Hacking the NFC cards for fun and honor degrees

Ricardo J. Rodríguez

All wrongs reversed

rjrodriguez@fi.upm.es ※ @RicardoJRdez ※ www.ricardojrodriguez.es

Universidad Politécnica de Madrid
Madrid, Spain

November 15, 2013

Universidad de Zaragoza
Zaragoza (España)

$whoami

- CLS member since early beginnings (2001)
- Ph.D. by University of Zaragoza (2013)
- Working for Technical University of Madrid

$whoami

- CLS member since early beginnings (2001)
- Ph.D. by University of Zaragoza (2013)
- Working for Technical University of Madrid
  - Performance analysis of complex systems
  - Secure software engineering
  - Fault-Tolerant systems (design and analysis)
  - Malware analysis (techniques and relative stuff)
  - Safety analysis in component-based systems

$whoami

- CLS member since early beginnings (2001)
- Ph.D. by University of Zaragoza (2013)
- Working for Technical University of Madrid
  - Performance analysis of complex systems
  - Secure software engineering
  - Fault-Tolerant systems (design and analysis)
  - Malware analysis (techniques and relative stuff)
  - Safety analysis in component-based systems
- Trainee at NcN, RootedCON, HIP...
- Speaker at NcN, HackLU, RootedCON, STIC CCN-CERT, HIP...

$whoami

- CLS member since early beginnings (2001)
- Ph.D. by University of Zaragoza (2013)
- Working for Technical University of Madrid
  - Performance analysis of complex systems
  - Secure software engineering
  - Fault-Tolerant systems (design and analysis)
  - Malware analysis (techniques and relative stuff)
  - Safety analysis in component-based systems
- Trainee at NcN, RootedCON, HIP...
- Speaker at NcN, HackLU, RootedCON, STIC CCN-CERT, HIP...
- Not an NFC (or RFID) expert!

$whoami

- CLS member since early beginnings (2001)
- Ph.D. by University of Zaragoza (2013)
- Working for Technical University of Madrid
  - Performance analysis of complex systems
  - Secure software engineering
  - Fault-Tolerant systems (design and analysis)
  - Malware analysis (techniques and relative stuff)
  - Safety analysis in component-based systems
- Trainee at NcN, RootedCON, HIP...
- Speaker at NcN, HackLU, RootedCON, STIC CCN-CERT, HIP...
- Not an NFC (or RFID) expert!

Not giving any new 0-day or vulnerability, just recalling the state-of-the-art

Outline

Outline

Near Field Communication: What is it? (I)

Near Field Communication (NFC)

- Standard to establish radio communication between devices
  - By touching or bringing them into close proximity
- Builds upon RFID
  - Radio-Frequency ID: identify and track (things/animals/people) using radio waves
  - Works at 13.56MHz band on ISO/IEC 18000-3 (no license needed)
- Distance needed: $\leq$ 10cm (theoretically $\leq$ 20)
- Rates: $106 - 424$ kbit/s
- Two main actors
  - Initiator: generates a RF field
  - Target
- Two working modes
  - Passive: initiator device provides a carrier field. Target is a transponder
  - Active: initiator + target generate their own fields

Near Field Communication: What is it? (II)

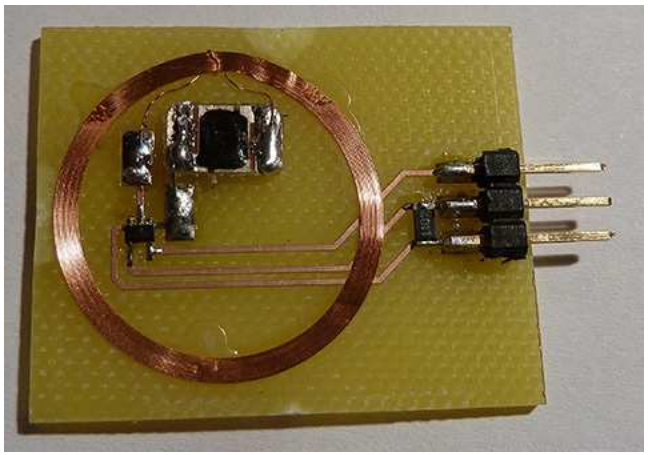
"Big" actors

NFC Forum

- Non-profit industry association
- Formed on March 18, 2004
- Founders: NXP Semiconductors (formerly Philips Semiconductors), Sony and Nokia
- Promotes implementation and standardisation of NFC
- 190 member companies (June 2013). Some located at Spain:
  - Applus
  - AT4 Wireless

Near Field Communication: What is it? (III)

Real actors (1)

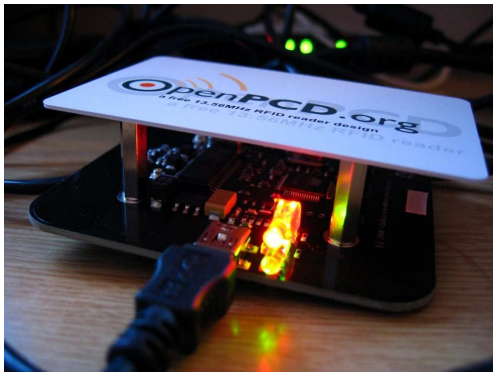


PICC

- Proximity Integrated Circuit Card
- Commonly named as *tag*
- Passive or active (depends on power supply)
  - Widely used (cheaper): passive ones
- It contains:
  - Internal capacitor
    - Stores the energy coming from the reader
  - Resistor

Near Field Communication: What is it? (III)

Real actors (2)

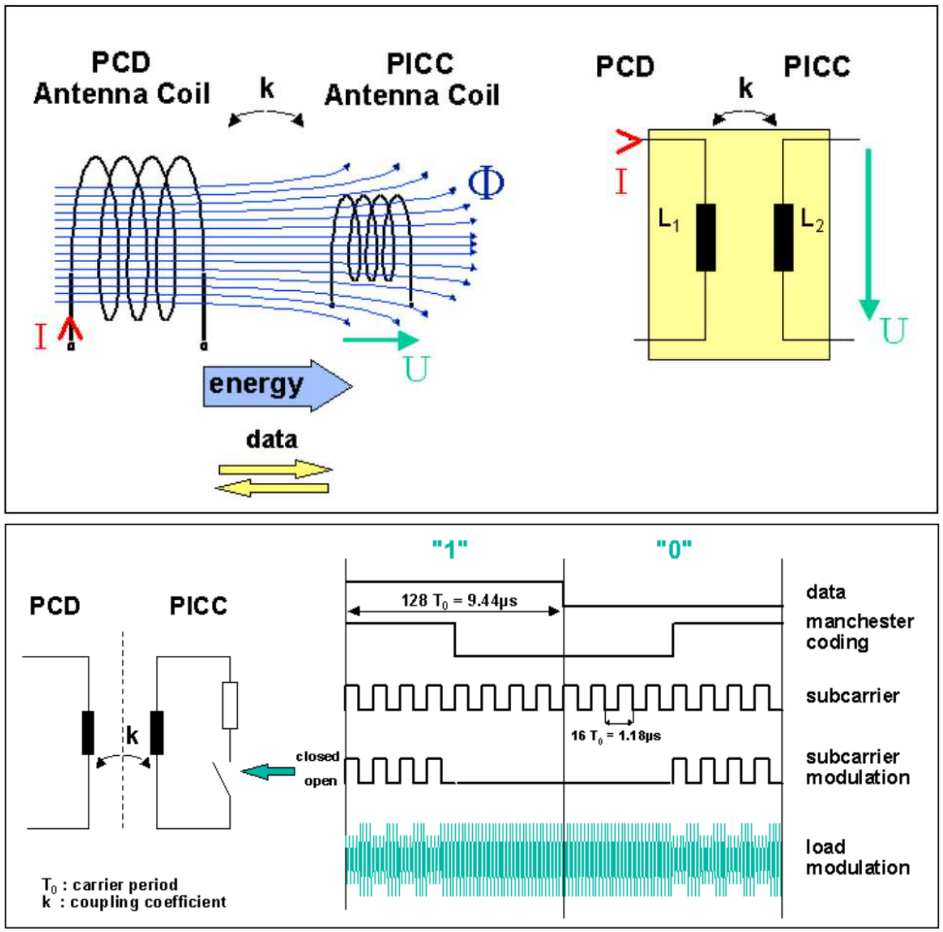


PCD

- Proximity Coupling Device
- Commonly named as *reader/writer*
- Active (forced)
- Contains the antenna
  - Communication at the 13.56MHz ($\pm$7kHz) frequency
  - Electronic field

Near Field Communication: What is it? (IV)

An interesting reading on this topic...



