



**Middle East Technical University  
Informatics Institute**

## **Security Assessment of the Smartcard Systems at Middle East Technical University**

**Advisor Name: Assoc. Prof. Dr. Cihangir TEZCAN  
(METU)**

**Student Name: Berkay AKÇÖREN  
(CSEC)**

**January 2025**

**TECHNICAL REPORT  
METU/II-TR-2025-**



**Orta Doęu Teknik Üniversitesi  
Enformatik Enstitüsü**

## **Orta Doęu Teknik Üniversitesi'ndeki Akıllı Kart Sistemlerinin Deęerlendirilmesi**

**Danışman Adı: Doç. Dr. Cihangir TEZCAN  
(ODTÜ)**

**Öğrenci Adı: Berkay AKÇÖREN  
(CSEC)**

**Ocak 2025**

**TEKNİK RAPOR  
ODTÜ/II-TR-2025-**

# REPORT DOCUMENTATION PAGE

<b>1. AGENCY USE ONLY (Internal Use)</b>	<b>2. REPORT DATE</b> 10.01.2025
<b>3. TITLE AND SUBTITLE</b>  Security Assessment of the Smartcard Systems at Middle East Technical University	
<b>4. AUTHOR (S)</b>  Berkay AKÇÖREN	<b>5. REPORT NUMBER (Internal Use)</b>  <i>METU/II-TR-2025-</i>
<b>6. SPONSORING/ MONITORING AGENCY NAME(S) AND SIGNATURE(S)</b> Non-Thesis Master's Programme, Department of Cyber Security, Informatics Institute, METU Advisor: Assoc. Prof. Dr. Cihangir TEZCAN      Signature:	
<b>7. SUPPLEMENTARY NOTES</b>	
<b>8. ABSTRACT (MAXIMUM 200 WORDS)</b> Smartcards are portable devices that can communicate wirelessly and contain microchips capable of transferring, processing or storing data. They serve practical and reliable solutions for everyday problems such as identification, access control, and payment. Therefore, security of these systems is cardinally important. Smartcard systems serve a critical role in the Middle East Technical University by enabling access control to restricted areas, on-campus payments and managing identification for student and personnel. Given the importance of data and the usage of the system's role in daily operations, in the scope of this project at hand, we are conducting a security assessment of the current smartcard system at Middle East Technical University. In this assessment, we aim to identify vulnerabilities, weaknesses, and potential threats with their consequences by applying passive attack techniques on smartcards.	
<b>9. SUBJECT TERMS</b>	<b>10. NUMBER OF PAGES</b>  23

# Table of Contents

1. INTRODUCTION .....	1
2. MIFARE CLASSIC SMARTCARDS .....	5
2.1 Darkside Attack .....	6
2.2 Nested Authentication Attack .....	6
2.3 Hard Nested Attack .....	7
2.4 Fudan Backdoor Keys .....	8
3. METU SMARTCARDS .....	9
4. CONCLUSION .....	21
References .....	22

## List of Figures

Figure 1: CRYPTO1 Stream Cipher Structure .....	5
Figure 2: Initialization of MIFARE Classic Authentication Protocol .....	6
Figure 3: Proxmark3 Hardware with Low Frequency and High Frequency Antenna .....	9
Figure 4: Proxmark3 Terminal User Interface .....	10
Figure 5: Proxmark3 Client Output for Card Number 1 with identified Fudan backdoor key ....	11
Figure 6: Blank Card with Changeable UID. Purple Sticker is an Indicator. ....	12
Figure 7: Proxmark3 output for the Magic Gen2 ghost card .....	12
Figure 8: Simple Android application that can automatically copy UID of the card in the proximity of the phone .....	13
Figure 9: Terminal Output of Proxmark3 Client on MIFARE Classic Commands .....	14
Figure 10: Dump of the card number 1 .....	15
Figure 11: Comparison of the same card before and after lunch payment .....	16
Figure 12: Ghost card with a balance more than million TRY .....	17
Figure 13: Picture of the verified payment with a ghost key fob .....	17
Figure 14: General information about the ID card number 3 .....	18
Figure 15: Secret keys obtained from the METU ID cards. Left column is used only in the new cards. Middle column is used in both old and new ID cards. Right column is only used in the old card. Default keys are indicated. One of the keys has also a meaning AnKaRa in ASCII format, which makes it a predictable key. ....	18

## List of Tables

Table 1: MIFARE Classic 1K Data Structure.....	7
Table 2: Universal Backdoor keys in Fudan based cards. ....	8
Table 3: List of Identification Cards .....	11

# CHAPTER 1

## 1. INTRODUCTION

Smartcards are portable devices that can communicate wirelessly and contain microchips capable of transferring, processing or storing data. These cards use radio-frequency identification systems also known as RFID. A regular RFID system is a combination of transponder, receiver and transmitter. In this project at hand, we solely focus on high frequency (HF) 13.56 MHz smartcard technology.

