



How I bricked my Istanbulkart

@0xabc0

Table of Contents

Ixtanbulkart App

Mifare Desfire EV1

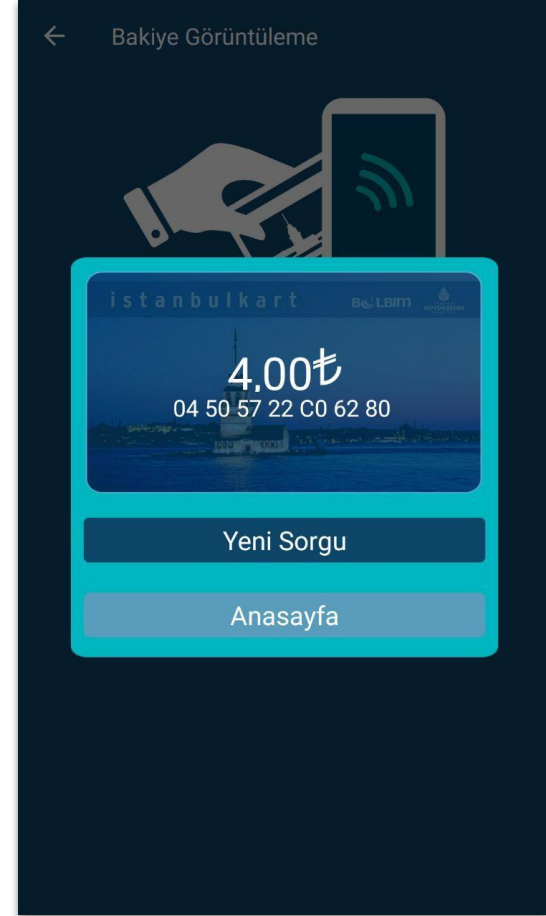
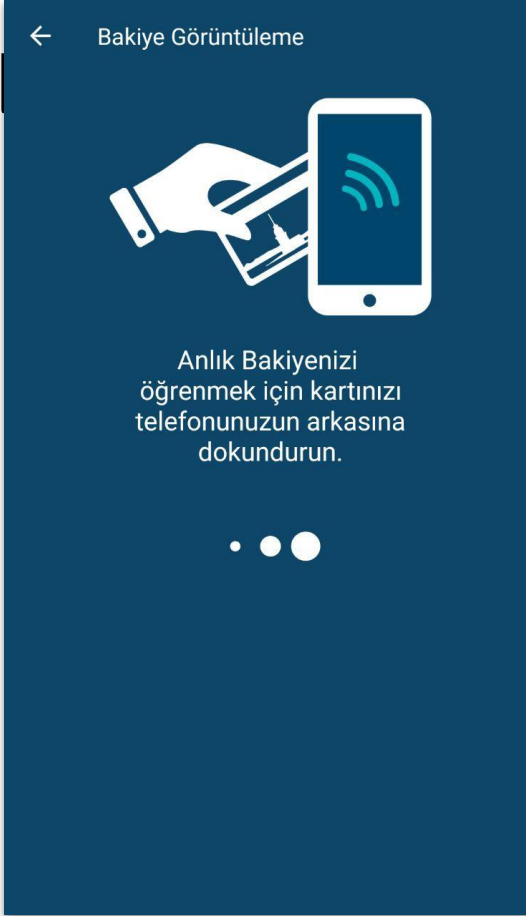
Proxmark/Frida



Ixtanbulkart



Ixtanbulkart



Ixtanbulkart



Istanbulkart



Ixtanbulkart

Flow of the app:

1. Receive commands from server, send it to card
2. Send response of the card to server
3. Go to 1

Ixtanbulkart

App Traffic:

1. Web
2. NFC

Ixtanbulkart

App Traffic:

1. Web - SSLPinning
2. NFC

Ixtanbulkart

App Traffic:

1. Web - SSLPinning | Bypass : remove pin,rebuild
2. NFC

Ixtanbulkart

App Traffic:


