# Black-Box Side-Channel Attacks Highlight the Importance of Countermeasures - An Analysis of the Xilinx Virtex-4 and Virtex-5 Bitstream Encryption Mechanism -


**Amir Moradi, Markus Kasper and Christof Paar**

**Horst Görtz Institute for IT-Security**

**Ruhr-Universität Bochum**


**Presented by: Benedikt Gierlichs, KU Leuven**


Session ID: CRYP-107

Session Classification: Advanced

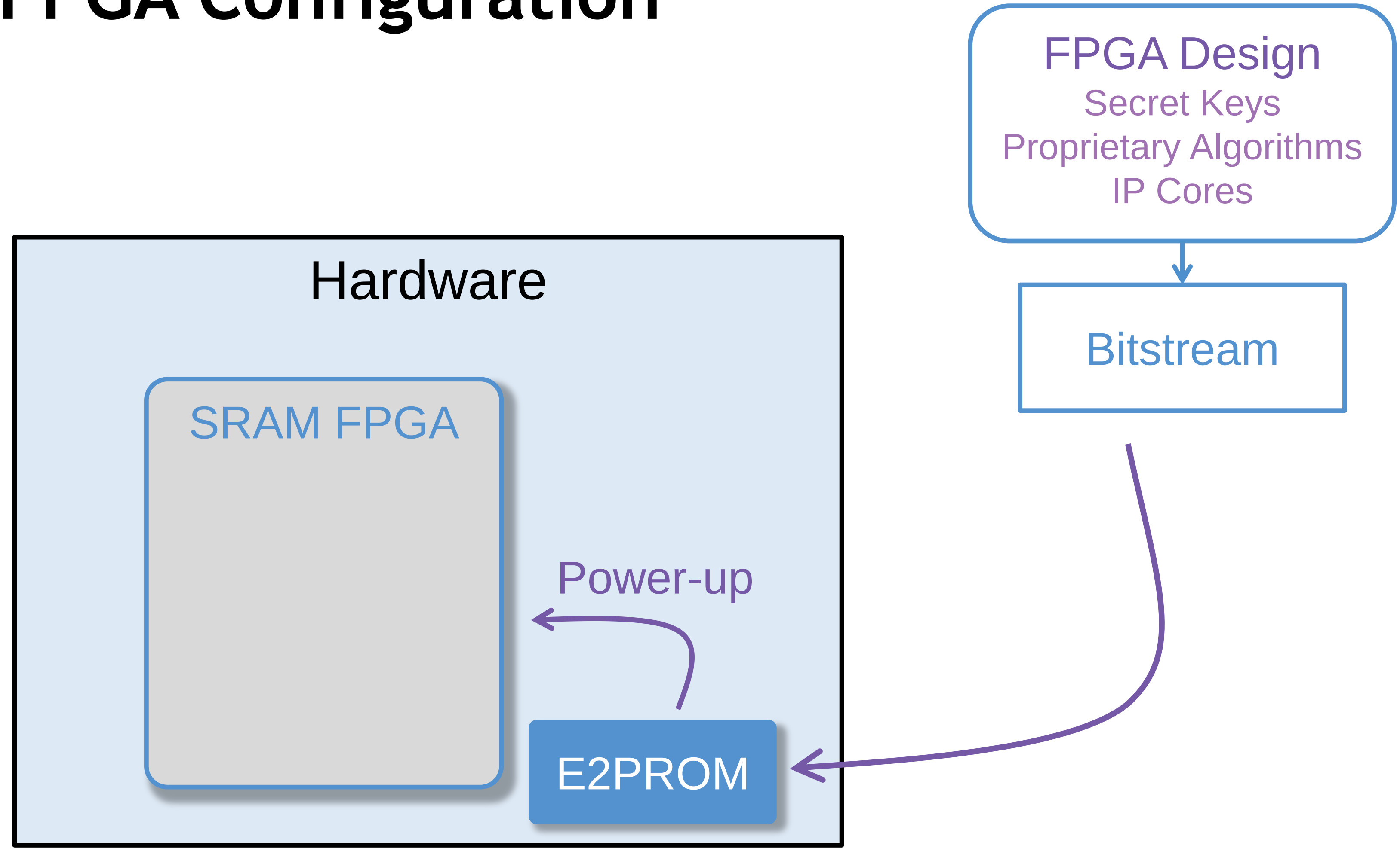
RSA CONFERENCE 2012

# Agenda

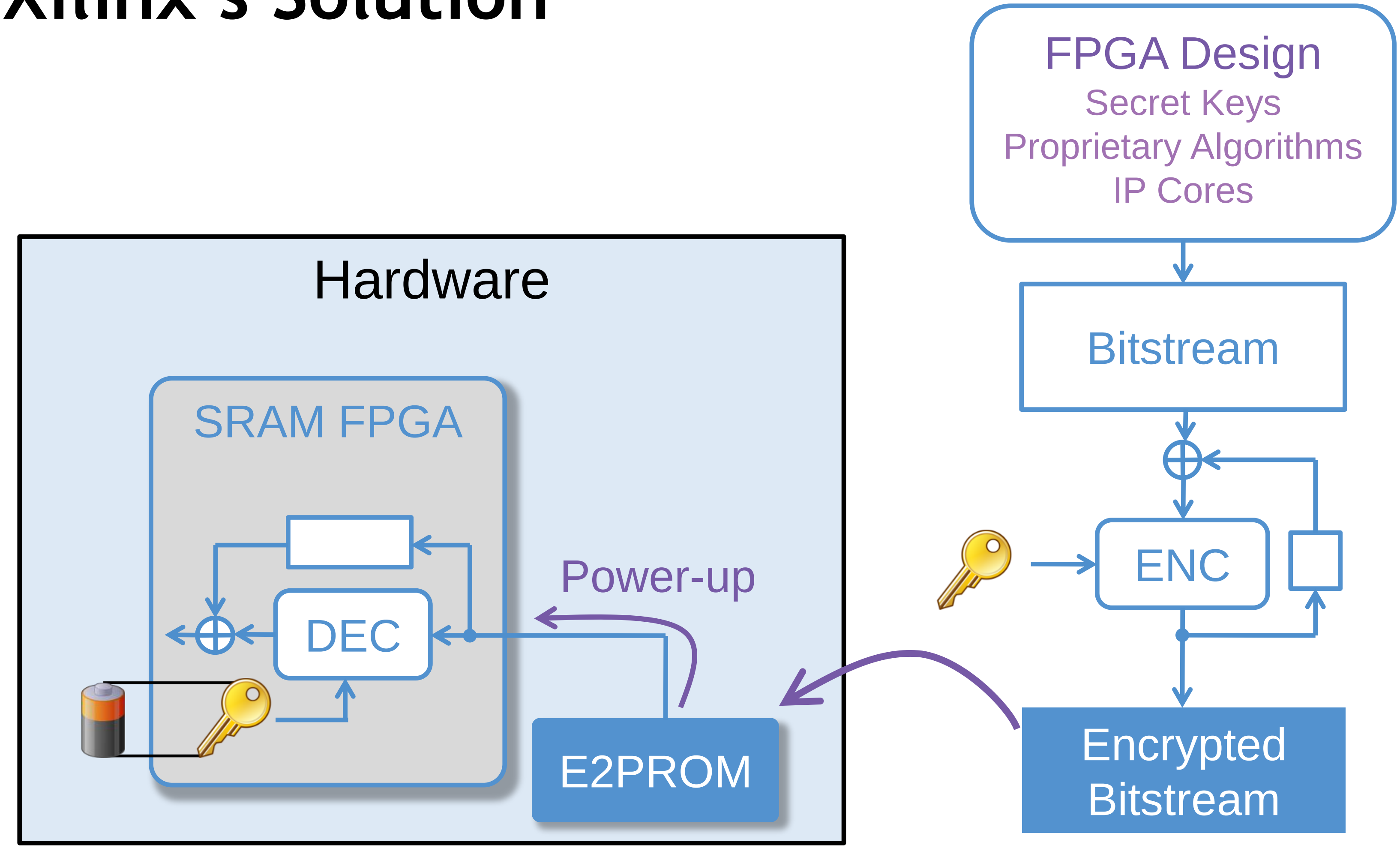
- Introduction to Xilinx Bitstream Encryption
- Motivation: Real-World Security Evaluation
- Our Attack
- Summary