High frequency cards are mostly used in the access control systems, modern payment systems, bank cards etc. Like in most communities, smartcards play a crucial role in our university's campus life. Personalized ID cards are used for access control and payment process. These cards act as a personalized key, granting students, staff and faculty members access to specific areas such as dormitories, classrooms, libraries, and laboratories based on their role and permissions. In addition to access control, these ID cards can be used in the payment process in an isolated campus environment. Students use their smartcards to pay for food at student cafeterias, books at libraries, home appliances in dormitories and several campus life activities. With a quick proximity tap, a simple transaction can be processed instantly. This reduces waiting times, dependence on cash and doing access control and payment at the same time. Due to the critical applications and services provided by smartcards, security of smartcard systems is cardinally important. In our university, smartcards are used in the access control of main campus entrance, dormitories, library, departmental buildings, study halls, meeting rooms and even personal offices in some buildings. Due to these use cases, smartcards should be secure to avoid possible unauthorized entries and dishonest payments.

Specifications for contactless proximity cards are standardized in ISO 14443-A [1]. This standard specifies details about physical characteristics, radio frequency power, signal interface, initialization and collision, and transmission protocol.

MIFARE, one of the most used contactless smartcards on the market, is a family of smartcards produced by NXP Semiconductors and furnished by proprietary solutions that are compatible with ISO/IEC 14443 Type-A standard. MIFARE Classic is compatible with the first three parts of the ISO 14443 Type-A standard, but it employs its own communication layer. It utilized a proprietary security protocol for authentication and encryption.

Contactless communication in MIFARE Classic products is secured via symmetric stream cipher called CRYPTO1. NXP Semiconductors kept CRYPTO1 cipher as a trademark secret until 2008. MIFARE Classic cards are reverse engineered in 2008 by Nohl *et. al.* [2]. After the reverse engineering, both stream cipher and communication protocol are exposed to public, and many vulnerabilities and attacks were provided by researchers. For the convenience, we can group attacks in the literature as intrinsic and non-intrinsic. Intrinsic attacks cannot be mitigated by breaking backwards compatibility.

The first intrinsic vulnerability is CRYPTO1 stream cipher itself, which lies in the core of the intrinsic attacks. The first practical attack against CRYPTO1 is the key stream recovery proposed by Garcia *et al.* [3]. The second intrinsic vulnerability is parity bits that are used in the communication process are applied on plaintext but sent as encrypted. These parity bits are encrypted with reused keystream bits. This already breaks the confidentiality of communication [4]. The third intrinsic vulnerability is that, in an encrypted re-authentication process, challenge nonce  $n_T$  is encrypted with the new key [2]. In the following sections, we describe practical attacks in detail that have emerged from these vulnerabilities.

Moreover, there are many non-intrinsic vulnerabilities which are patched in the next generation of MIFARE cards. These vulnerabilities include weak pseudo random generator (PRNG), which leads to recovery of the stream ciphers internal state generated by the linear feedback shift register (LFSR). In addition, in the old MIFARE cards, PRNG is based on the clock cycle which repeats itself every 0.6 seconds. Finally, 4-bit authentication codes in the communication process are sent encrypted. Details about attacks that have emerged from these vulnerabilities are given in the following sections.

In this project at hand, we demonstrate MIFARE Classic cards that are used in Middle East Technical University (METU) can be hacked and copied. We also present tools that we used in

the hacking process, analyze some old cards and point out security flaws in the smartcard system in our university.



## CHAPTER 2

### 2. MIFARE CLASSIC SMARTCARDS

METU uses MIFARE Classic 1K cards for smartcard systems. These cards use CRYPTO1 stream cipher with 48-bit secret key. MIFARE Classic 1K cards have 1 kilobyte memory. This memory is structured in 16 sectors in which every sector has a 4-block memory and each block has a 16-byte memory. Every sector trailer has 2 different keys namely key A and key B, with corresponding access bits for defining access conditions. Most of the time, the very first sector of the cards, so-called manufacturer block, is written by manufacturer and contains 4- or 7- byte unique identifier (UID). Other memory blocks can be used to store information for application specific purposes. Keys A and B can be set per sector for the corresponding sector for fine-tuned access control. Data structure is visualized in Table 1.

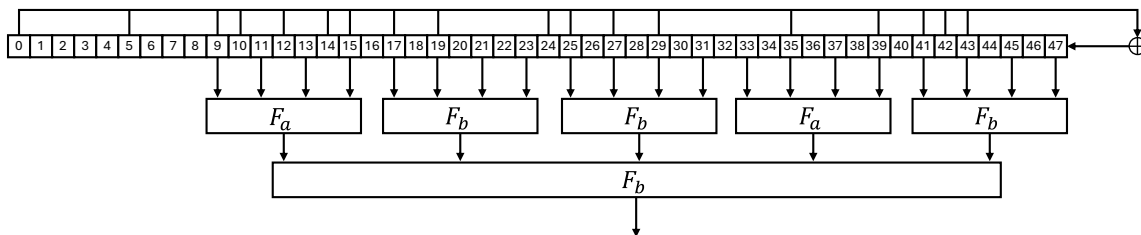


Figure 1: CRYPTO1 Stream Cipher Structure

MIFARE Classic 1K cards use CRYPTO1 stream cipher with 48-bit secret key. CRYPTO1 is a linear feedback shift register (LFSR) based stream cipher which consists of 48-bit LFSR and filter functions. General structure of the cipher can be seen in Figure 1. 48-bit secret key is too short for today's standards. It can be easily brute-forced in a relatively short time. This can be prevented thanks to the delay which is introduced in the communication protocol. However, this still makes the cipher vulnerable to pre-computed rainbow tables attacks. Also, offline brute-force attacks are efficiently implemented in literature [5].