1. Web - SSLPinning | Bypass : remove pin, rebuild
2. NFC - hook iso.transceive with Frida

Ixtanbulkart

App Traffic:

1. Web - SSLPinning | Bypass : remove pin, rebuild
2. NFC - hook iso.tranceive with Frida
3. listen adb logcat xD (grep "Retrofit")

Boring Stuff

1. What is Desfire anyway ?
2. How it works ?
3. RTFM but where is the documentation ? 

Mifare Desfire EV1

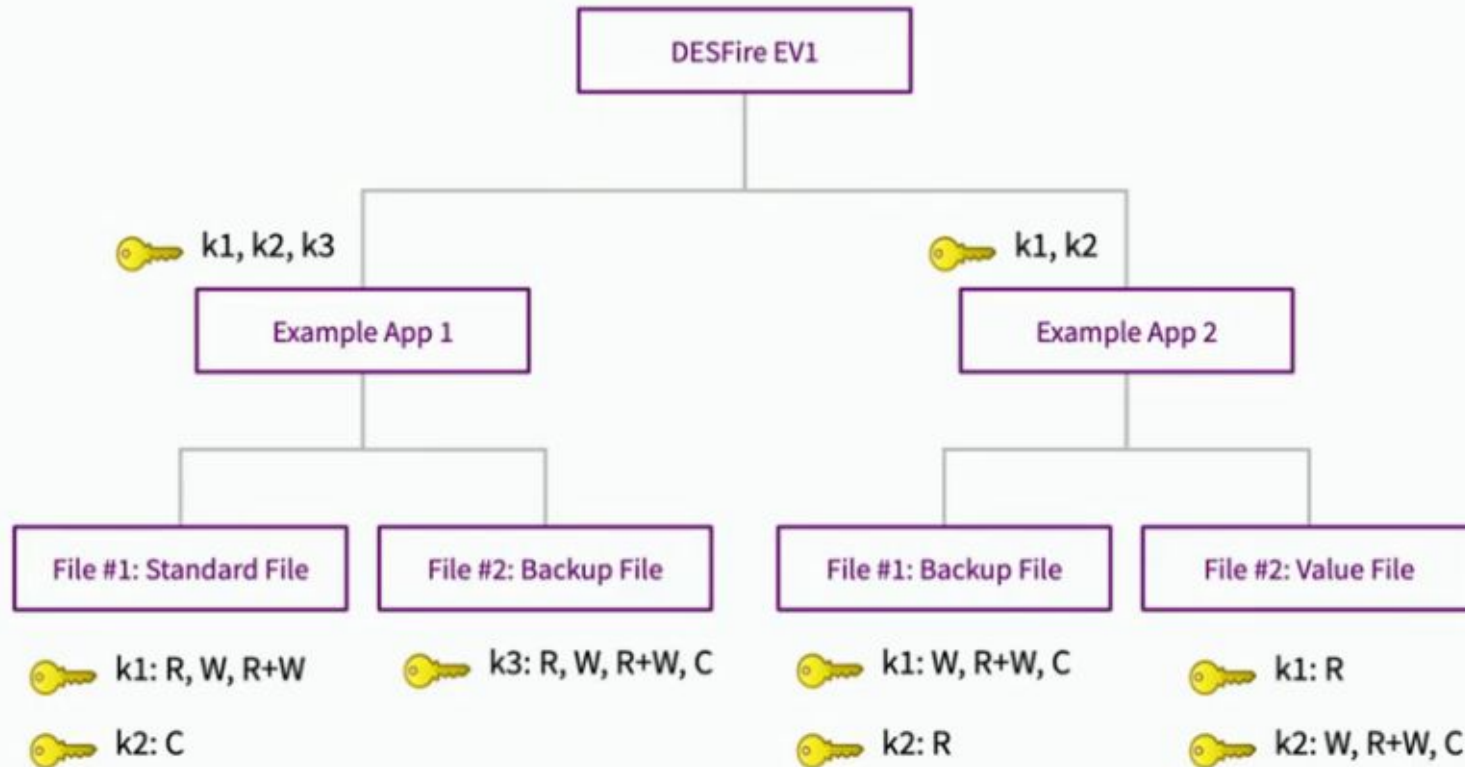
Authentication : Card and Reader make 3 way handshake