# Introduction to Xilinx's Bitstream Encryption

# FPGA Configuration



# Xilinx's Solution



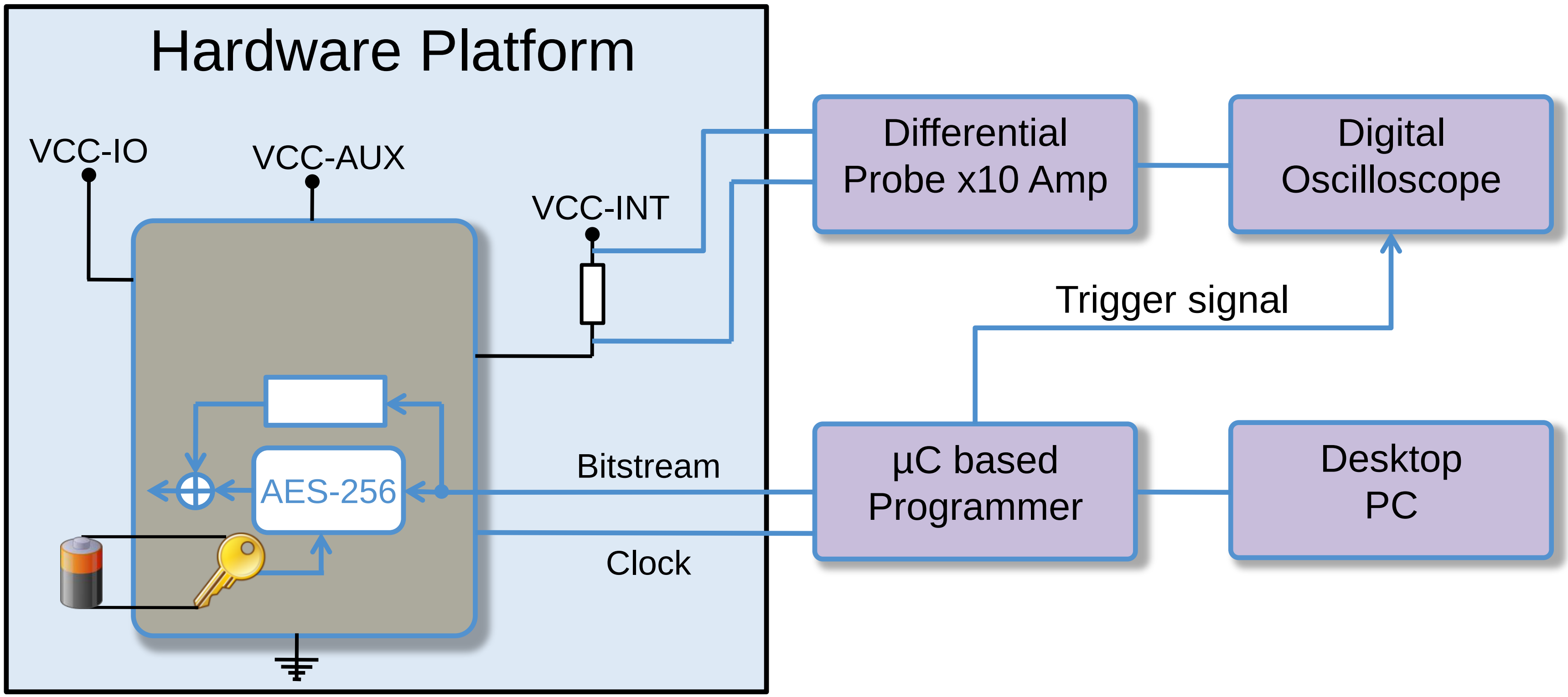
# Xilinx's Solution

- Virtex-II Pro Series
  - 3-DES encryption in CBC mode
  - Broken in 2011 by Moradi et. al
- Virtex-4 to Virtex-6 series, Xilinx 7 series and several Spartan-6
  - AES-256 encryption in CBC mode