Because of the proprietary transmission protocol and security by obscurity approach, there are also design flaws in the authentication process. Practical card-only attacks due to these design flaws are listed below.

## 2.1 Darkside Attack

The Darkside attack [6] exploits two vital design flaws. Namely, the leakage of not acknowledged codes (NACKS) and initial nonce repeating itself. In the authentication process seen in Figure 2, card sends its UID ( $u$ ) and challenge nonce  $n_T$ . Then the internal state of the stream cipher is initialized and the reader replied with answer  $\{a_R\}$  and challenge response nonce  $\{n_R\}$ . Curly brackets indicate encrypted data during the transmission process. Tag responses with  $\{a_T\}$  and authentication is completed. During this authentication process, when card receives  $\{n_R\}$  and  $\{a_R\}$ , it checks if parity bits are correct. If answer  $\{a_R\}$  is wrong, but the 8 encrypted parity bits are correct, the reader sends encrypted 4-bit NACKS. So, with probability of  $1/256$ , the card replies with 4-bit NACKS and leak 4-bit of keystream. Combining this with repeating nonce, an attacker can break a key even if there is no known key already. This attack can be mitigated by strengthening PRNG and removing NACK codes.

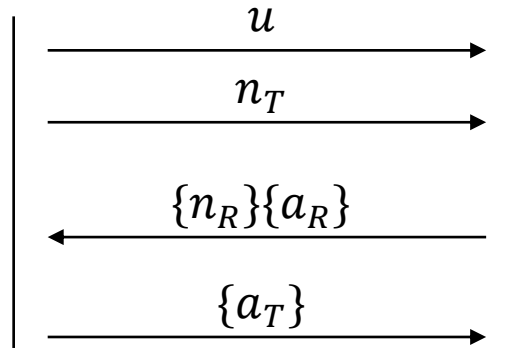


Figure 2: Initialization of MIFARE Classic Authentication Protocol

## 2.2 Nested Authentication Attack

The Nested Authentication Attack is discovered by [4]. This attack requires an already known key to be executed. In an authenticated card, when reauthentication process is initiated, card sends  $\{n_T\}$ , instead of  $n_T$ . With a predictable nonce generated by weak PRNG, an attacker can recover

32-bit of the 48-bit key stream wirelessly. This attack can be mitigated by strengthening the PRNG, without breaking the backwards compatibility.

Table 1: MIFARE Classic 1K Data Structure

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0																	Manufacturer Block
	1																	Data
	2																	Data
	3	Key A					Access Bits					Key B					Sector Trailer	
1	4																	Data
	5																	Data
	6																	Data
	7	Key A					Access Bits					Key B					Sector Trailer	
2	8																	Data
	9																	Data
	10																	Data
	11	Key A					Access Bits					Key B					Sector Trailer	
3	12																	Data
	13																	Data
	14																	Data
	15	Key A					Access Bits					Key B					Sector Trailer	
4	16																	Data
	17																	Data
	18																	Data
	19	Key A					Access Bits					Key B					Sector Trailer	
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
14	56																	Data
	57																	Data
	58																	Data
	59	Key A					Access Bits					Key B					Sector Trailer	
15	60																	Data
	61																	Data
	62																	Data
	63	Key A					Access Bits					Key B					Sector Trailer	

### 2.3 Hard Nested Attack

The Hard Nested Attack is proposed by [7] is so far the most sophisticated attack in literature. This attack is solely based on parity bits and is a ciphertext-only attack. Hence, this attack cannot be mitigated without breaking backwards compatibility. Hard Nested Attack is a nested attack,

which requires a known key to be executed. This attack can be applied to hardened versions of MIFARE Classic cards such as EV1. If a card is MIFARE Classic, it is susceptible to this attack.

## 2.4 Fudan Backdoor Keys

Fudan cards are produced for low-cost alternatives that are compatible with MIFARE protocol. Fudan variant cards are common in the smartcards systems, since they offer cheaper option to the MIFARE cards. Philippe Teuwen [8] found that Fudan based cards has its own verification method with the Fudan Android application for determining if the card is genuine or not. This application uses a different sector and block (namely blocks 128-135, which should not be available in the original card). Through nested attacks and static nonce generation, they discover there is a back door key in the entire production line for every sector. These keys are listed in Table 2.

*Table 2: Universal Backdoor keys in Fudan based cards.*

A396EFA4E24F
A31667A8CEC1
518B3354E760

In the following section we summarize the enumeration and exploitation of the smartcard system at METU. We showcase the tools that are used in the enumeration process and briefly mention the access control and payment process in METU.

## CHAPTER 3

### 3. METU SMARTCARDS

As we mentioned in the previous chapters, MIFARE Classic 1K smartcards are used in the Middle East Technical University. To analyze the security of METU smartcards, Proxmark3 is used. Proxmark3 is a multi-purpose, open-source hardware tool for RFID research, development and security analysis. It can be programmed to automate read, write, emulate, fuzz and eavesdrop communication.