[Taken from 13.56 MHz RFID Proximity Antennas (`http://www.nxp.com/documents/application_note/AN78010.pdf`)]

Near Field Communication: Where is it used? (V)

Outline

MIFARE Classic (I): What is it?

MIFARE product family

- Introduced in 1995 by NXP
- "Advanced technology for RFID identification"
- Based on ISO/IEC 14443 Type A 13.56 MHz standard
- Several products:
  - Ultralight
  - Classic
  - DESFire
  - SmartMX

MIFARE Classic (I): What is it?

MIFARE product family

- Introduced in 1995 by NXP
- "Advanced technology for RFID identification"
- Based on ISO/IEC 14443 Type A 13.56 MHz standard
- Several products:
  - Ultralight
  - Classic
  - DESFire
  - SmartMX
- 50M reader and 5B card components sold
- $\sim$ 80% contactless ticketing credentials (according to ABI Research)

MIFARE Classic (II): Some of its common uses

Some systems using MIFARE Classic

- Access Controls
  - University of Zaragoza
  - Personal entrance Schiphol Airport (AMS)
  - Dutch military bases
  - Hotel room keys
  - Many office and official buildings
- Ticketing events
- Public transport systems
  - OV-Chipkaart (NL)
  - Oyster card (London, UK)
  - Smartrider (AU)
  - EMT (Málaga, Spain)
  - Wikipedia: `http://en.wikipedia.org/wiki/MIFARE`

MIFARE Classic (III): Internal Structure (1)

Logical Structure

- EEPROM memory
- Basic unit: 16B block
- A sector is a set of blocks
- Two size variants:
  - 1KB (16 sectors, 4 blocks each)
  - 4KB (40 sectors, first 32 sectors are 4-block, the rest 16-block)

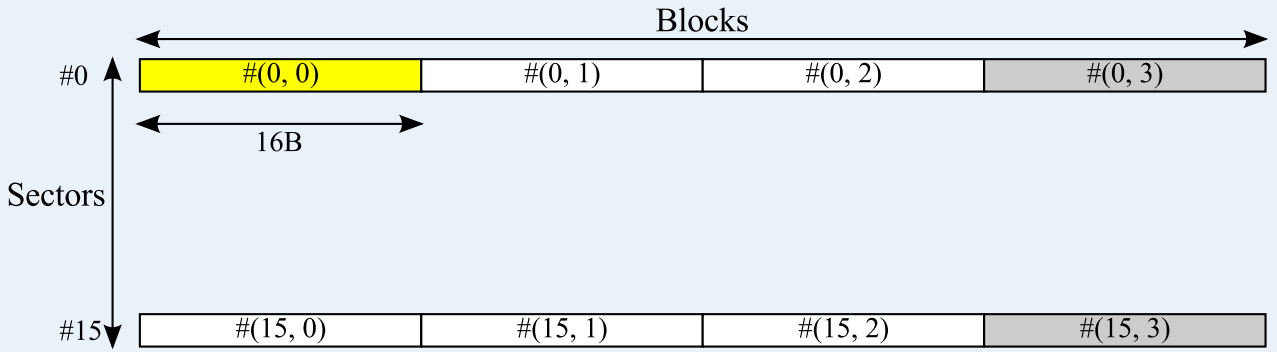
MIFARE Classic (III): Internal Structure (1)

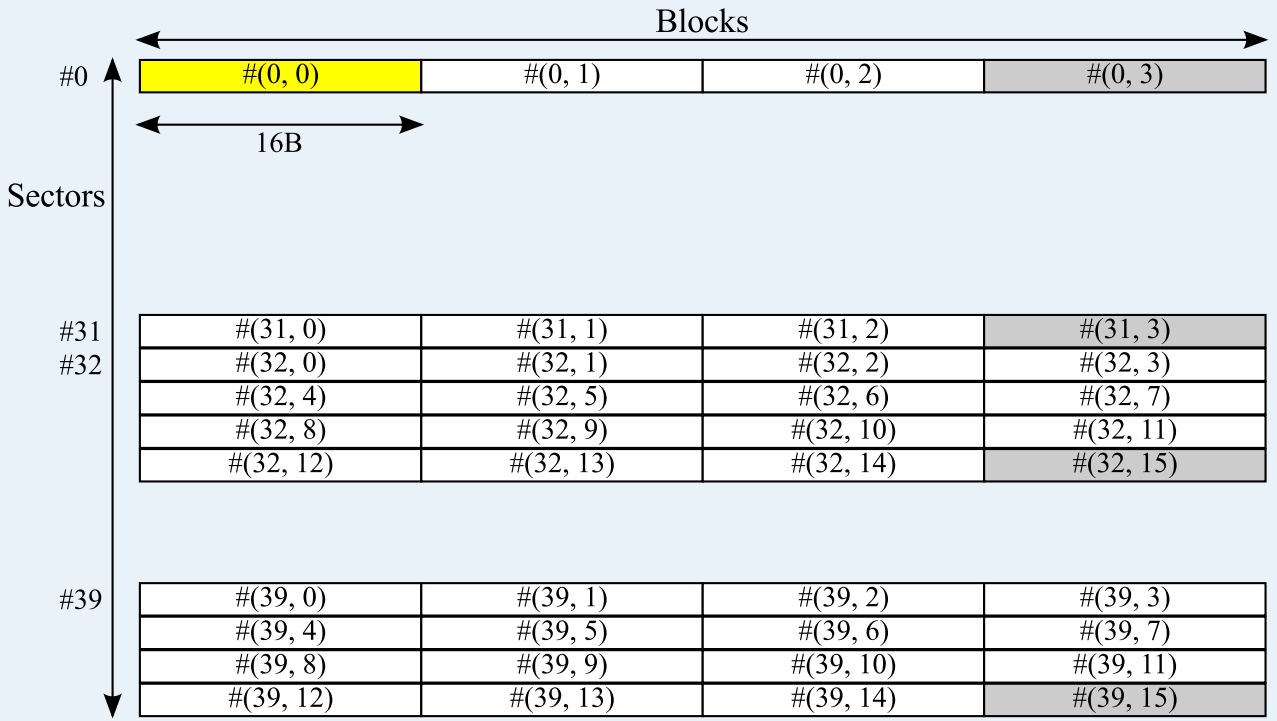
Logical Structure

- EEPROM memory
- Basic unit: 16B block
- A sector is a set of blocks
- Two size variants:
  - 1KB (16 sectors, 4 blocks each)
  - 4KB (40 sectors, first 32 sectors are 4-block, the rest 16-block)

Let me show you this graphically...

MIFARE Classic (III): Internal Structure(2)





MIFARE Classic (III): Internal Structure (3)

Manufacturer block

- Sector 0, block 0 (yellow one in previous slide)
- Contains:
  - UID (4B)
  - BCC (bit count check, 1B): XOR-ing of UID bytes
  - Manufacturer data (11B)
- Set and locked by manufacturer → read only!

MIFARE Classic (III): Internal Structure (3)

Manufacturer block

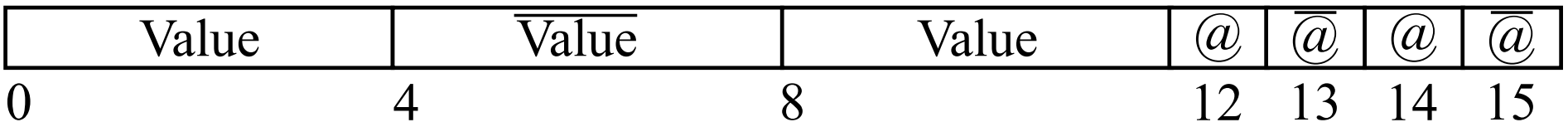
- Sector 0, block 0 (yellow one in previous slide)
- Contains:
  - UID (4B)
  - BCC (bit count check, 1B): XOR-ing of UID bytes
  - Manufacturer data (11B)
- Set and locked by manufacturer → read only!
  - Not the case for some Chinese cards ☺

MIFARE Classic (III): Internal Structure (4)

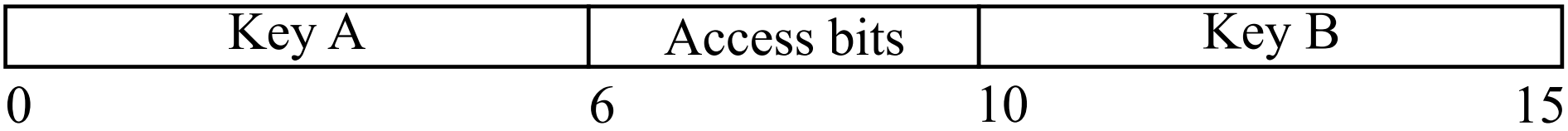
Storing data...

