3	read	write	read	write	read	write
0	0	0	never	key A	key A	never	key A	key A
0	1	0	never	never	key A	never	key A	never
1	0	0	never	key B	key A or B	never	never	key B
1	1	0	never	never	key A or B	never	never	never
0	0	1	never	key A	key A	key A	key A	key A
0	1	1	never	key B	key A or B	key B	never	key B
1	0	1	never	never	key A or B	key B	never	never
1	1	1	never	never	key A or B	never	never	never

but ZTM (Institution for management of public transport in Warsaw) said that even this year, the card will be upgraded with a more secure 3DES algorithm.

In this article we will try to verify, after three years of publication that report, if the card is now secure. Moreover the immunity to known old attacks will be checked [**4**].

The attack [**4**] can be used applying a special software, to control timing of generation of the same start value for authentication every time we need. Using just [**4**] attack, we are sending about 300 queries to the reader to obtain the key for one sector. It will take about 5 min for a standard reader, and a few seconds for Proxmark 3. With extension for [**4**] of Nested Attack [**5**] - allowing us to obtain every key very fast, after getting one key we can clone the card in a few seconds as well.

The main difference between the card from last 5 years and from today is the design of the card cover, due to the personalization process. Name, surname and photo, after personalization card change there are the only new elements on the whole card. The attack above, done three years ago is applicable as well today. Our tests seem to indicate that the layout and the security of the date on this card have not changed since 2009.

Moreover, the tests on the Warsaw City Card pointed out that as it was before, the keys can be different for different sectors and different for key A and key B, but will always be identical for two different Warsaw cards. This is not at all the case with the London Oyster cards. The author tried to find the possible solution for the question why that is. Referring to the document: Practical Attacks on the MIFARE Classic by Hon Tan Wee [**6**], in sections 4.3.1 Cipher keystream initialization & Generation, it was found that the authentication process is possible only in case of the card and the reader will be initialized in the same way. There is one special sequence to initialize the cipher: 48 bit key K, the value of the card $N_T \bigoplus UID$ and a random value chosen by the reader N_R. The reader must implement the same initialization sequence. Currently, the reader will choose the proser key for a given sector it wants to read and the initialization process on the reader side is the same as on the card side.

The author of this article claims that the transport card readers in Warsaw have implemented fixed keys for each sector used by the application, as the card. This is a simplistic design which allows the interaction of each card with each card reader in the same way, without key diversification. If the keys on each card were different, which seems to be more secure, the reader would need to be able to recompute the key for each card, which might be more costly and harder to manage.

On the other hand, if we focus on the London public transport, the same attack applied on the London Oyster card - used as a public transport card there, brings a completely different solution. We can see that the keys for each card are different. According to investigations of the Oyster card we know that the keys computed by the reader are different for each card which confirms our results.

Comparing both electronic ticketing systems in Warsaw and in London, we can see that the chip is the same but the application solution and the back-end support and infrastructure used in each system are completely different. It is more sophisticated and more secure in London. Specifying, the system used in London is much more secure, especially after the practical attack [**4, 5**] was published. There seem to be some serious changes on both card and system

of London Oyster Card. Contrary to Oyster, the company responsible for the security of Warsaw City Card did not do so.

The conclusion is that not just the algorithm implemented on the card should be hard to break, but the whole system for transport management including card, readers, infrastructure etc. Considering Warsaw City Card, we can see that the implemented ticket system is very poor from the technical point of view. The solution used in Warsaw is simple but unfortunately, much less secure: if we find one key for one card, we can obtain every key for every transport card due to implementation of the same key on each card which is a serious disadvantage in developing a smart card system.

5. Warsaw city card exploration

By applying the card-only Dark Side Attack [**4**] using Open Source MFCUK (MiFare Classic Universal toolKit) application [**7**] which has implemented the Attack, Warsaw City Card can be totally broken. That case allows to perform some tests of the card application which brought interesting findings for the analysis. In order to interpret the entries to the blocks, numerous experiments are needed. The point was to make repeated scanning of different cards in different states. By different cards, we understand different types of tickets encoded on the card i.e. student card discount, senior card discount etc.

The first observation is that some sectors of the card have different permission bits from other sectors (as found in the sector trailer). The sector trailer contains only permission bits set to 787788, (all other bits are set to 0). Each of the other sector trailers after permission bits has set key A.

Block 007: Login OK. Data: 00000000000078778800000000000000 ....xw.....