*Figure 3: Proxmark3 Hardware with Low Frequency and High Frequency Antenna*

Sample Proxmark3 hardware can be seen in Figure 3. It can be bought for approximately ~100 USD from the internet. Since it is an open-source hardware, one can build their own hardware or buy it from third-party vendors for a much cheaper price. The software and the hardware behind Proxmark3 are maintained by a community full of RFID security enthusiasts [9]. Most updated and well-furnished software for Proxmark3 is maintained by Iceman [10]. This software helps analysts to analyze a wide range of cards with different protocols by mimicking the communication protocol. Analysts use Proxmark3 to identify security flaws and exploits in the

RFID systems. Proxmark3 Client software is a command line interface application. Simple terminal screenshot is provided in Figure 4.

```
berkay@GAMEBOX:~/repositories/proxmark3$ ./pm3
[=] Session log /home/berkay/.proxmark3/logs/log_20241228112121.txt
[+] loaded '/home/berkay/.proxmark3/preferences.json'
[+] Using UART port /dev/ttyACM0
[+] Communicating with PM3 over USB-CDC

88888888b. 888b   d888   .d8888b.
888   Y88b 8888b d8888 d88P   Y88b
888   888 88888b.d88888   .d88P
888   d88P 888Y88888P888   8888"
88888888P" 888 Y888P 888   "Y8b.
888   888   Y8P 888 888   888
888   888   "   888 Y88b d88P
888   888   888   "Y8888P"   [ ● ]

[ Fuel open source revolution! ]
  Patreon - https://www.patreon.com/iceman1001/

[ Proxmark3 RFID instrument ]

MCU..... AT91SAM7S512 Rev A
Memory.... 512 KB ( 71% used )

Client.... Iceman/master/v4.19552-88-g9cdef9ceb 2024-12-11 21:35:09
Bootrom... Iceman/master/v4.19552-88-g9cdef9ceb-suspect 2024-12-11 21:35:07
OS..... Iceman/master/v4.19552-88-g9cdef9ceb-suspect 2024-12-11 21:35:12
Target.... PM3 GENERIC






[usb] pm3 --> |
```

Figure 4: Proxmark3 Terminal User Interface

As mentioned before, the software comes with several built-in tools. Since our university uses MIFARE cards, we mostly focus on that part of the software. Tools for MIFARE Classic cards can be seen by entering the `hf mf` command, which stands for high frequency MIFARE. As can be seen from the Figure 9, the attacks mentioned before are already available in the client software. For the sake of completeness, we investigate multiple ID cards that have been granted from METU with different validity dates and from different manufacturers. These cards are listed in Table 3. Even though we analyze ~30 id cards that are owned by different people, for the sake of keeping their identities anonymous, only the cards owned by the author are presented in this report.

We start our investigation by getting general information about the ID cards. In order to point out our motivation, we start with ID card number 1, most recent one. Proxmark3 Client immediately identifies the backdoor present in card number 1, with command `hf mf info`. The Figure 5 shows the general information about the card.

Table 3: List of Identification Cards

Card Number	Photo	Format Date	Vulnerabilities
1		02/10/2023	Fudan Backdoor Key Default Key Hard Nested
2		03/12/2021	Hard Nested
3		23/06/2021	Hard Nested Default Keys
4		??/03/2020	Nested Predictable Key
5		??/??/2017	Hard Nested Default Key Predictable Key

```
[usb] pm3 --> hf mf info

[=] --- ISO14443-a Information -----
[+] UID: 6D 54 07 60
[+] ATQA: 00 04
[+] SAK: 08 [2]

[=] --- Keys Information
[+] loaded 2 user keys
[+] loaded 61 hardcoded keys
[+] Backdoor key.... A396EFA4E24F
[+] Block 0.... 6D5407605E0804000337213D1F1AA090 | .?!=...

[=] --- Fingerprint
[+] Fudan FM11RF085

[=] --- Magic Tag Information
[=] <n/a>

[=] --- PRNG Information
[+] Prng..... weak
[+] Static enc nonce... yes

[usb] pm3 --> |
```

Figure 5: Proxmark3 Client Output for Card Number 1 with identified Fudan backdoor key

Before going into the details of the attack, it is worth mentioning the enumeration process of the overall attack. We started investigation of the authentication procedure by only copying the UID of the card. This can be done by copying the UID of the original card to a changeable UID (CUID) card. These cards are referred as Magic Gen2 cards and can be obtained from Chinese vendors from internet for ~2 USD. Sample CUID card can be seen from Figure 6.



Figure 6: Blank Card with Changeable UID. Purple Sticker is an Indicator.