Storing data into blocks

- Read/write block
  - You can store data as you want, no matter how
- Data block
  - Predefined format (look below!)
  - Don't worry: APIs will help you!
    - Only need a *value*, it puts all the values properly on its own...)
  - Contains:
    - Value (twice)
    - Value negated (once)
    - 1-byte address (twice)
    - 1-byte address negated (twice)



MIFARE Classic (III): Internal Structure (5)



Sector trailer

- Last one in each sector (grey ones in previous slide)
- Contains:
  - Key A
  - Access Bits
  - Key B
- Authentication per sector before any operation is allowed
- Access bits define how is the auth. required and what operations are allowed
- Having fun with access bits may provoke a useless tag!
- Keys are set to `FFFFFFFFFFFFh` at delivery

MIFARE Classic (III): Internal Structure (6)

Operations

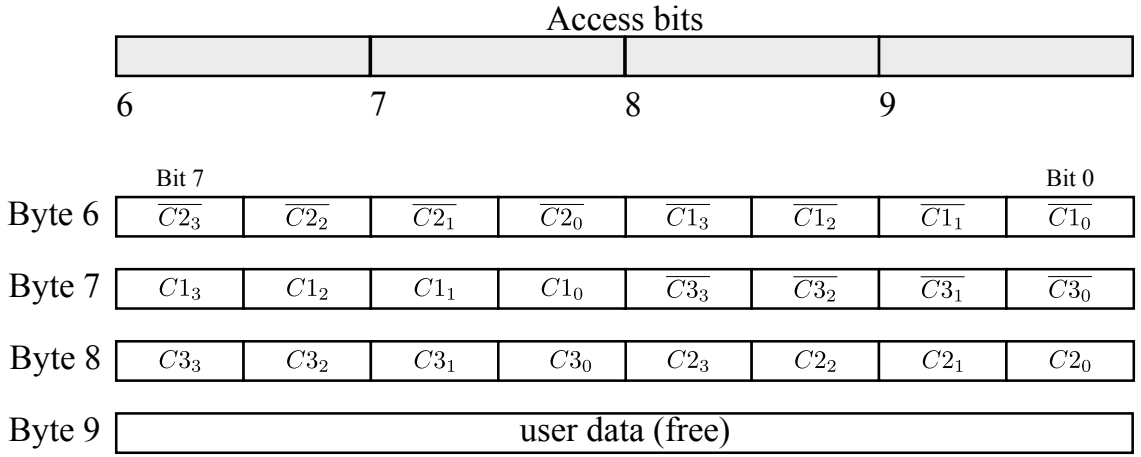
Operation	**Description**	**Valid for...**		
		R/W block	Value block	Sector trailer
Read	Reads a memory block	√	√	√
Write	Writes a memory block	√	√	√
Increment	Reads the value, increments it and stores		√	
Decrement	Reads the value, decrements it and stores		√	
Transfer	Transfers contents of internal register to a block		√	
Restore	Loads contents of a block to internal register		√	

MIFARE Classic (III): Internal Structure (7)

Access Conditions

Access Bits	Valid Commands	Block
$C1_0\,C2_0\,C3_0$	(all operations)	0
$C1_1\,C2_1\,C3_1$	(all operations)	1
$C1_2\,C2_2\,C3_2$	(all operations)	2
$C1_3\,C2_3\,C3_3$	Read, Write	3

- 3 bits defines the access conditions for every data block and sector trailer
- Stored non-negated and negated
- Commands are executed only after a successful authentication



MIFARE Classic (III): Internal Structure (8)

Access Conditions for sector trailer

Access Bits			**Access condition for...**					
			Key A		**Access bits**		**Key B**	
$C1$	$C2$	$C3$	read	write	read	write	read	write
0	0	0	-	key A	key A	-	key A	key A
0	0	1	-	key A	key A	key A	key A	key A
0	1	0	-	-	key A	-	key A	-
0	1	1	-	key B	key A (or B)	key B	-	key B
1	0	0	-	key B	key A (or B)	-	-	key B
1	0	1	-	-	key A (or B)	key B	-	-
1	1	0	-	-	key A (or B)	-	-	-
1	1	1	-	-	key A (or B)	-	-	-

(- means never)

MIFARE Classic (III): Internal Structure (9)

Access Conditions for data blocks

Access Bits			Access condition for...				Application
$C1$	$C2$	$C3$	Read	Write	Increment	Decrement, Transfer, Restore	
0	0	0	key A (or B)†	key A (or B)	key A (or B)	key A (or B)	Transport configuration
0	0	1	key A (or B)†	-	-	key A (or B)	Value block
0	1	0	key A (or B)†	-	-	-	R/W block
0	1	1	key B	key B	-	-	R/W block
1	0	0	key A (or B)	Key B	-	-	R/W block
1	0	1	key B	-	-	-	R/W block
1	1	0	key A (or B)	key B	key B	key A (or B)	Value block
1	1	1	-	-	-	-	R/W block

(- means never)

† if key B can be read in the sector trailer, then it cannot be used for authentication

MIFARE Classic: Communication Protocol (I)

Protocol steps

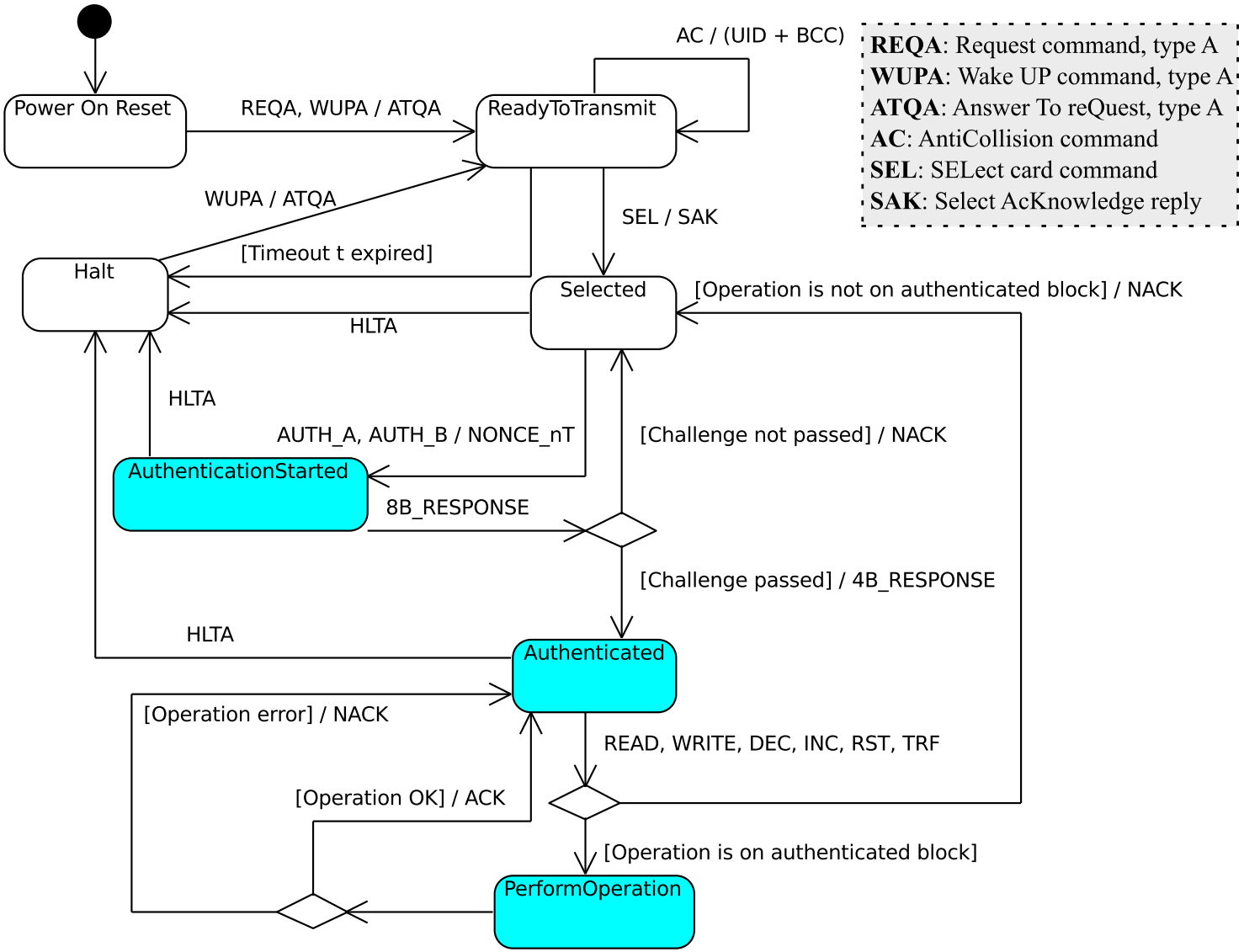
1. Get the tags in the reader's range
2. Select only one tag (anticollision loop)
3. Access a block, with key A or key B (starts authentication step)