Table 3. Access bits in the Sector Trailer.

Binary(0x78)	0	1	1	1	1	0	0	0
Access bit	$\overline{C2_3}$	$\overline{C2_2}$	$\overline{C2_1}$	$\overline{C2_0}$	$\overline{C1_3}$	$\overline{C1_2}$	$\overline{C1_1}$	$\overline{C1_0}$
Binary(0x77)	0	1	1	1	0	1	1	1
Access bit	$C1_3$	$C1_2$	$C1_1$	$C1_0$	$\overline{C3_3}$	$\overline{C3_2}$	$\overline{C3_1}$	$\overline{C3_0}$
Binary(0x88)	1	0	0	0	1	0	0	0
Access bit	$C3_3$	$C3_2$	$C3_1$	$C3_0$	$C2_3$	$C2_2$	$C2_1$	$C2_0$

With reference to [**1**], the interpretation of the bits 787788 is that the sectors can be read with keys A and B, but to write operation we need only key B.

Table 4. Access conditions for the Sector Trailer.

Access bits			Access condition for					
			KEY A		Access bits		KEY B	
$C1_3$	$C2_3$	$C3_3$	Read	Write	Read	Write	Read	Write
0	1	1	never	key B	key A or B	key B	never	key B

Table 5. Access conditions for Data Blocks.

Access bits			Access condition for				
C1	C2	C3	Read	Write	Increment	Decrement, Transfer, Restore	Application
1	0	0	key A or B	key B	never	never	read/write block

Thus we assume that only sectors without key A being equal to 0 must be used by the Warsaw City Card system. The application is encoded on the 3 sectors of the card. Moreover, the first sector of the card contains a well-known default key A0A1A2A3A4A5. That information is very useful due to the fact that one known sector key allows to recover other known keys quite easily by the Nested Authentication Attack [**5**]. For a new ticket top up process there is exactly one block of one card sector overwritten. The activation process will overwrite current and the next sector, but still one block of each sector. Other blocks which are not used will contain zeros, which is the default state of memory in the new MiFare chips.

It is worth noting that during the activation and top-up process of two cards previously containing the same data, done in two different readers, the new data stored on the card will be different in each card. In addition, stored data depends on the time of activation. On the other hands, activation of the two cards, containing the same data, in the same reader, within the same minute, overwrite the proper block with the same data. Activation in the next minute causes slight modification of data on the second card. This makes us conclude that the ticket data depends on the reader and on the time of activation, but it seems that it does NOT depend on the UID of the card. The fact that the ticket is encoded only in one block of one sector on the card, means that to overwrite the card block with a proper ticket data, being in possession of old card, which has expired, we need just one key to have access to one required sector. All other sectors used by the public transport application will be updated automatically.

As we can see, the security of widely used Warsaw City Card is very poor. The cards used in public transport in Warsaw are the same as three years ago, despite the assurances from the Public Transport Company about the change for

a new card with a secure 3DES algorithm. The main reason for not changing the old technology was a large number of cards (500,000) bought by the company. Changing the technology after just a few years means for the company waste of large amount of money. On the other hand, there is no evidence that card cloning is done, and there is no visible impact on the business, therefore there is no real need to change the whole system.

6. Default and easy keys use

In many articles there was described a serious bug of Mifare classic - default keys. Their aim was to make new Mifare Classic clients take care of key change. One of the fundamental points of smart cards security engineering is to make the card as secure as possible including the security of internal structure and that of the secret keys. Default keys - published in a specification, simply break this rule at the beginning. The Mifare Classic card is used as a form of ID in many universities and businesses, especially in firms and organisations which should require a high level of security. From the logical point of view, it means that a corporation card should be more secure than e. g. Public Transport Card. The University ID card can be also used as a Warsaw City Card due to its multiple applications capability. The main mistake made by the developers of these systems is not changing default keys, or using very simple keys which are not random, and therefore easy to guess, even without any knowledge about cryptographic protections used.

In order to see how serious the situation is, we have conducted a number of experiments for a number of the public institution identifier cards, with cards being used by a large number of people. The tests have shown that some of the tested cards possessed default keys, others had a simple key like 11FFFFFFFFF, where probability of finding it by guessing is very high.

The main problem of the simple key is its vulnerability to an optimized key-guessing attack which could be much faster than brute force. The developers have created a simple multi-application card system based on the MiFare Classic cards. It means that the card can be used as e. g. Transport Card, Parking Card, University Identifier etc. Before the attack proper preparation dismissing not used sectors by the application which we are going to hack should be made.