These CUID cards are specifically produced to clone MIFARE cards. These cards mostly have backdoor keys with advanced properties, i.e. changeable UID. These cards respond differently to authentication protocol, highlighting its properties. The terminal output of ``hf mf info`` command for the ghost CUID card is given in Figure 7.

```
[usb] pm3 --> hf mf info
[=] --- ISO14443-a Information -----
[+] UID: 6A 13 B9 30
[+] ATQA: 00 04
[+] SAK: 08 [2]

[=] --- Keys Information
[+] loaded 2 user keys
[+] loaded 61 hardcoded keys
[+] Sector 0 key A... FFFFFFFFFF
[+] Sector 0 key B... FFFFFFFFFF
[+] Sector 1 key A... FFFFFFFFFF
[+] Block 0.... 6A13B930F00804006263646566676869 | bcdefghi

[=] --- Fingerprint
[+] Fudan based card

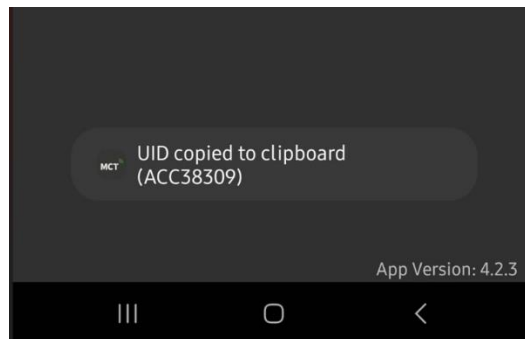
[=] --- Magic Tag Information
[+] Magic capabilities... Gen 2 / CUID

[=] --- PRNG Information
[+] Prng..... weak

[usb] pm3 --> |
```

Figure 7: Proxmark3 output for the Magic Gen2 ghost card

As can be seen from Figure 7, cards have `magic capabilities`, which means, its 4-byte or 7-byte UID value is changeable and capable of mimicking the MIFARE communication protocol without problems. With the same UID ghost card, we discovered that access control in the campus **only** depends on the UID value of the card. The same UID ghost card can open every door in the campus area like the original card. At this point, it is worth mentioning, UID value is not a secret value, in fact, in the MIFARE communication protocol, UID value must be sent in plaintext. One can read the UID value simply tapping the card back to an NFC activated phone as can be seen in Figure 8.



*Figure 8: Simple Android application that can automatically copy UID of the card in the proximity of the phone*

This is a vital security issue. Since one can carry a bag with a relatively large antenna and copy all the UIDs just walking by other people. As a result, we were able to copy UID and enter all the privileged areas. This also leads to an impersonation of the person who paired to the original UID. We continued copying the whole original card to a ghost card. As mentioned before, Proxmark3 Client software has pre-built-in attacks in the software. These attacks can be seen in the Figure 9.

```

[usb] pm3 --> hf mf
help          This help
list         List MIFARE history
-----
info         mfc card Info
isen        mfc card Info Static Encrypted Nonces
darkside    Darkside attack
nested      Nested attack
hardnested  Nested attack for hardened MIFARE Classic cards
stationested Nested attack against static nonce MIFARE Classic cards
brute       Smart bruteforce to exploit weak key generators
autopwn     Automatic key recovery tool for MIFARE Classic
nack        Test for MIFARE NACK bug
chk         Check keys
fchk        Check keys fast, targets all keys on card
decrypt     Decrypt Cryptol data from sniff or trace
supercard   Extract info from a `super card`
-----
authn       ISO14443-4 AES authentication
acl         Decode and print MIFARE Classic access rights bytes
dump        Dump MIFARE Classic tag to binary file
mad         Checks and prints MAD
personalize Personalize UID (MIFARE Classic EV1 only)
rdbl       Read MIFARE Classic block
rdsc       Read MIFARE Classic sector
restore     Restore MIFARE Classic binary file to tag
setmod     Set MIFARE Classic EV1 load modulation strength
value      Value blocks
view       Display content from tag dump file
wipe       Wipe card to zeros and default keys/acc
wrl        Write MIFARE Classic block
-----
sim         Simulate MIFARE card
ecfill     Fill emulator memory with help of keys from emulator
ecclr      Clear emulator memory
egetblk    Get emulator memory block
agetsec    Get emulator memory sector
ekeyprn    Print keys from emulator memory
eload      Upload file into emulator memory
esave     Save emulator memory to file
esetblk    Set emulator memory block
eview     View emulator memory
-----
cgetblk    Read block from card
cgetsc     Read sector from card
cload      Load dump to card
csave     Save dump from card into file or emulator
csetblk    Write block to card
csetuid    Set UID on card
cview     View card
cwipe     Wipe card to default UID/Sectors/Keys

```

Figure 9: Terminal Output of Proxmark3 Client on MIFARE Classic Commands

With this information at hand, we were able to break and clone the whole card. We can dump the whole card with command ``script run fm11rf08s_recovery.py`` which exploits the Fudan Backdoor vulnerability. The dumped information on the card can be seen in Figure 10.

We continued with copying the whole original card to a ghost card. We were able to pay with the ghost card. We enumerate further if we able to change the balance in the card. For that, we dump the card before paying with the card for lunch at the student cafeteria and compare the information in the dumps before and after. This comparison can be seen in the Figure 11.