Reader proof that it knows the key without sending it.(magic)

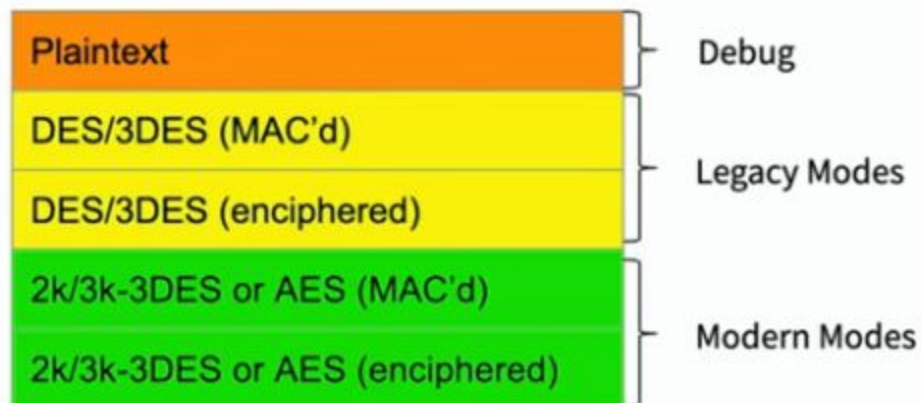
Card >> Nonce(1) | Reader >> Nonce(2) | Card >> Auth1

After auth, communication continue on mode of the command OR file

A cryptographically authenticated filesystem



DESFire EV1 Communication Modes



 Used for debugging  Used by Leap Card  Used by Oyster Card

Desfire Commands

90		00	00		00	
CLS	INS	P1	P2	Lc	Data	Le

Response Codes

9100 : OK

91AF : Additional Frame, More data can be send or received

91AE : Authentication error

Apps and Files

- GetApplicationIds : 0x6A
- Select App : 0x5A <app no>
- GetFileIds : 0x6F
- GetFileSettings : 0xF5 <filenumber>

File Types and Communication Modes

```
+-----+
|FileType|Code|
+-----+
|Standart|0x00|
+-----+
|Backup  |0x01|
+-----+
|Value   |0x02|
+-----+
|Linear  |0x03|
+-----+
|Cyclic  |0x04|
+-----+
```

```
+-----+
|Comm Mode|Value|
+-----+
| Plain   | 0x0 |
+-----+
| Plain.MAC| 0x1 |
+-----+
| 3DES enc| 0x3 |
+-----+
```

File Access Permissions

15	12	11	8	7	4	3	0
Read Access		Write Access		Read&Write		Change Access Rights	

16 different value:

maximum 14 key: Value of each byte corresponds to assigned key

14 (0xE) means free access

15 (0xF) means no access

File Access Permissions Example:

```
>> 5A 42 22 01          // SELECT APP 42 22 01

<< 00                   // OK, IM COOL WITH THAT

>> F5 01                 // GET FILE SETTINGS 01

<< 00 01 01 21E2 200000  // Resp 1,Type 1,Comm 1, Acc 2,Fsize 3

E2 21
```

File Access Permissions Example:

```
>> 5A 42 22 01          // SELECT APP 42 22 01

<< 00                   // OK, IM COOL WITH THAT

>> F5 01                 // GET FILE SETTINGS 01

<< 00 01 01 21E2 200000  // Resp 1,Type 1,Comm 1, Acc 2,Fsize 3

E2 21                   R*,W2,RW2,C1
```