# Motivation: Real-World Security Evaluation

# Our Attack

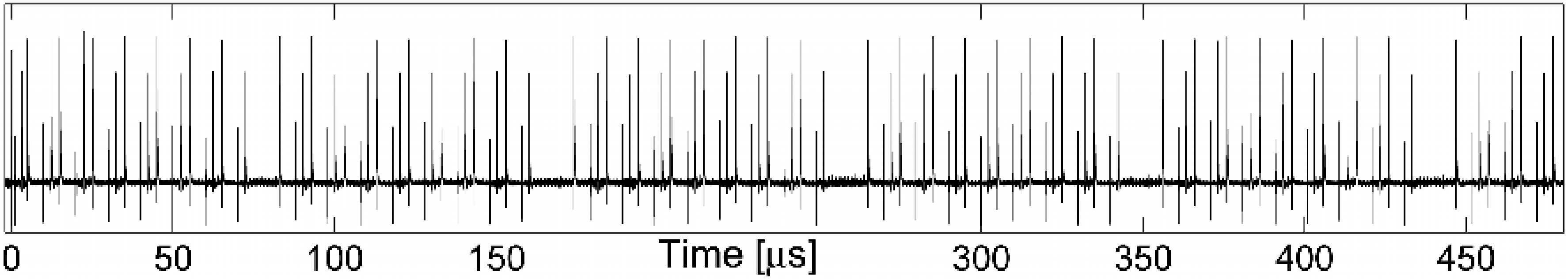
# Setup



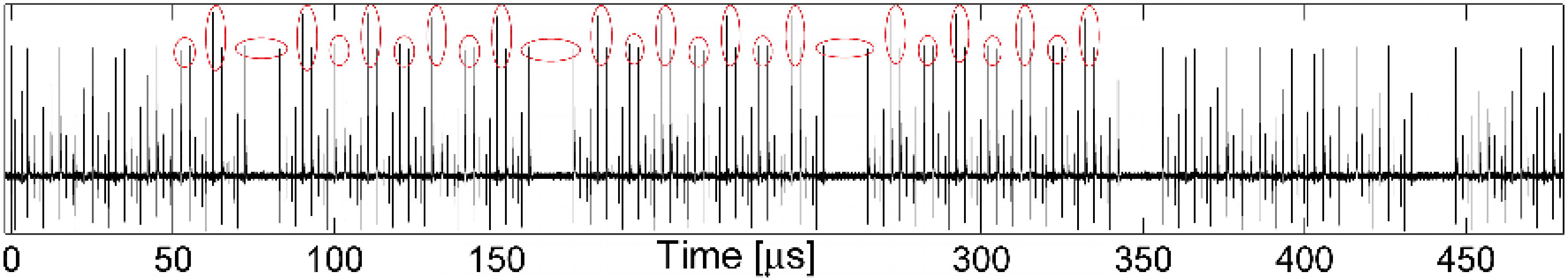
Hardware Platform
VCC-IO
VCC-AUX
VCC-INT
AES-256
Bitstream
Clock
Differential
Probe x10 Amp
Digital
Oscilloscope
Trigger signal
µC based
Programmer
Desktop
PC