Authentication step

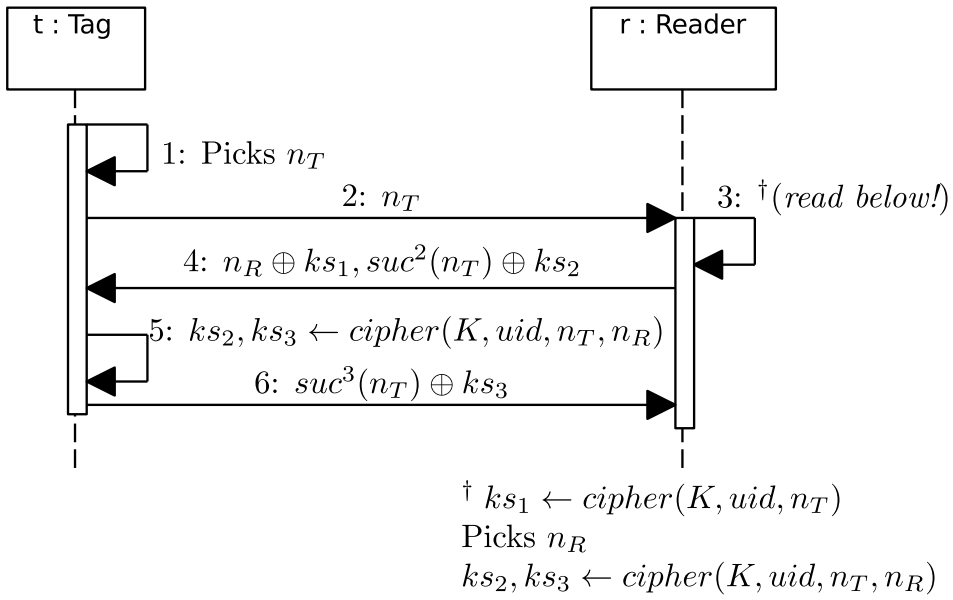
- Challenge-response mutual authentication using nonces
  - Nonce: randomly generated information
  - Nonces generated from a LFSR (next slides)

MIFARE Classic: Communication Protocol (II)

UML-SM of a NFC tag

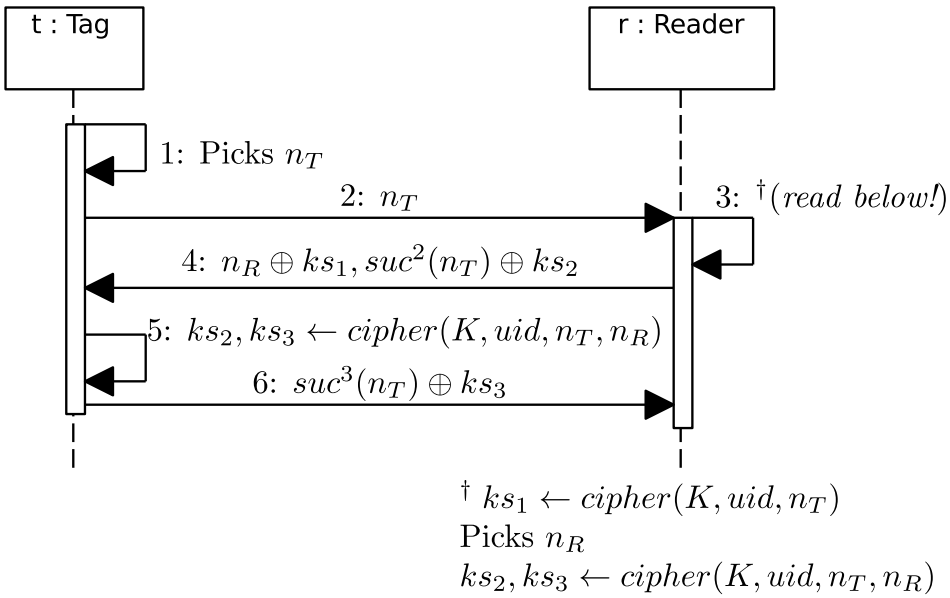


MIFARE Classic: Communication Protocol (III)



- Three-pass authentication
  1. Send nonce (n_T) as challenge
     - Generated by a 16-bit LFSR ($g(x) = x^{16} + x^{14} + x^{13} + x^{11} + 1$)
  2. Send response and other nonce n_R as challenge
  3. Send response
- Note: from n_T, communication is ciphered

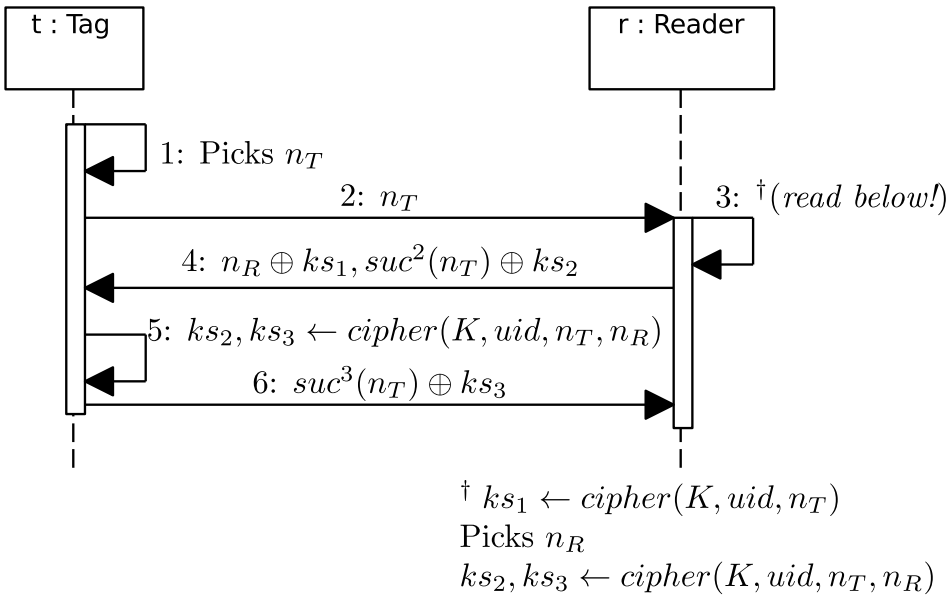
MIFARE Classic: Communication Protocol (IV)



Known plaintext [GKMRVSJ-ESORICS-08]

- Recall: n_T is in plaintext

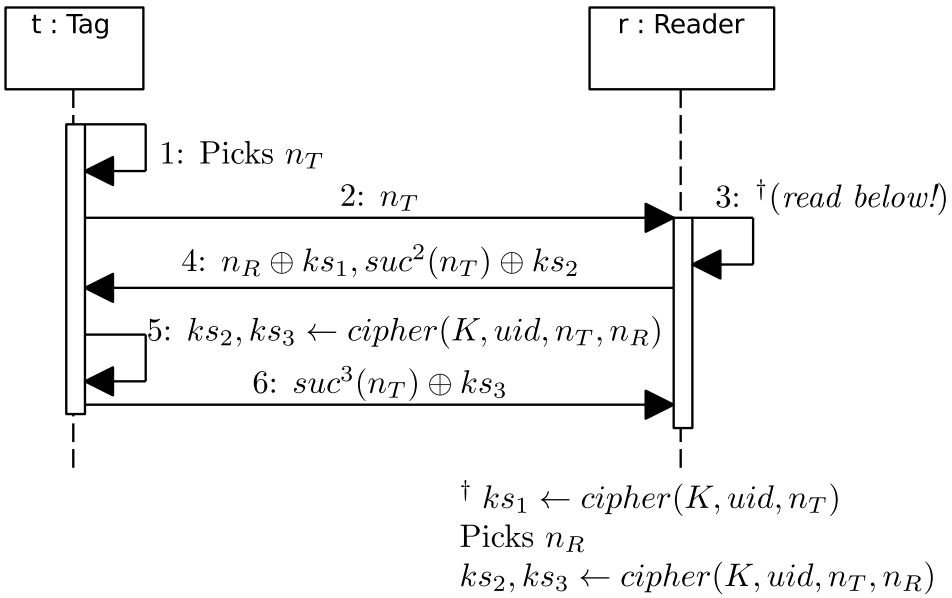
MIFARE Classic: Communication Protocol (IV)



Known plaintext [GKMRVSJ-ESORICS-08]

- Recall: n_T is in plaintext
- Given n_T, compute $suc^2(n_T) \rightarrow ks_2 = n_T \oplus suc^2(n_T)$