Communication Modes

Mode		Description
<code>CommMode.Plain</code>	<code>0x0</code>	message is transmitted in plaintext
<code>CommMode.MAC</code>	<code>0x1</code>	protection for integrity and authenticity
<code>CommMode.Full</code>	<code>0x2</code>	Full protection for integrity, authenticity and confidentiality, also referred to as "FullProtection" mode

Communication Example

>> 5A 42 22 01 // SELECT APP 42 22 01

<< 00

>> F5 01 // GET FILE SETTINGS 01

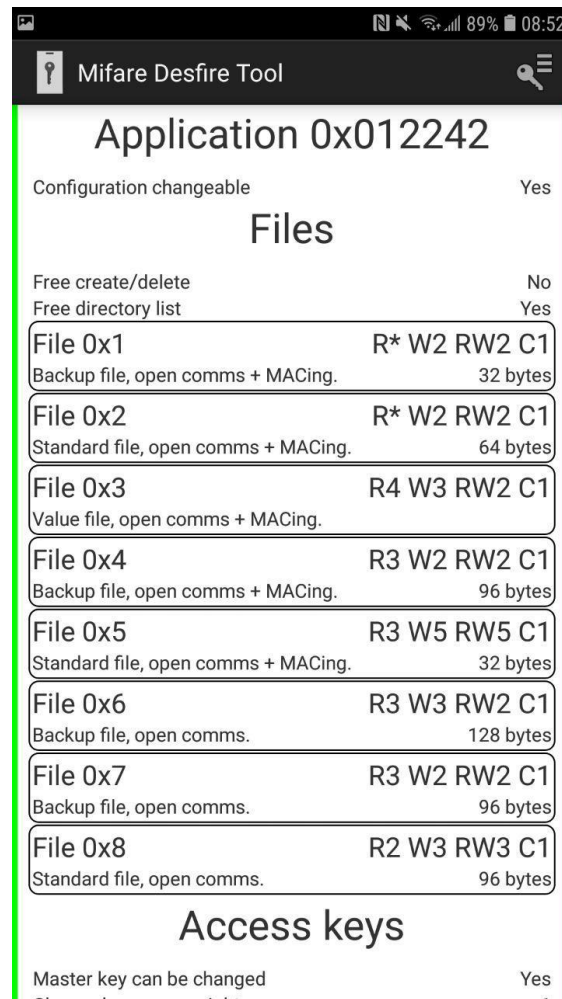
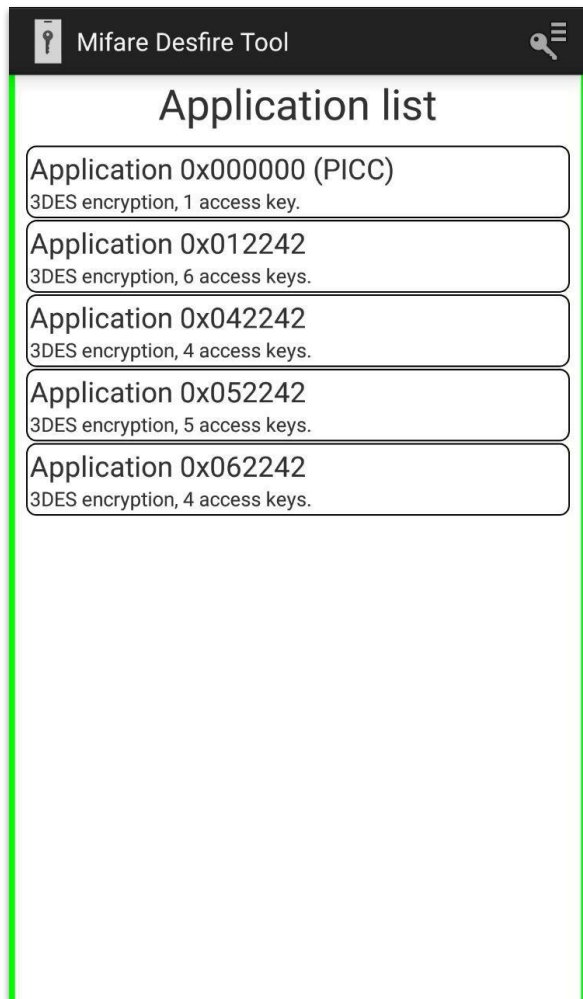
<< 00 01 01 21E2 200000 // Resp 1,Type 1,Comm 1, Acc 2,Fsize 3

01 Means DES/3DES MAC


Communication Modes

The specified communication mode is applied if there is an active authentication regardless of whether this authentication is required by the command or not.