# Finding the Decryption

- Compare average power consumption
  - Unencrypted bitstream

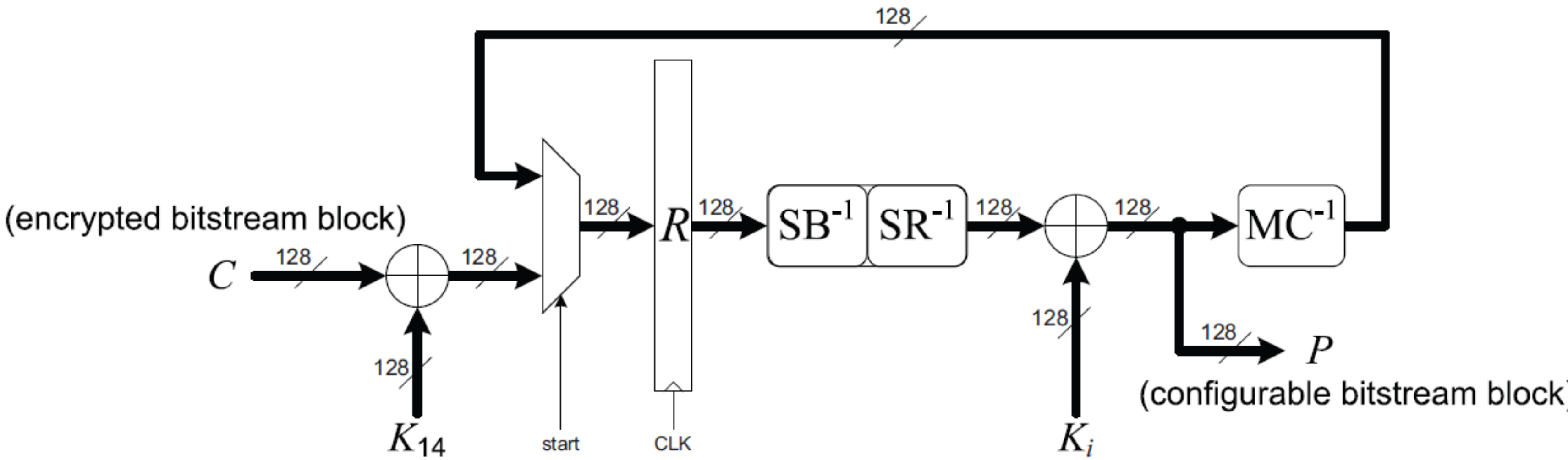


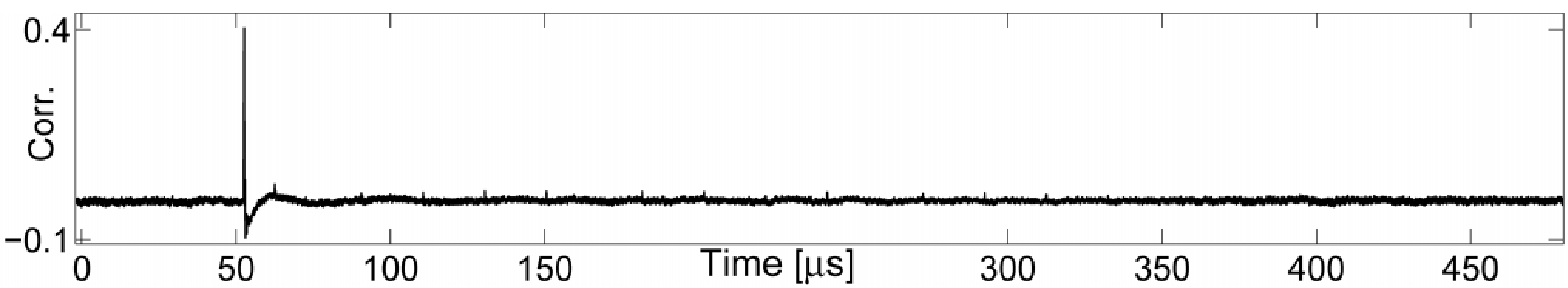

  - Encrypted bitstream



# Finding the Decryption

- Correlate models in known key scenario



# Model for Power Consumption

- Hamming Distance of state register R

$$\Delta R_{1,2} = \Big[\underbrace{C \oplus K_{14}}_{R_1}\Big] \oplus \Big[\underbrace{\mathrm{MC}^{-1}\Big(\mathrm{SB}^{-1}\big(\mathrm{SR}^{-1}(R_1)\big) \oplus K_{13}\Big)}_{R_2}\Big]$$

- Problem:
  At least 64-bit hypothesis to attack power consumption of 32-bit leakage

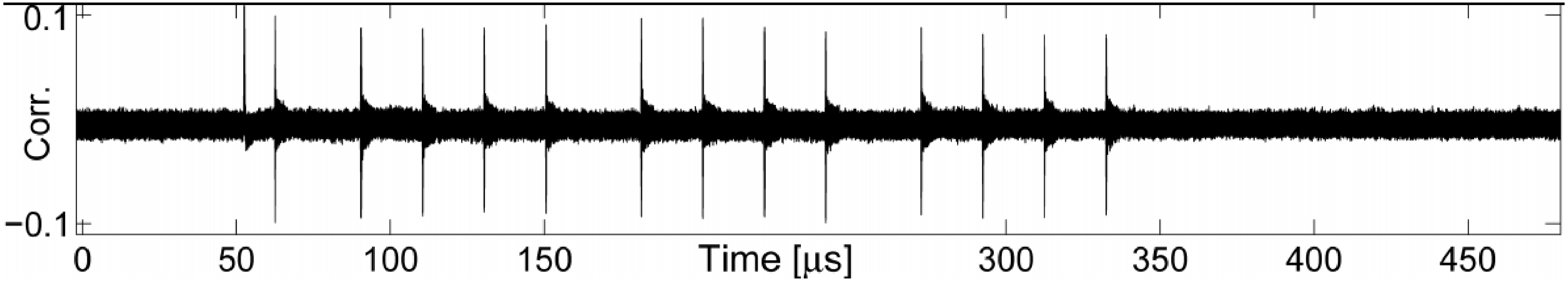
# Model for Power Consumption

$$\underbrace{\mathrm{MC}^{-1}\left(\mathrm{SB}^{-1}\left(\mathrm{SR}^{-1}(R_1)\right)\right)}_{R'_2} \oplus \underbrace{\mathrm{MC}^{-1}\left(K_{13}\right)}_{K'_{13}}$$