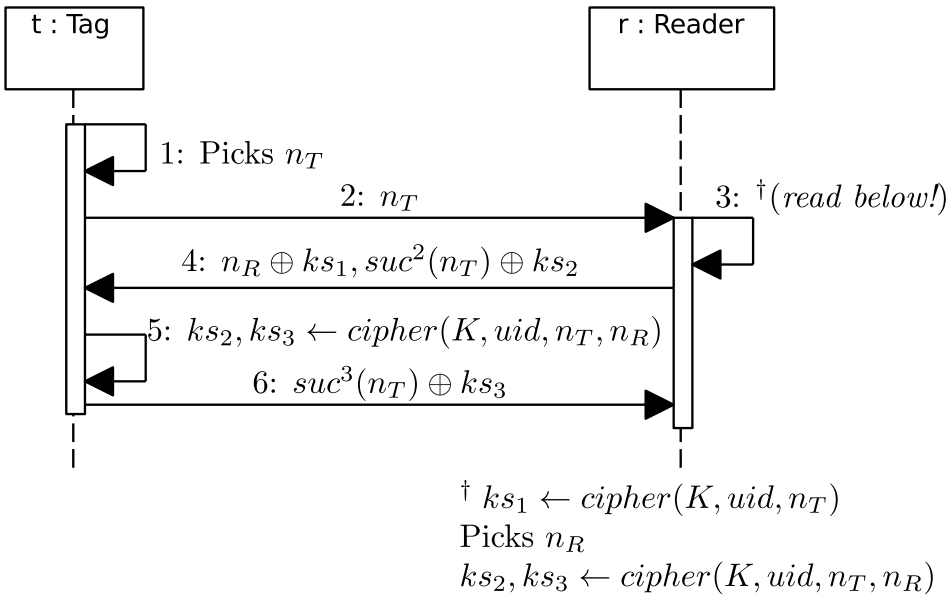
MIFARE Classic: Communication Protocol (IV)



Known plaintext [GKMRVSJ-ESORICS-08]

- Recall: n_T is in plaintext
- Given n_T, compute $suc^2(n_T) \rightarrow ks_2 = n_T \oplus suc^2(n_T)$
- When tag does not send last response, some readers time out and send `HLT` command XORed ks_3
  - `HLT` command is known, then we recover ks_3

MIFARE Classic: Communication Protocol (IV)



Known plaintext [GKMRVSJ-ESORICS-08]

- Recall: n_T is in plaintext
- Given n_T, compute $suc^2(n_T) \rightarrow ks_2 = n_T \oplus suc^2(n_T)$
- When tag does not send last response, some readers time out and send `HLT` command XORed ks_3
  - `HLT` command is known, then we recover ks_3
- Eavesdropping a successful authentication session
  - ks_2, ks_3 recovered from $suc^2(n_T) \oplus n_T, suc^3(n_T) \oplus n_T$

MIFARE Classic: CRYPTO1 (I)

- Proprietary stream cipher. Key length of 48 bits
- "Security by obscurity" principle
- Hardware on-chip: faster cryptographic operations!
- Reverted some years ago...:
  - K. Nohl and H. Plötz: "Mifare: Little Security, Despite Obscurity", in *Chaos Communication Congress*, 2007. Reverse engineering on silicon implementation
  - García et al.: "Dismantling MIFARE Classic", in *ESORICS* 2008. Fully disclosed the entire encryption algorithm

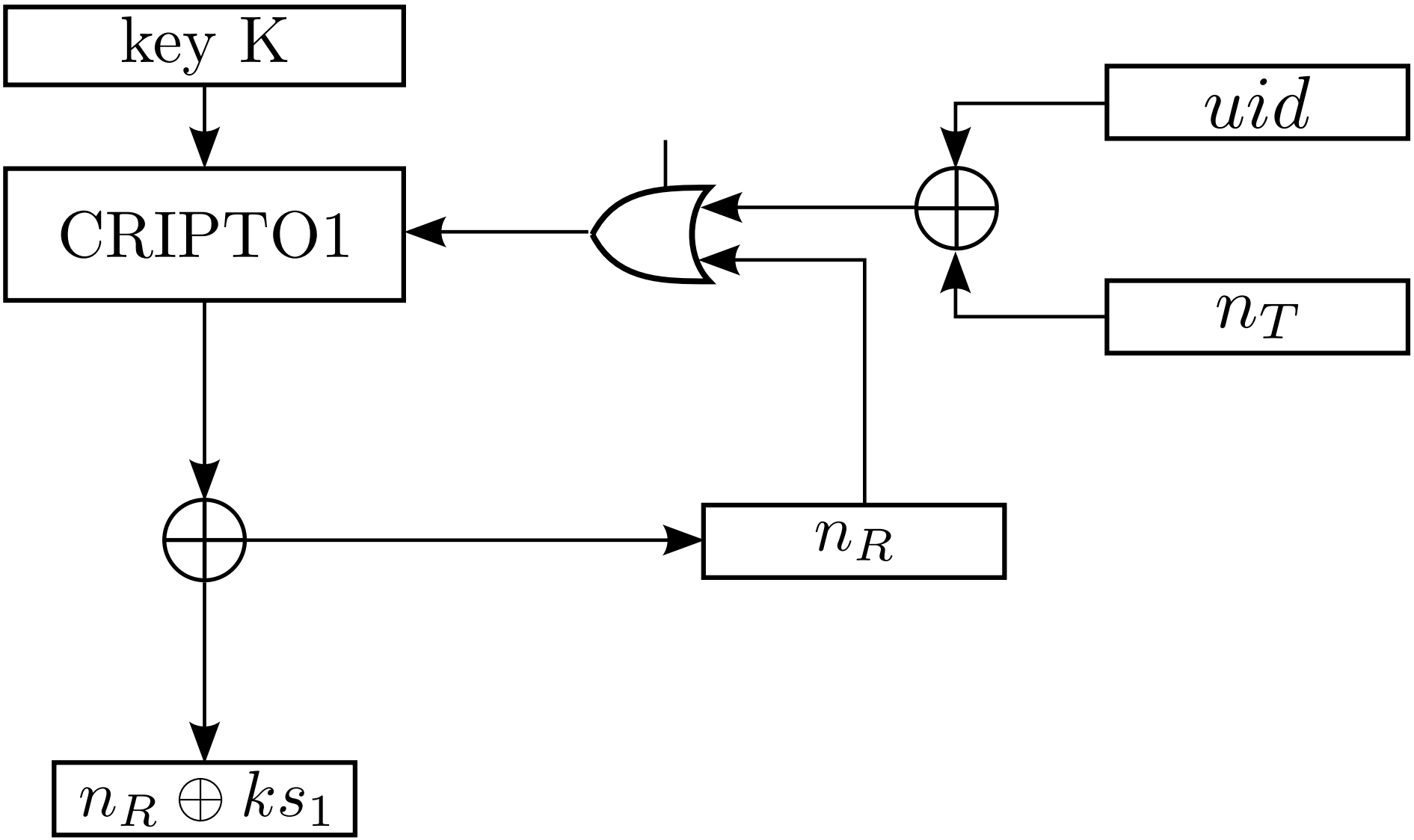
MIFARE Classic: CRYPTO1 (I)

- Proprietary stream cipher. Key length of 48 bits
- "Security by obscurity" principle
- Hardware on-chip: faster cryptographic operations!
- Reverted some years ago...:
  - K. Nohl and H. Plötz: "Mifare: Little Security, Despite Obscurity", in *Chaos Communication Congress*, 2007. Reverse engineering on silicon implementation
  - García et al.: "Dismantling MIFARE Classic", in *ESORICS* 2008. Fully disclosed the entire encryption algorithm
- Linear Feedback Shift Register (LFSR) + two-layer non-linear filter generator
  - At every clock tick, register is shifted one bit to the left
  - Leftmost bit: discarded
  - Feedback bit: computed with $g(x)$

$$g(x) = x^{48} + x^{43} + x^{39} + x^{38} + x^{36} + x^{34} + x^{33} + x^{31} + x^{29} + x^{24} + x^{23} + x^{21} + x^{19} + x^{13} + x^{9} + x^{7} + x^{6} + x^{5} + 1$$

MIFARE Classic: CRYPTO1 (II)

Initialisation diagram



MIFARE Classic: Known Weaknesses (I)

On the Pseudo-Random Number Generator

MOST CRITICAL weakness

Low entropy

- LFSR generating nonces: 16-bit length
- 0.6 seconds to generate ALL possible nonces ([NESP-USENIX-08])
- Generator resets to a known state every time the tag starts operating
  - Just a wait a fixed number of clock cycles...
  - Experimentally possible to get the same nonce every 30ms using Proxmark 3 reader

MIFARE Classic: Known Weaknesses (II)