At file level, the communication mode is defined by the file

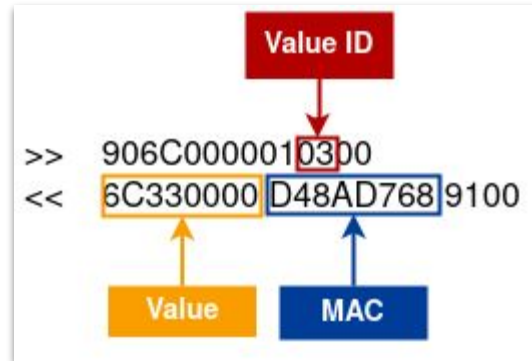


Auth : 0x0A



```
>> 900A0000 01 02 00
<< 5D923084ABC454F0 91AF
>> 90AF000010 CD6A463D360D5AE56D57635196ECE6E9 00
<< 96ACE0D3CF7DB36 9100
```

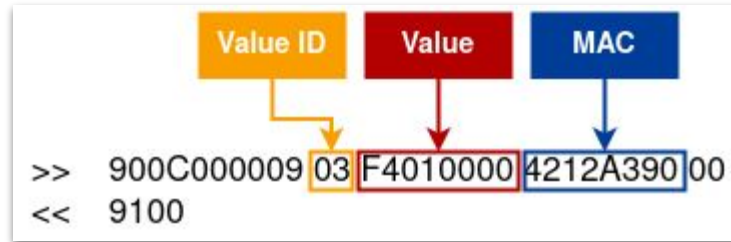
Get Value : 0x6C



Credit : 0x0C

Add value (4byte) to stored value (4byte) LittleEndian

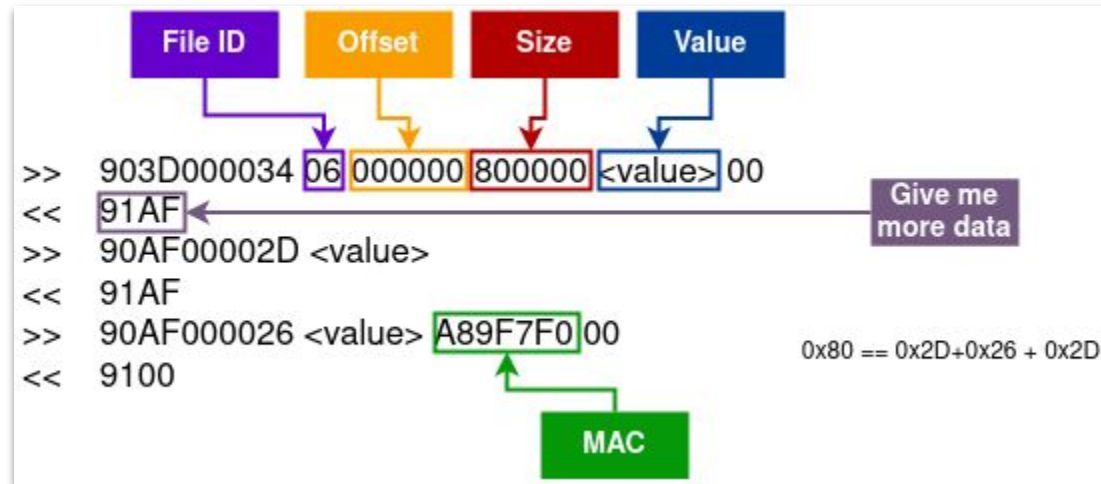
Value += a



Read Data : 0xBD



Write Data : 0x3D



Lets talk about Security