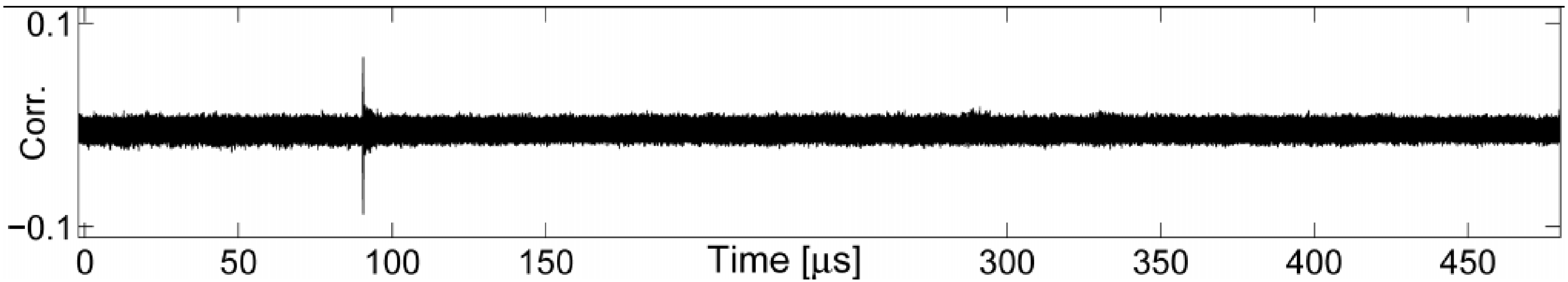
- Exploit linearity
- 32-bit hypotheses on $K_{14}$ (in $R_1$) to attack with single bit power model
- Fine in theory, but can we detect the leakage of a single bit in practice?

# Model for Power Consumption

- Yes we can!




- …and we learn our model is not accurate…



# The Attack

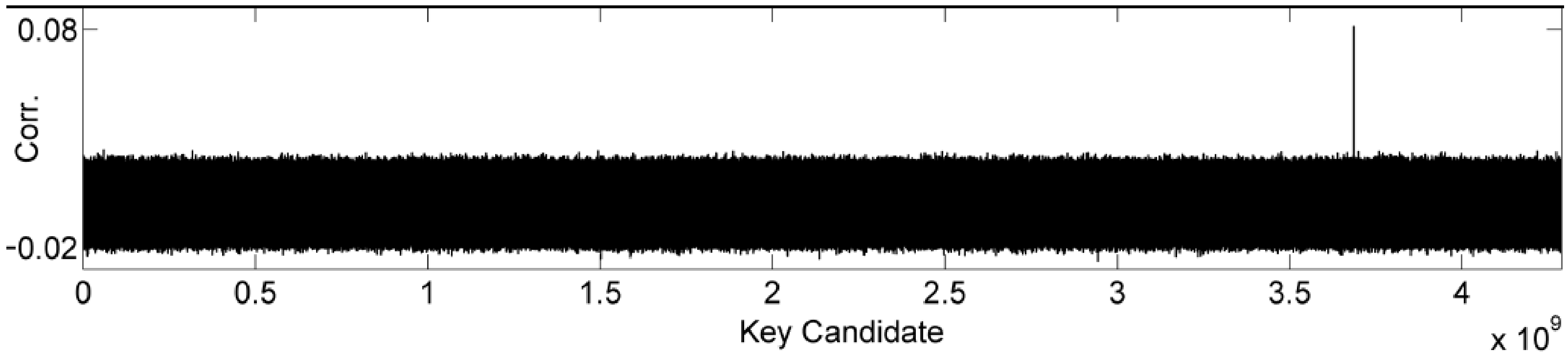
- $2^{35}$ (= 34,359,738,368) keys to test
- 60,000 power traces
- 128 GiB of 32-bit floating point results
- Can be done but not practical on CPUs

# GPUs for Power Analysis

- Used System
  - 4x Nvidia Tesla C2070 GPUs
  - Each one has 6 GB of RAM and 448 cores
  - Clocked at 1.15 GHz
- HDD is not the bottleneck
- Full attack in around 4.5 hours (V4, 60k traces)

# Result

- Virtex-4 60k traces




- Other Columns show similar results
- Virtex-5:
  The same attack works (6.5 hours, 90k traces)

# Lessons Learned

- Bitstream encryption is vulnerable to SCA
- New modern CMOS technology can be attacked in practice (90nm/65nm/45nm)
- Reusing crypto cores simplifies analyses
- Attacks on 32-bit hypotheses are realistic threats
- GPUs are a nice tool for attacks where computation time dominates

# Recent Results and Further Work

# Recent Results and Ongoing Work

- Same attack works on 45nm Spartan-6 devices
- Ongoing work: Testing other FPGAs
- Expect significantly improved attack