The attack is going to concentrate on some well-chosen sectors (out of 16 possible secotrs in a 1K card). Then we can apply a kind of optimized brute force attack which tries various keys not at random, but in the order of their simplicity, with keys with repetitions and patterns being tried first, and with really random keys being tried last. This is expected to work well, with a succes

similar to a dictionary attack on passwords. One example is given on the next page.

Studying the structure of the card provides us some knowledge about the use of the internal sectors by the application. The first sector is locked for any modification. Moreover, it contains mostly a default key A0A1A2A3A4A5 for key A and B0B1B2B3B4B5 for key B, hence access to the sector is possible. That knowledge lets us exclude the first sector from the suspects. If the card is used for multi-application, we can assume that the earlier application is placed on the first few sectors. It aims to avoid applications overlapping. The latest applications should be placed on the end sectors of the card. Such assumption lets us efficiently reduce suspicious sectors.

As an example some of the tested keys had a simple structure with a constant central part as follows:

$$\text{XX00112233XX, where } X \in \{0, \ldots, F\}.$$

We need to query just $16 \cdot 16 \cdot 16 \cdot 16 = 65536$ keys in the worst case for each body. Being in a possession of a fast reader (30 query per sec) it takes about 30min to search the entire space of solutions for one key-body. To make optimized brute force on such a key we do not need any specialized equipment to recover a proper sector key. Being in a possession of a cheap reader like ACR122 for 30 euro and simply PC (or Laptop) it is strongly possible to make the attack as well. One key searching time (with a chosen key-body) takes less than 24 hours which is still possible on a PC.

Below we give some examples of widely used keys:

$FFFFFFFFFFFF$	$A0A1A2A3A4A5$
$B0B1B2B3B4B5$	$C0C1C2C3C4C5C6$
$D0D1D2D3D4D5D6$	$4D3A99C351DD$
$1A982C7E459A$	000000000000
00112233445566	$D3F7D3F7D3F7$
$AABBCCDDEEFF$	$AAAAAAAAAAAA$

7. Card security improvement

The Mifare Classic card is a widely used card in public transportation applications and the faulty Crypto-1 algorithm cannot be easily replaced. However, it is possible to use the same card in many different ways. Here are some solutions proposed by the author for a fast and low-cost security improvement of existing systems with existing cards.

7.1. Storage of random data

All blocks not used by the application implemented on public transport cards based on Mifare Classic are filled with zeros. This situation allows the attacker to recognize quickly which sectors are used by the application. The solution is to fill every unused byte of memory on the card with some random data, which has no influence on the proper application. In addition these data could also be changed to other random data when the card is used. This will make it much more difficult for the attacker to reverse engineer the application.

7.2. Key Diversification

The cryptographic key to access each card should be different and depend on the card unique UID. For this reason, the Oyster card in London, even before the recent upgrade of this chip, was more secure than Warsaw card today, though it used exactly the same chip Mifare Classic 1K.

8. Conclusions

The security of the multi-application Warsaw City Card based on Mifare Classic technology as well as any other contactless smartcard based on the same technology is poor. Experiments on the Warsaw City Card brought pessimistic conclusion that not just card but the whole public transport electronic system is not secure and it would give the possibility for cloning the card. The experiments on the other card based on Mifare Classic show that in some cases due to the default key used there is no necessity of applying cryptographic attack to obtain the secret keys but simply brute force attack is enough.

References

[1] MF1ICS50 functional specification, January 2008.
[2] ISO/IEC 14443. Identification cards - Contactless integrated circuit(s).
[3] http://www.computerworld.pl/artykuly/346011/Klonowanie.reaktywacja.html.
[4] Curtois N., The Dark Side of Security by Obscurity and Cloning MiFare Classic Rail and Building Passes Anywhere, Anytime, In SECRYPT 2009 proceedings, International Conference on Security and Cryptography, Milan, Italy, 7-10 July 2009.
[5] Garcia F. D., van Rossum P., Verdult R., Wichers Schreur R. (2009). Wirelessly Pickpocketing a Mifare Classic Card. In Oakland IEEE Symposium on Security and Privacy.
[6] Wee Hon Tan, Practical Attacks on the MIFARE Classic, Imperial College London, September 2009
[7] MFCUK MiFare Classic Universal toolKit - Dark Side attack implementation. http://code.google.com/p/mfcuk/source/checkout.