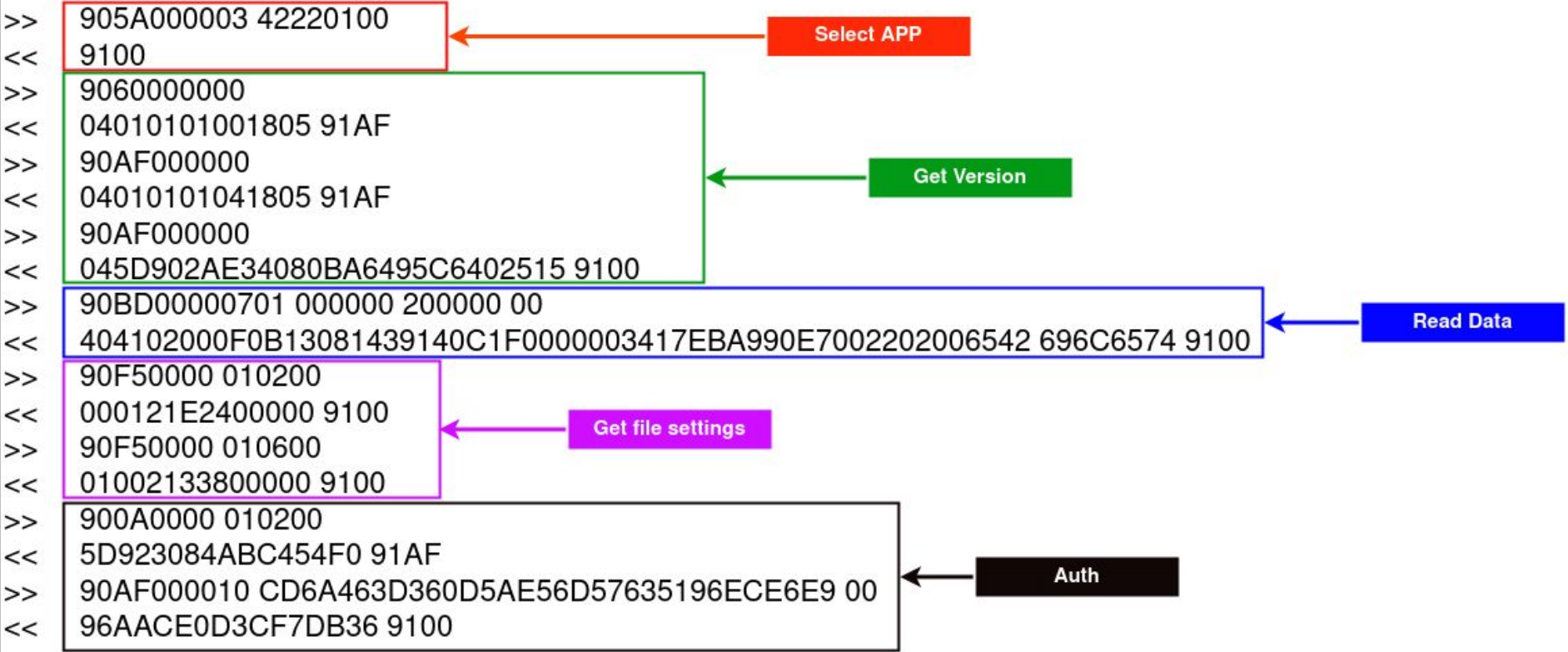
Previous Works (including Mifare Classic)

- Mostly focused on breaking card authentication without knowing keys
 - Bruteforcing keys
 - Darkside
 - Nested
 - Hardnested
- But these vectors work on Mifare Classic
- No vuln is published for Desfire (other than theories)
- But does it stop us ? Nope :=)

Ixtanbulkart : Flow

- MBSearchCardWithAuthP1
 - Create Transaction with 0 amount (DB Transaction)
- AuthP1
- AuthP2
 - Auth complete
- ReadCardResult
 - If any Instruction; WriteCardResult