# The HMAC Feature

- Virtex-6 series and Xilinx 7 series
  - Additional SHA256 HMAC authentication feature
- Aims to prevent fuzzing attacks
- Relies on bitstream encryption
  - HMAC and HMAC key embedded in encrypted bitstream

# Consequences: A Threat Analysis

# Stolen Bitstream Threats

1. Copy the design
2. Reverse engineer the design
3. Modify the design

# Threat Summary

- Cloning threat is real
- Expect others to come within the next years
- Remember:
  Each bitstream in an FPGA deployed today will also be available for analysis the next years

# Solutions/Actions for Manufacturers

- No reliable digital solution available
- Tamper resistance: Deny access to side-channels
- Expect former products to be attacked
- Make sure to minimize gain from bitstream reversal
- Ask for devices with improved encryption features
- Don’t put sensitive IP in FPGA prototypes/engineering samples of ASICs
- There is no new threat!

# Summary

# Summary

- Virtex-II Pro, Virtex-4, Virtex-5 and Spartan-6 shown to be vulnerable
- FPGA bitstream encryption not reliable
- Expect more and faster attacks the next years

# Thanks For Your Attention

- For more info on our work visit the website:
  - http://www.emsec.rub.de/research/projects/BitEnc/
- Latest results available in eprint version
  - http://eprint.iacr.org/2011/391
- Contact:
  - Email: emsec+BitEnc@rub.de

# It's Q&A Time!

# Backup Slides

# Analysis for End Customers

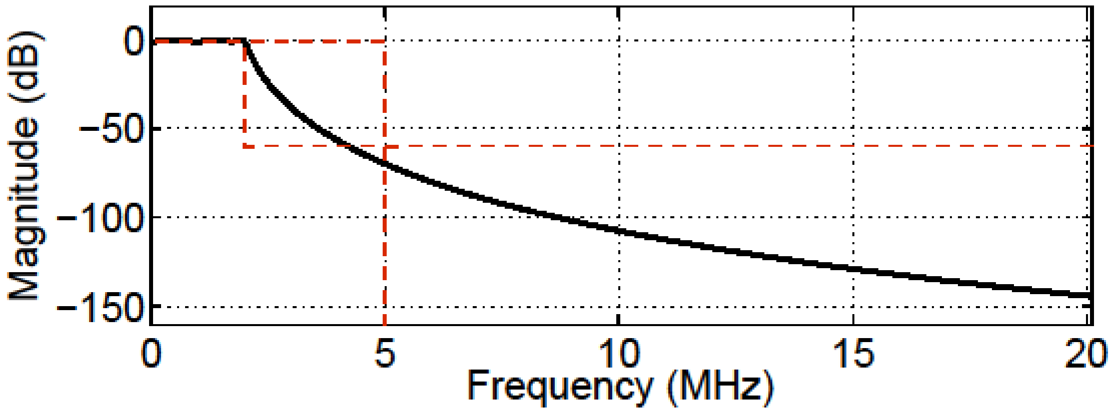
- Security architects are aware of this
  - There is no big surprise
  - System security should not rely on bitstream encryption
- Bitstreams unprotected for years
- Counterfeit Cisco router incident ~ 2008
  - Lessons learned:
    - Watch out for counterfeit IT products
    - Verify your supply chain
    - Don't trust our infrastructure

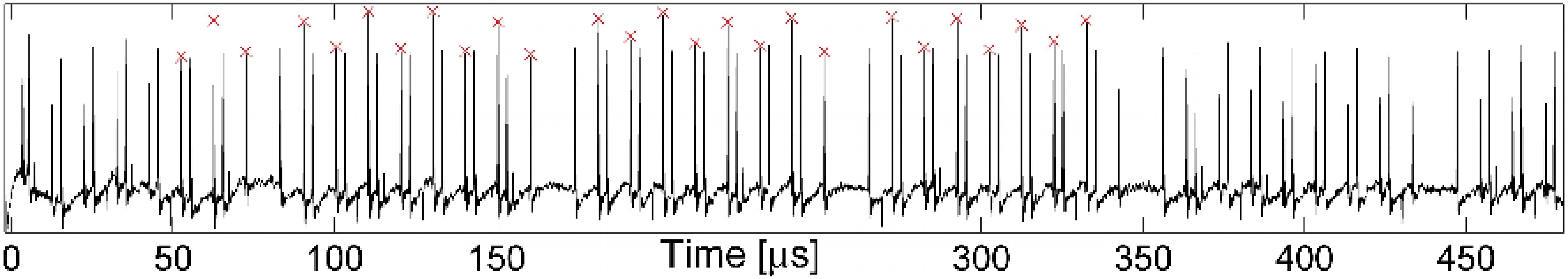
# Analysis for FPGA/ASIC

- Consider SCA countermeasures
  - Obviously improves resistance, but no guarantee
- Consider non-volatile memory
  - At least big enough to allow customers to implement (SCA resistant) secure bootloaders
- Consider obscurity
  - We are in side-channel land!
    Obscurity can significantly harden attacks
  - Make use of the Device DNA ("Serial Number") to get device specific individual keys
- Consider re-designing security blocks in new products
  - Avoids a single point of failure