On the Cryptographic Cipher

$$x_9, x_{11}, x_{13}, \ldots, x_{47}$$

Keystream generation

- Odd bits as inputs to the filter functions
- Divide-and-Conquer technique
  - Split even, odd bits in groups
  - Firstly focus on odd group:
    - After 2 shifts, new input is $x_{11}, x_{13}, \ldots, x_{47}$ and x_{49}
    - Used for generating two keystreams
    - Explore what bits generate the right keystreams
- Attack: Recover all sector keys without the needed of a genuine reader

MIFARE Classic: Known Weaknesses (III)

On the Cryptographic Cipher

$$x_9, x_{11}, x_{13}, \ldots, x_{47}$$

Leftmost bit not used in filter generator

- First 9 bits unused
- Attack: Rollback LFSR state bit a bit
  - Recover the initial state of LFSR

Statistical Bias [C-SECRYPT-09]

- With a $\pi = 0.75$, ks_1 is independent of the last three bits of n_R
- Attack: card-only attack
  - Recover one key, then apply nested authentication attack ([GKMRVSJ-ESORICS-08])
  - Does not require any pre-computation
  - Extremely fast, and requires a few hundred queries
  - More in the paper: `http://eprint.iacr.org/2009/137.pdf`

MIFARE Classic: Known Weaknesses (IV)

On the Communication Protocol

One-Time Padding (OTP)

- ISO-14443-A: every byte sent is followed by a parity bit
- MIFARE Classic computes parity bit over plaintext instead of ciphertext
- LFSR is not shifted after parity bit encryption

MIFARE Classic: Known Weaknesses (IV)

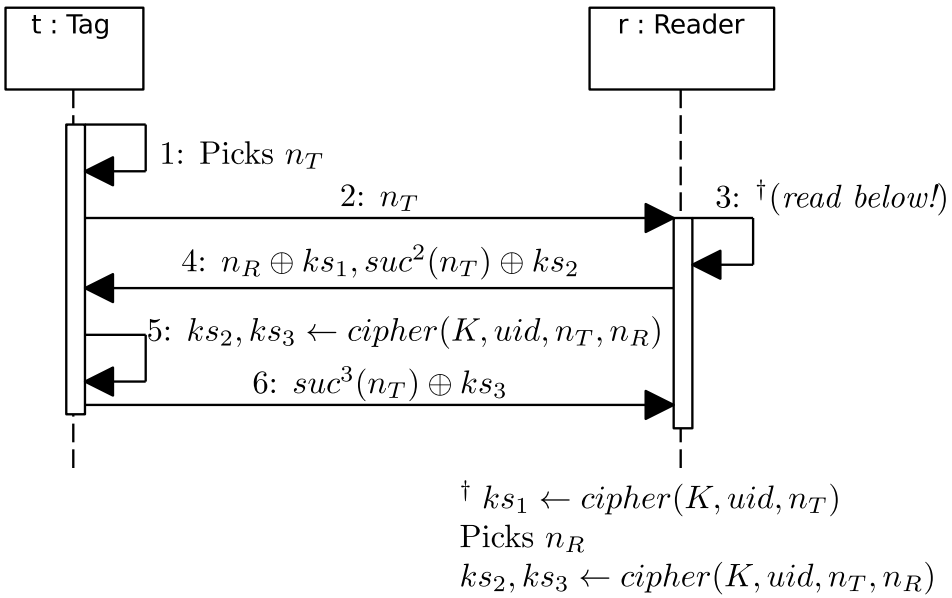
On the Communication Protocol

One-Time Padding (OTP)

- ISO-14443-A: every byte sent is followed by a parity bit
- MIFARE Classic computes parity bit over plaintext instead of ciphertext
- LFSR is not shifted after parity bit encryption
- Next plaintext and parity bit use the same keystream $\rightarrow$ OTP seems not to be OTP...
- More examples of violating OTP property:
  - Venona Project (U.S. counter-intelligence program during Cold War)
  - Point-to-Point Tunneling Protocol (PPTP)
  - IEEE 802.11 WEP

MIFARE Classic: Known Weaknesses (V)

On the Communication Protocol



Information Leak from Parity

- Second step in authentication, reader sends $n_R, suc^2(n_T)$
- PICC checks parity bits in n_R before checking $suc^2(n_T)$
  - When parity is incorrect, PICC does not answer
  - When $suc^2(n_T)$ is incorrect, it answers `NACK` (transmission error)
- `NACK` sent encrypted $\rightarrow$ ks_3 can be recovered

MIFARE Classic: Known Weaknesses (VI)

On the Deployment

Default Keys

- Some chip manufacturers leave default keys on chips
- This is obvious, as companies must make the effort to do system integration for clients. . . (sic!)
- RTFM: Chip manufacturer warns about CHANGING default keys
- Default keys are well-known and documented

FFFFFFFFFFFFh	000000000000h	1A982C7E459Ah
A0A1A2A3A4A5h	B0B1B2B3B4B5h	AABBCCDDEEFFh
D3F7D3F7D3F7h	4D3A99C351DDh	

Outline

Related Work (I)

On MIFARE Classic weaknesses analysis (1)

NP-CCC-07 K. Nohl and H. Plötz, "Mifare: Little Security, Despite Obscurity", in *Chaos Communication Congress*, 2007.

GKMRVSJ-ESORICS-08 García et al., "Dismantling MIFARE Classic", in *Procs. of the European Symposium on Research in Computer Security* (ESORICS), 2008.

KHG-CARDIS-08 G.d Koning Gans et al., "A Practical Attack on the MIFARE Classic", in *Procs. of the Smart Card Research and Advanced Applications Conference* (CARDIS), 2008.

NESP-USENIX-08 K. Nohl et al., "Reverse-Engineering a Cryptographic RFID Tag". In *USENIX Security Symposium*, 2008.

GRBS-SP-09 F.D. García et al., "Wirelessly Pickpocketing a Mifare Classic Card", in *Procs. of the 30th IEEE Symposium on Security and Privacy* (S&P), 2009.

Related Work (II)

On MIFARE Classic weaknesses analysis (2)

C-SECRYPT-09 N.T. Courtois, "The Dark Side of Security by Obscurity and Cloning MiFare Classic Rail and Building Passes Anywhere, Anytime". In *Procs. of the Int. Conf. on Security and Cryptography* (SECRYPT), 2009

GRBS-SP-09 F.D. García et al., "Wirelessly Pickpocketing a Mifare Classic Card", in *Procs. of the 30th IEEE Symposium on Security and Privacy* (S&P), 2009

Tan-MScThesis-09 W.H. Tan, "Practical Attacks on the MIFARE Classic", Imperial College London, 2009

On NFC Attacks

VK-NFC-11 R. Verdult and F. Kooman, "Practical Attacks on NFC Enabled Cell Phones". In *Procs. of the 3rd Int. Workshop on Near Field Communication*, 2011

Related Work (III)

On MIFARE Attacks

- Sogeti ESEC Pentest: "Playing with NFC for fun and coffee"
- BackTrack Linux: "RFID Cooking with Mifare Classic" (2012)
- C. Miller, "Exploring the NFC Attack Surface", in *BlackHat US*, 2012.
- ComputerWorld article: "Android NFC hack enables travelers to ride subways for free, researchers say" (2012)
- HackPlayers: "Cómo colarse en el metro de forma elegante" (2012)
- Security ArtWork: "Hacking RFID, rompiendo la seguridad de Mifare" (2010)

On NFC-related issues

- R. Lifchitz, Hacking the NFC credit cards for fun and debit (Hackito Ergo Sum 2012)
- J.M. Esparza, Give me your credit card, the NFC way (NcN'12)

Outline

A Case Study (I)

Once upon a time. . .

- Imagine a place using MIFARE Classic cards
- Used for multiple purposes:
  - Access to public transport services
  - Use of public facilities

A Case Study (I)

Once upon a time. . .