sec	blk	data	ascii
0	0	6D 54 07 60 5E 08 04 00 03 37 21 3D 1F 1A A0 90	mT.`^....?!=....
	1	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	2	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	3	70 14 FB 3F 89 CE 78 77 88 FF D4 D0 7A 11 C6 2A	p..?.xw....z..*
1	4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	5	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	6	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	7	F1 D2 E1 C2 B1 A2 0A 55 AF FF A1 B2 C1 E2 D1 F2	.....U.....
2	8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	9	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	10	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	11	59 CB 7A 75 2B C9 78 77 88 FF 8F 37 EA 86 DD 16	Y.zu+.xw...7....
3	12	00 29 2C 83 01 00 00 00 00 00 00 00 00 00 00	.),.....!?=.....
	13	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	14	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	15	C9 5F D3 2E 72 8A 3A 57 8C FF B0 58 E2 D3 20 5B	...r.:W...X.. [
4	16	BC C4 01 00 43 3B FE FF BC C4 01 00 10 EF 10 EF	...C;.....!?=.....
	17	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	18	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	19	8B AE 47 BA 54 58 08 77 8F FF A5 20 49 D0 E8 D3	..G.TX.w... I...
5	20	30 30 31 01 00 00 00 E8 07 19 0C 01 00 00 00 00	001.....!?=.....
	21	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	22	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	23	76 6B F5 71 3D 5F FF 07 80 69 5F 3D 71 F5 6B 76	vk.q=...i_=q.kv
6	24	42 45 52 4B 41 59 00 00 00 00 00 00 00 00 00	BERKAY.....!?=.....
	25	41 4B 43 4F 52 45 4E 00 00 00 00 00 00 00 00	AKCOREN.....!?=.....
	26	55 4E 56 41 4E 20 59 4F 4B 00 00 00 00 00 00	UNVAN YOK.....!?=.....
	27	76 6B F5 71 3D 5F FF 07 80 69 5F 3D 71 F5 6B 76	vk.q=...i_=q.kv
7	28	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	29	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	30	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	31	9F 63 ED 5C 9A 6B 78 77 88 FF 8B 23 7C A2 41 DD	.c.\.kxw...# .A.
8	32	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	33	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	34	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	35	9F 63 ED 5C 9A 6B 78 77 88 FF 8B 23 7C A2 41 DD	.c.\.kxw...# .A.
9	36	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	37	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	38	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	39	9F 63 ED 5C 9A 6B 78 77 88 FF 8B 23 7C A2 41 DD	.c.\.kxw...# .A.
10	40	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	41	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	42	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	43	00 00 00 00 00 00 78 77 88 FF 8B 23 7C A2 41 DD	.....xw...# .A.
11	44	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	45	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	46	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	47	9F 63 ED 5C 9A 6B 78 77 88 FF 8B 23 7C A2 41 DD	.c.\.kxw...# .A.
12	48	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	49	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	50	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	51	9F 63 ED 5C 9A 6B 78 77 88 FF 8B 23 7C A2 41 DD	.c.\.kxw...# .A.
13	52	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	53	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	54	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	55	9F 63 ED 5C 9A 6B 78 77 88 FF 8B 23 7C A2 41 DD	.c.\.kxw...# .A.
14	56	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	57	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	58	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	59	9F 63 ED 5C 9A 6B 78 77 88 FF 8B 23 7C A2 41 DD	.c.\.kxw...# .A.
15	60	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	61	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	62	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....!?=.....
	63	9F 63 ED 5C 9A 6B 78 77 88 FF 8B 23 7C A2 41 DD	.c.\.kxw...# .A.

Figure 10: Dump of the card number 1

#	../../../../proxmark-dumps/backup-keys/hf-mf-6D540760-dump-004.bin	../../../../proxmark-dumps/backup-keys/hf-mf-6D540760-dump-008.bin
[=] 000	6D 54 07 60 SE 08 04 00 03 37 21 3D 1F 1A A0 90 mT.'^.....7!=-...	6D 54 07 60 SE 08 04 00 03 37 21 3D 1F 1A A0 90 mT.'^.....7!=-...
[=] 010	00 00	00 00
[=] 020	00 00	00 00
[=] 030	70 14 FB 3F 89 CE 78 77 88 FF D4 D0 7A 11 C6 2A p..?.xw....z.*	70 14 FB 3F 89 CE 78 77 88 FF D4 D0 7A 11 C6 2A p..?.xw....z.*
[=] 040	00 00	00 00
[=] 050	00 00	00 00
[=] 060	00 00	00 00
[=] 070	F1 D2 E1 C2 B1 A2 0A 55 AF FF A1 B2 C1 E2 D1 F2 .....U.....	F1 D2 E1 C2 B1 A2 0A 55 AF FF A1 B2 C1 E2 D1 F2 .....U.....
[=] 080	00 00	00 00
[=] 090	00 00	00 00
[=] 0A0	00 00	00 00
[=] 0B0	59 CB 7A 75 2B C9 78 77 88 FF 8F 37 EA 86 DD 16 Y.zu+.xw...7....	59 CB 7A 75 2B C9 78 77 88 FF 8F 37 EA 86 DD 16 Y.zu+.xw...7....
[=] 0C0	00 29 2C 83 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 29 2C 83 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[=] 0D0	00 00	00 00
[=] 0E0	00 00	00 00
[=] 0F0	C9 5F D3 2E 72 8A 3A 57 8C FF 00 58 E2 D3 20 58 .....r:W...X.. [	C9 5F D3 2E 72 8A 3A 57 8C FF 00 58 E2 D3 20 58 .....r:W...X.. [
[=] 100	04 29 00 00 FB D6 FF FF 04 29 00 00 10 EF 10 EF .....L.....	4C 1D 00 00 B3 E2 FF FF 4C 1D 00 00 10 EF 10 EF .....L.....
[=] 110	00 00	00 00
[=] 120	00 00	00 00
[=] 130	8B AE 47 BA 54 58 08 77 8F FF A5 20 49 D0 E8 D3 ..G.TX.w... I..	8B AE 47 BA 54 58 08 77 8F FF A5 20 49 D0 E8 D3 ..G.TX.w... I..
[=] 140	30 30 31 01 00 00 00 E8 07 1E 01 01 00 00 00 0001.....	30 30 31 01 00 00 00 E8 07 05 00 01 00 00 00 0001.....
[=] 150	00 00	00 00
[=] 160	00 00	00 00
[=] 170	76 68 F5 71 3D 5F FF 07 80 69 5F 3D 71 F5 68 76 vk.q=...i=q.kv	76 68 F5 71 3D 5F FF 07 80 69 5F 3D 71 F5 68 76 vk.q=...i=q.kv
[=] 180	42 45 52 4B 41 59 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	42 45 52 4B 41 59 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[=] 190	41 4B 43 4F 52 45 4E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	41 4B 43 4F 52 45 4E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[=] 1A0	55 4E 56 41 4E 20 59 4F 4B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	55 4E 56 41 4E 20 59 4F 4B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Figure 11: Comparison of the same card before and after lunch payment