Ixtanbulkart : Authentication



Ixtanbulkart : Authentication

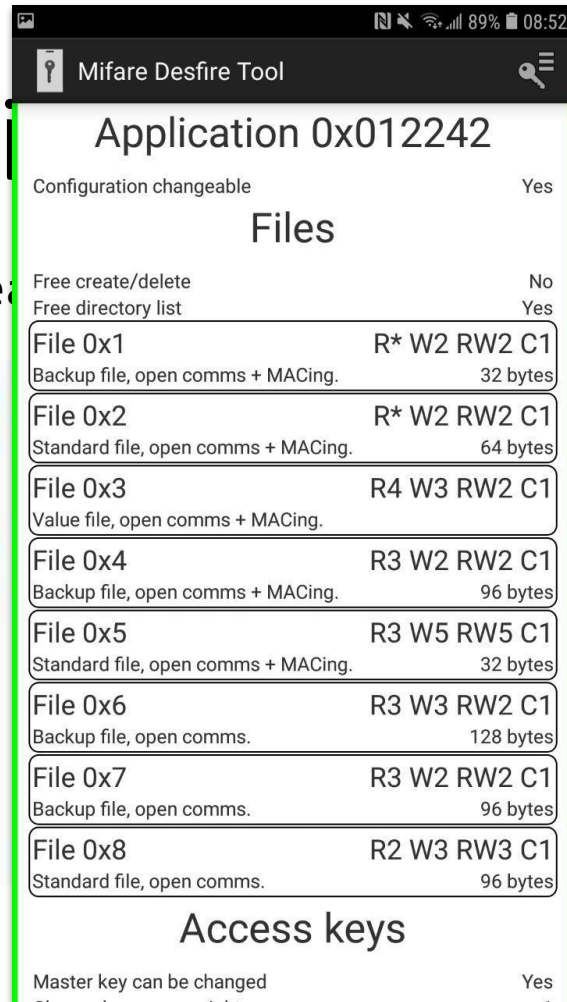
No auth session means we can't read file that needs authentication

```
>>>> 90BD000007 01 000000 20000000
<<<< 40 01 02 00 14 02 04 0C 17 3B 14 02 04 0C 17 3B 34 17 EB 9A 05 7F
91 00 - OK
>>>> 90BD000007 02 000000 32000000
<<<< 63 70 60 03 14 45 35 37 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 91 00
91 00 - OK
>>>> 906C0000 01 03 00
<<<< 91 AE
91 ae - Authentication status does not allow the requested command
```

Ixtanbulkart : Authentication

No auth session means we can't read

```
>>>> 90BD000007 01 000000 20000000
<<<< 40 01 02 00 14 02 04 0C 17 3B
91 00 - OK
>>>> 90BD000007 02 000000 32000000
<<<< 63 70 60 03 14 45 35 37 00 00
00 00 00 00 00 00 00 00 00 00 91 00
91 00 - OK
>>>> 906C0000 01 03 00
<<<< 91 AE
91 ae - Authentication status does
```