- Imagine a place using MIFARE Classic cards
- Used for multiple purposes:
  - Access to public transport services
  - Use of public facilities
- In the (near) future:
  - Taxi payments
  - Citizen rent info for discounts

A Case Study (II)

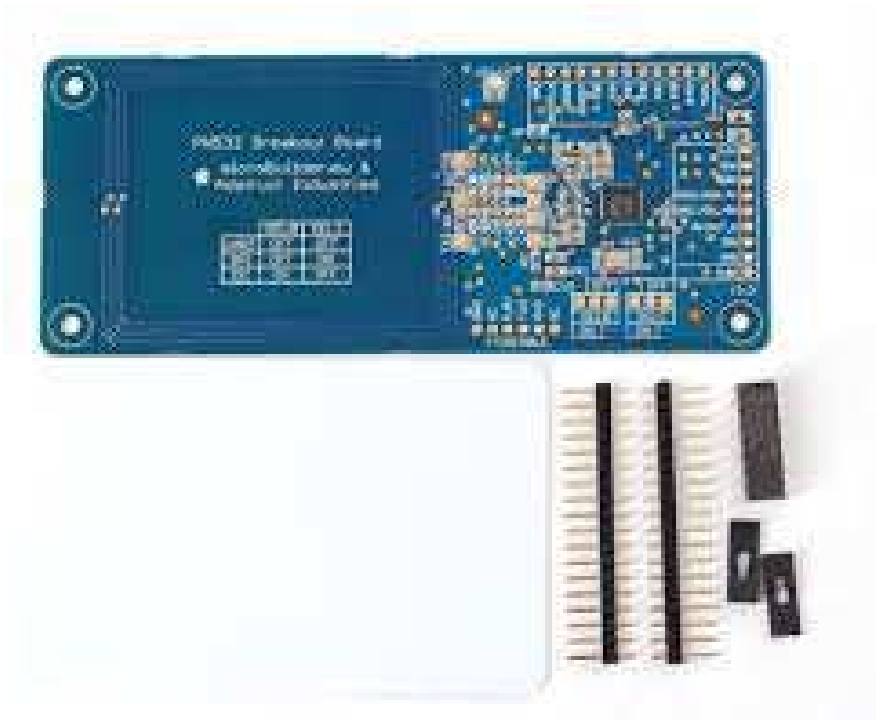
Problem Analysis

Specific goals

- Figure out the pair of keys (A, B)
- Make a dump of a real card
- Study the card content
- Check any integrity about unauthorised content alteration
- Make a clone card
- Do a mobile app for card-hacking

A Case Study (III)

Lab Environment



Hardware

- AdaFruit PN532 and USB-FTDI cable
- A computer
- A NFC-enabled phone*

Software

- C compiler
- NFC Library (`libnfc`)
- NFC tools (`nfc-tools`)
- Mifare Offline Cracker (`mfoc`)

Recall: Tell the story about phones

A Case Study (IV)

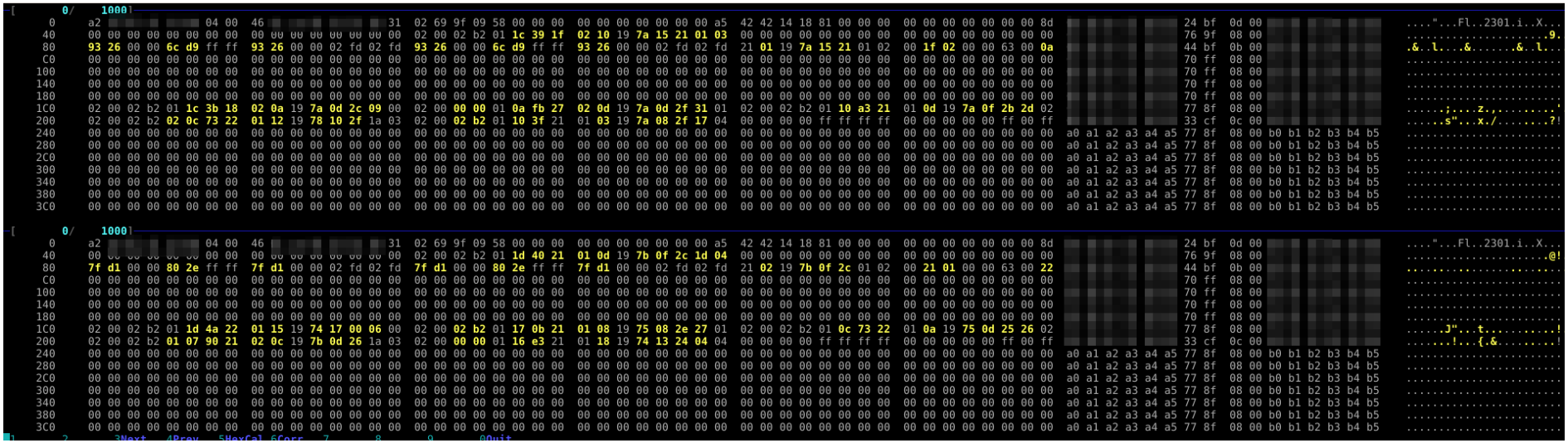
- Two different Classic version
  - MIFARE Classic 1K (T1)
  - MIFARE Classic 4K (T2)

```
death@mulita:~/Downloads/mfoc-0.10.2/src$ time sudo mfoc -O out -P 100
    ATQA (SENS_RES): 00  04
* UID size: single
* bit frame anticollision supported
      UID (NFCID1):
     SAK (SEL_RES): 08
* Not compliant with ISO/IEC 14443-4
* Not compliant with ISO/IEC 18092
Fingerprinting based on ATQA & SAK values:
* Mifare Classic 1K
* Mifare Plus (4-byte UID) 2K SL1
* SmartMX with Mifare 1K emulation
[Key: 000000000000] -> [................]
[Key: f          f] -> [................]
[Key: f          f] -> [................]
[Key: a          9] -> [xxxxxxxxxx......]
[Key: a          6] -> [xxxxxxxxxx......]
[Key: a          0] -> [xxxxxxxxxx......]
[Key: a          8] -> [xxxxxxxxxx......]
[Key: a          4] -> [xxxxxxxxxx......]
[Key: 7          b] -> [xxxxxxxxxxxx..xx]
[Key: 2          3] -> [xxxxxxxxxxxx..xx]
[Key: c          2] -> [xxxxxxxxxxxx..xx]
[Key: 3          0] -> [xxxxxxxxxxxx..xx]
[Key: 5          b] -> [xxxxxxxxxxxxxxxx]
[Key: 3          6] -> [xxxxxxxxxxxxxxxx]
[Key: 6          1] -> [xxxxxxxxxxxxxxxx]
[Key: 0          3] -> [xxxxxxxxxxxxxxxx]
[Key: 0          3] -> [xxxxxxxxxxxxxxxx]
[Key: 0          9] -> [xxxxxxxxxxxxxxxx]
[Key: a          5] -> [xxxxxxxxxxxxxxxx]
[Key: b          5] -> [xxxxxxxxxxxxxxxx]
[Key: 7          9] -> [xxxxxxxxxxxxxxxx]

Sector 00 -  FOUND KEY  [A]  Sector 00 -  FOUND KEY  [B]
```

```
Block 15, type A, key a          9 :00  00  00  0
Block 14, type A, key a          9 :00  00  00  0                                                  00  00
Block 13, type A, key a          9 :00  00  00  0                                                  00  00
Block 12, type A, key a          9 :00  00  00  0                                                  00  00
Block 11, type A, key a          9 :00  00  00  0                                                  3b  a2
Block 10, type A, key a          9 :00  00  00  0                                                  00  00
Block 09, type A, key a          9 :00  00  00  0                                                  02  fd
Block 08, type A, key a          9 :00  00  00  0                                                  02  fd
Block 07, type A, key a          9 :00  00  00  0                                                  3b  a2
Block 06, type A, key a          9 :00  00  00  0                                                  00  00
Block 05, type A, key a          9 :03  00  02  b                                                  35  04
Block 04, type A, key a          9 :00  00  00  0                                                  00  00
Block 03, type A, key a          9 :00  00  00  0                                                  3b  a2
Block 02, type A, key a          9 :54  5a  03  0                                                  00  2c
Block 01, type A, key a          9 :03  00  00  0                                                  00  dd
Block 00, type A, key a          9 :

real    0m35.551s
user    0m0.244s
sys     0m0.004s
```

A Case Study (V)

Understanding the card content...