As can be seen from the Figure 11, only the values in the data blocks 16 and 20 have changed. With further enumeration, we point out that, data block 16 contains the money balance in 100\*TRY in little-endian hexadecimal format. The first 4-byte is the balance value and second 4-byte value, and third 4-byte value is its binary complement and the same value respectively. The purpose of the data in block 20 is currently unknown. Further enumeration is needed but we suspect that it is related to information about whether if student is paid for today’s meal or not, since students are not allowed to eat multiple times at the student cafeteria. We changed the balance in the card and were able to pay for lunch with ghost card. We successfully cloned the card and manipulated the balance written in the card.

We further enumerate the system and find out how kiosk machines at student cafeteria interact with ID cards. The backend system does not permit users to load balance higher than 800 ~TRY. If we manipulate the balance value in ID cards higher than ~2000 TRY, it does not authenticate in the payment turnstile. However, if we kept the balance lower than ~2000 TRY, we were able to pay. Strangely, if the balance on the ID card is higher than the balance in the backend database, the software behind the payment process overwrites the value in the database with the balance value in the ID card.

We manipulated the balance field with various ghost cards and original cards. For the sake of completeness, we include some pictures of the cards in this report. One can see a ghost card with more than a million TRY in balance in Figure 12.

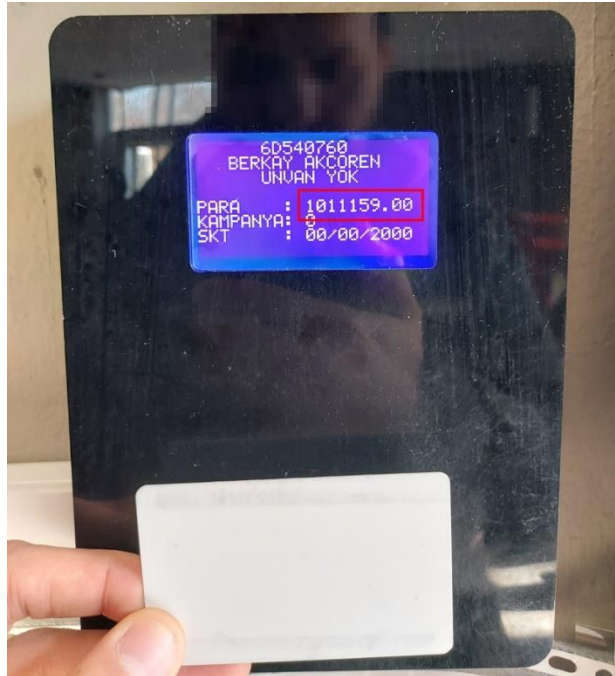


Figure 12: Ghost card with a balance more than million TRY

As we mention before, one cannot authenticate with this ghost card with more than ~2000 TRY in balance. However, we can create a ghost key fob with eligible amount of balance and able to pay with it. This can be seen in Figure 13.



Figure 13: Picture of the verified payment with a ghost key fob

We found another critical issue when we enumerate the number 3 ID card. The general information about the card that is identified by the Proxmark3 Client is given in Figure 14. Note that, this card has no Fudan backdoor key. ID card number 3 is an academic personal ID card with different privileges and from a different vendor. This card also has the **exact same keys**.

```
[usb] pm3 --> hf mf info

[=] --- ISO14443-a Information -----
[+] UID: AC C3 83 09
[+] ATQA: 00 04
[+] SAK: 08 [2]

[=] --- Keys Information
[+] loaded 2 user keys
[+] loaded 61 hardcoded keys

[=] --- Fingerprint
[=] <n/a>

[=] --- Magic Tag Information
[=] <n/a>

[=] --- PRNG Information
[#] Static nonce..... 01200145
[+] Static nonce..... yes
```

Figure 14: General information about the ID card number 3

Also, the cards we found around the campus have the same keys. Therefore, **ALL the ID cards** in our university share the exact same secret keys. This is a vital security flaw. The keys we found during this assessment are given in Figure 14.