Authentication

17 EB 9A 05 7F

00 00 00 00 00

ed command

Read Card Values

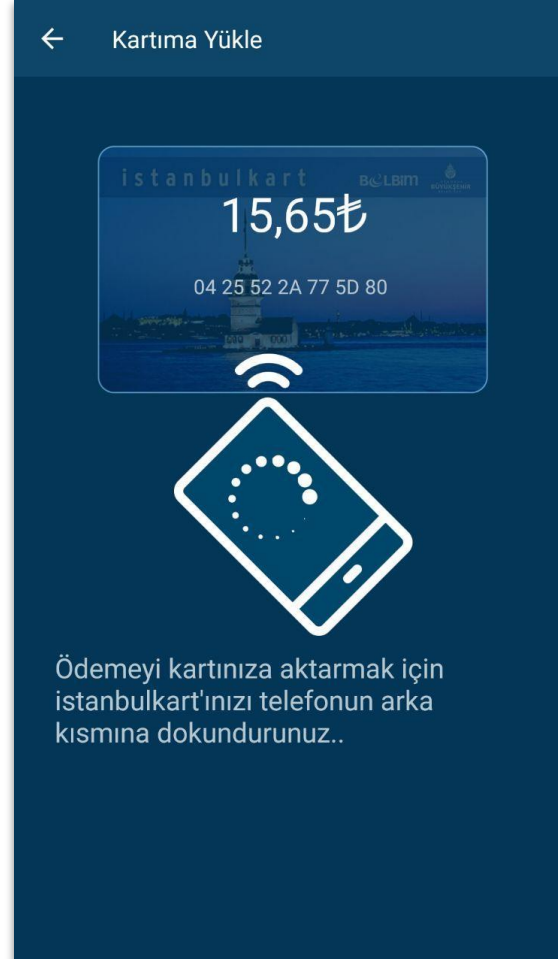
```
>> 90BD000007 01 000000 20000000
<< 404102000F0B13081439140C1F0000003417EBA990E7002202006542696C6574AB324107 9100
>> 90BD000007 02 000000 32000000
<< 637060007441362260021A00412E42696C616C2043616E0..0 87E8270A 9100
>> 90BD000007 02 320000 0E000000
<< 0..0 234C3EF8 9100
>> 906C000001 0300
<< 20160000 B68B8B58 9100
>> 90BD000007 04 000000 32000000
<< 8200010114030200010014040212050BCD00C80..0 2724AA9F 9100
>> 90BD000007 04 320000 2E000000
<< 0..0 15856939 9100
>> 90BD000007 06 000000 32000000
<< 1B0..02B9080000CA00030000050014030611252E788800370..014030611332204 9100
>> 90BD000007 06 320000 32000000
<< A00018A9B361000000DE5F5F432D313837396E14000000000000000030..0100FE1455F00300 9100
>> 90BD000007 06 640000 1C000000
<< 0000E705000014030610383B0..0 36570F08 9100
>> 90BD000007 07 000000 32000000
<< 0114030610383BB6080000CA000000000000000000000000000000220..018200010101F40100 9100
>> 90BD000007 07 320000 2E000000
<< 00000000000010101000000F40100007E170000010101000000000010270..0 9100
```

Read Data

Read Value

Read Data

Transfer (?) to Card



```
>> 903D000034 06 000000 800000 1B0..010002BA080000CB000000000050014030611252E78880037000000000000000140300
<< 91AF
>> 90AF00002D 0611332204A00018A9B361000000DE5F5F432D313837396E14000000000000000030..0
<< 91AF
>> 90AF000026 00000100C0688FF103000000E705000014030814092A0000000000000000000000000000 A89F7F0 00
<< 9100
>> 900C000009 03 F4010000 4212A390 00
<< 9100
>> 903D000034 07 000000 600000 0114030814092ABA080000CB00030..0220..0182000100
<< 91AF
>> 90AF00002D 0101F4010000000000000010101000000F4010000141800000101010000000000004290..0
<< 91AF
>> 90AF000006 0000000000000000
<< 9100
>> 90C7000000
<< 9100
>> 906C000001 03 00
<< 6C330000 D48AD768 9100
>> 90BD000007 06 5E000022000000
<< C0688FF103000000E705000014030814092A0000000000000000000000000000 A89F7F0 9100
>> 905A000003 42220100
<< 9100
```

Write Data

Credit

Write Data

Commit Transaction

Read Value

Read Data

Deselect App

Overall File Structure

Block 1 :	App id, App name
Block 2:	Information about card's holder
Block 3:	Balance
Block 6:	??? (Protected with MAC at the end)
Block 7:	Last transaction

Vulns ?

Because of the mode :

Credit Command can be replayed. IV is set to 0 => MAC is same

In modern mode, IV is set by CommandCounter

File Access:

File 6 and 7 can be written without MAC (plain mode)