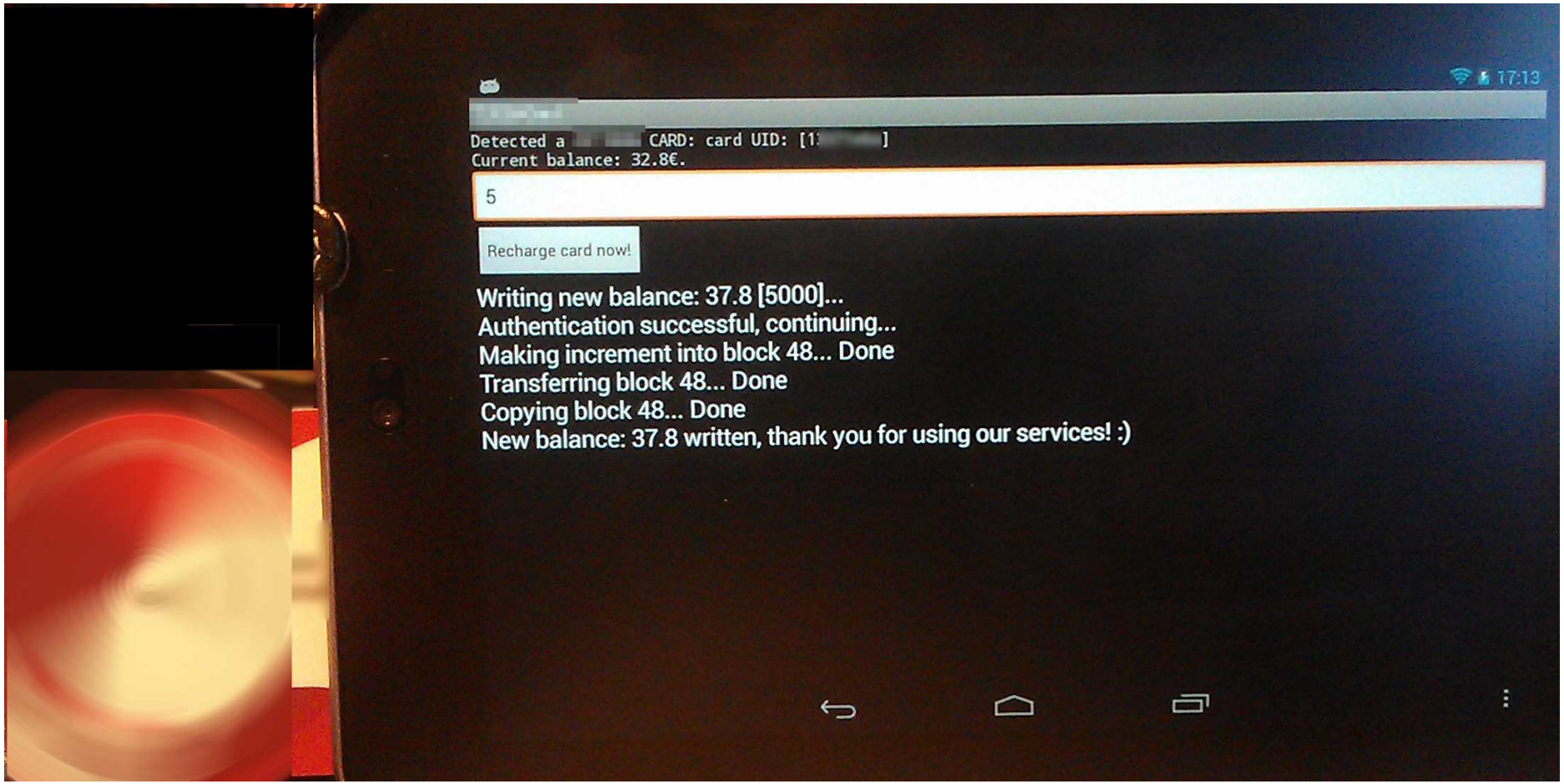


Summary of data

	T1	T2
Card ID	(0, 3)	(10, 3)
Last bus used	(1, 2)	(1, 2)
Current balance	(2, [1, 2])	(12, [1, 2])
Historic	(7, [1, 2, 3]), (8, [1, 2])	(7, [1, 2, 3]), (8, [1, 2])

A Case Study (VI)

Building a PoC in Android O.S. (1)



A Case Study (VII)

Building a PoC in Android O.S. (2)

It's demo time!

A Case Study (IIX)

Recalling the initial goals

Goal	Achieved?	Some remarks
Figure out the pair of keys (A, B)	$\surd$	Some keys are the default ones
Make a dump of a real card	$\surd$	Fast, and simple
Study the card content	$\surd$	Not a single bit encrypted
Check any integrity about unauthorised content alteration	$\surd$	no integrity
Make a clone card	$\surd^*$	A perfect clone (Chine cards rulez!)
Do a mobile app for card-hacking	$\surd$	Android fuc·ing rocks!

A Case Study (IIX)

Thinking (and acting?) badly... (1)

What else could be done...

- Identity spoofing
  - Possible penalties for spoofed people
  - Consume the real balance of someone else
- Use of all public services for free
- Black market?
  - Fake recharge point
  - Whether I sold a card illegitimately charged...
- Just put the app in Google Play, and have fun ☺

A Case Study (IX)

Thinking (and acting?) badly... (2): Replay attacks

A Case Study (IX)

Event timeline

Nov 2012 Nice chat with J.M. Esparza ⌣̈

A Case Study (IX)

Event timeline

Nov 2012 Nice chat with J.M. Esparza ⌣̈

Nov 2012 (ending) Lab environment set and tested (it works!)

A Case Study (IX)

Event timeline

Nov 2012 Nice chat with J.M. Esparza ☺

Nov 2012 (ending) Lab environment set and tested (it works!)

Dec 2012 Nice chat with C. Lorenzana ☺ (at STIC CCN-CERT conference)

A Case Study (IX)

Event timeline

Nov 2012 Nice chat with J.M. Esparza ☺

Nov 2012 (ending) Lab environment set and tested (it works!)

Dec 2012 Nice chat with C. Lorenzana ☺ (at STIC CCN-CERT conference)

Mar 2013 Confidential report is sent to GDT

A Case Study (IX)

Event timeline

Nov 2012 Nice chat with J.M. Esparza ☺

Nov 2012 (ending) Lab environment set and tested (it works!)

Dec 2012 Nice chat with C. Lorenzana ☺ (at STIC CCN-CERT conference)

Mar 2013 Confidential report is sent to GDT

Apr 2013 Report is being handled by CNPIC

A Case Study (IX)

Event timeline

Nov 2012 Nice chat with J.M. Esparza ¨⌣

Nov 2012 (ending) Lab environment set and tested (it works!)

Dec 2012 Nice chat with C. Lorenzana ¨⌣ (at STIC CCN-CERT conference)

Mar 2013 Confidential report is sent to GDT

Apr 2013 Report is being handled by CNPIC

May 2013 Company says the problem is known, but does not really care about it...

A Case Study (IX)

Event timeline

Nov 2012 Nice chat with J.M. Esparza ☺

Nov 2012 (ending) Lab environment set and tested (it works!)

Dec 2012 Nice chat with C. Lorenzana ☺ (at STIC CCN-CERT conference)

Mar 2013 Confidential report is sent to GDT

Apr 2013 Report is being handled by CNPIC

May 2013 Company says the problem is known, but does not really care about it...

(today) As they don't care, me neither. Here I am! ☺

A Case Study (X)

Lessons Learned

- It's good to collaborate with police... but you need to be patient
  - You'll have a good sleep at night and not in jail...

A Case Study (X)

Lessons Learned

- It's good to collaborate with police... but you need to be patient
  - You'll have a good sleep at night and not in jail...
  - You also get some free beer from C. Lorenzana ☺

A Case Study (X)

Lessons Learned

- It's good to collaborate with police... but you need to be patient
  - You'll have a good sleep at night and not in jail...
  - You also get some free beer from C. Lorenzana ⌣̈
- Security is not considered (as normally) in a Spanish company
  - Not at the beginning of a product design
  - Not even when someone spots out the problem
  - They quantify the risk of people exploiting the problem...
- This is not U.S., unfortunately (in this case)

A Case Study (X)

Lessons Learned

- It's good to collaborate with police... but you need to be patient
  - You'll have a good sleep at night and not in jail...
  - You also get some free beer from C. Lorenzana ⌣̈
- Security is not considered (as normally) in a Spanish company
  - Not at the beginning of a product design
  - Not even when someone spots out the problem
  - They quantify the risk of people exploiting the problem...
- This is not U.S., unfortunately (in this case)

Remember, not economic gain but free beer instead!

Outline

Conclusions

Some conclusions...

- MIFARE Classic is like a memory card
- Vulnerable from 2009
- Weaknesses and attacks very well-known and widely documented

Conclusions

Some conclusions...

- MIFARE Classic is like a memory card
- Vulnerable from 2009
- Weaknesses and attacks very well-known and widely documented
- Need to defend against
  - Unauthorised content alteration
  - Replay attacks
  - Clone attacks

Conclusions

Some conclusions...

- MIFARE Classic is like a memory card
- Vulnerable from 2009
- Weaknesses and attacks very well-known and widely documented
- Need to defend against
  - Unauthorised content alteration
  - Replay attacks
  - Clone attacks

Thinking to deploy MIFARE Classic as an access control system?

Conclusions

Some conclusions...

- MIFARE Classic is like a memory card
- Vulnerable from 2009
- Weaknesses and attacks very well-known and widely documented
- Need to defend against
  - Unauthorised content alteration
  - Replay attacks
  - Clone attacks

Thinking to deploy MIFARE Classic as an access control system?

Don't.

Hacking the NFC cards for fun and honor degrees

Ricardo J. Rodríguez

All wrongs reversed

rjrodriguez@fi.upm.es ※ @RicardoJRdez ※ www.ricardojrodriguez.es

Universidad Politécnica de Madrid
Madrid, Spain

November 15, 2013

Universidad de Zaragoza
Zaragoza (España)