F1D2E1C2B1A2	Default Key	7014FB3F89CE	FAFBFCFFFEFD	Default Key
A1B2C1E2D1F2	Default Key	D4D07A11C62A	416E4B615261	AnKaRa
766BF5713D5F		59CB7A752BC9	5C8FF9990DA2	
5F3D71F56B76		8F37EA86DD16	D01AFEEB890A	
		C95FD32E728A	75CCB59C9BED	
		B058E2D3205B	4B791BEA7BCC	
		8BAE47BA5458	4CA20B875FFE	
		A52049DOE8D3	44EE310D4D4E	
		9F63ED5C9A6B		
		8B237CA241DD		

Figure 15: Secret keys obtained from the METU ID cards. Left column is used only in the new cards. Middle column is used in both old and new ID cards. Right column is only used in the old card. Default keys are indicated. One of the keys has also a meaning AnKaRa in ASCII format, which makes it a predictable key.

Another important finding was, if we change another value block in the card, which is used in the authentication process, such as name, surname or title, the kiosk machines overwrite the correct values to the card, by ONLY checking the UID. In other words, if we change the UID value of the ghost card and insert it into the kiosk machine, the kiosk machine overwrites the value blocks in the ghost card with the values of the original card with corresponding UID. Hence, we can impersonate that person by simply knowing their cards UID.

There are differences in the value blocks that are non-zero, when we compare first and third ID cards. Further enumeration is needed to identify the meaning of them. However, we suspect that they are related to title and registered departments.



## CHAPTER 4

### 4. CONCLUSION

Smartcard systems play a crucial role in everyday life. Due to the critical services provided by them, security of smartcard systems is important. In this project at hand, we analyze the security of the smartcard system at Middle East Technical University. We conduct practical attacks in the literature and able to clone any ID card in the campus area. We also found critical design flaws in the access control and payment processes. The most critical security flaw is all ID cards in the campus area share the same secret keys. We show that, a dedicated attacker can easily craft a ghost card with infinite amount of balance for dishonest payment and can impersonate anyone around the campus area with a minimal effort.

MIFARE Classic cards are notoriously known for their security flaws. It is strongly advised that; companies or individuals should migrate to more secure options if they are using MIFARE Classic.

I would like to express my sincere gratitude to Mikail Yılmaz, a fellow RFID security researcher, for his invaluable support and insightful discussions on smartcards and MIFARE Classic card exploits. His curiosity and passion for smartcard security research greatly contributed to the project at hand. I wish him all the best in his career.

## References

- [1] Type-A, ISO/IEC 14443, *Identification cards – Contactless integrated circuit cards – Proximity cards*, 2018.
- [2] K. Nohl, D. Evans, Starbug and H. Plötz, "Reverse-Engineering a Cryptographic RFID Tag," in *17th USENIX Security Symposium (USENIX Security 08)*, San Jose, CA, 2008.
- [3] F. D. Garcia, G. de Koning Gans, R. Muijrrers, P. van Rossum, R. Verdult, R. W. Schreur and B. Jacobs, "Dismantling MIFARE Classic," in *Computer Security - ESORICS 2008*, Berlin, Heidelberg, 2008.
- [4] F. D. Garcia, P. van Rossum, R. Verdult and R. W. Schreur, "Wirelessly Pickpocketing a Mifare Classic Card," in *2009 30th IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2009.
- [5] C. Tezcan, "Brute Force Cryptanalysis of MIFARE Classic Cards on GPU," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy - Volume 1: ICISSP*, 2017.
- [6] N. T. Courtois, "THE DARK SIDE OF SECURITY BY OBSCURITY - and Cloning MiFare Classic Rail and Building Passes, Anywhere, Anytime," in *Proceedings of the International Conference on Security and Cryptography (ICETE 2009) - SECRIPT*, 2009.
- [7] C. Meijer and R. Verdult, "Ciphertext-only Cryptanalysis on Hardened Mifare Classic Cards," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2015.
- [8] P. Teuwen, "MIFARE Classic: exposing the static encrypted nonce variant," *Cryptology ePrint Archive, Paper 2024/1275*, 2024.
- [9] Iceman, "RFID Research Discord Community," [Online]. Available: <https://discord.com/invite/iceman>. [Accessed 07 01 25].
- [10] Iceman, "Proxmark3 Client Github Page," [Online]. Available: <https://github.com/RfidResearchGroup/proxmark3>. [Accessed 7 1 2025].

- [11] Y. Liu, D. Gu, B. Li and B. Qu, "Legitimate-reader-only attack on MIFARE Classic," *Mathematical and Computer Modelling*, vol. 58, no. 1-2, pp. 219-226, 2013.
- [12] Iceman, "Proxmark3 Iceman Fork," [Online]. Available: <https://github.com/RfidResearchGroup/proxmark3>. [Accessed 30 12 2024].
- [13] G. de Koning Gans, J.-H. Hoepman and F. D. Garcia, "A Practical Attack on the MIFARE Classic," in *Smart Card Research and Advanced Applications*, Heidelberg, Berlin, 2008.
- [14] Y.-H. Chiu, W.-C. Hong, L.-P. Chou, J. Ding, B.-Y. Yang and C.-M. Cheng, "A Practical Attack on Patched MIFARE Classic," in *Information Security and Cryptolog*, Cham, 2014.