# Applied Filter

- Encryption after lowpass filter



# Evaluating Side-Channel Security

- An attack that did not work does not provide reliable insights
- Even worse, it suggests security…

“... back in the Virtex II Pro days, we issued a challenge, and more than 7 universities and research groups accepted the challenge.

We provided a 2vp7 [Ed.: Virtex2 Pro VP7] pcb with usb port, and pins for access to power, that had the key battery installed (300 mA lithium coin cell), and the part was programmed with a 3DES encrypted bitstream.

All 7 challengers gave up. Their basic conclusion was all the things they thought would work, differential power attack, spoofing by power glitches, attack with freeze spray, etc. FAILED.”

Principal engineer, Xilinx, on comp.arch.FPGA, 3/5/2008

# Power Analysis of Atmel CryptoMemory - Recovering Keys from Secure EEPROMs

**Josep Balasch**[1], Benedikt Gierlichs[1], Roel Verdult[2], Lejla Batina[1,2], and Ingrid Verbauwhede[1]

**[1] ESAT/COSIC, KU Leuven**

**[2] ICIS/Digital Security Group, R.U. Nijmegen**

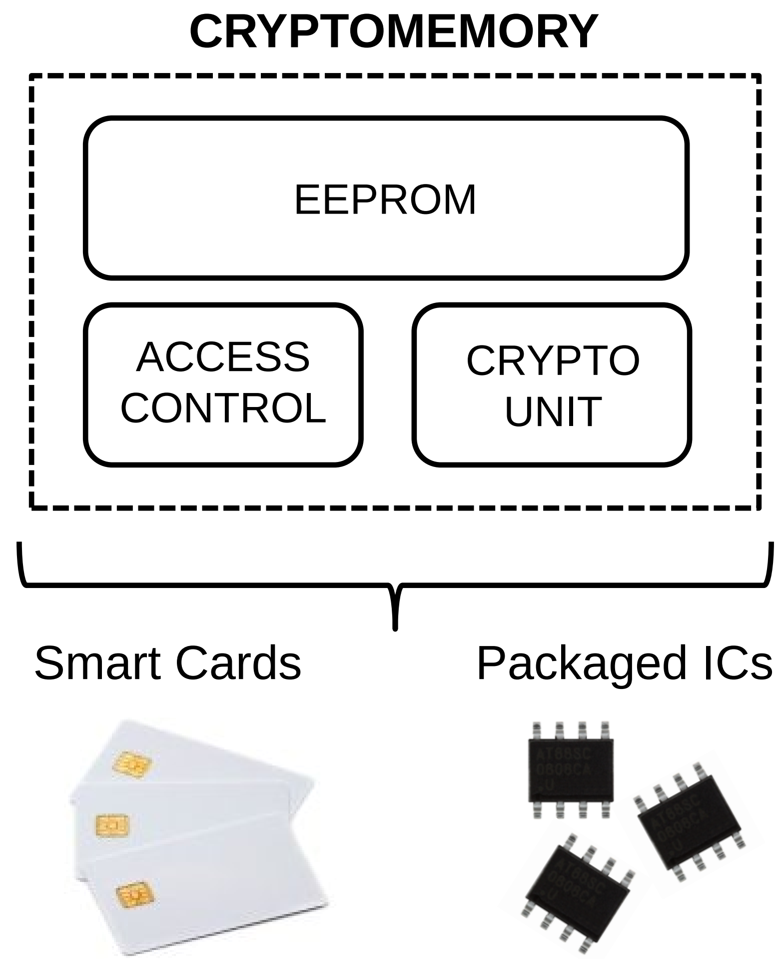
Session ID: CRYP-107

# Outline

- Background on CryptoMemory
- Experimental setup
- Study of power traces
- Enabling power analysis
- Straightforward DPA attack
- Conclusions

# CryptoMemory. Background (I)

- Secure memories with authentication
- Read/write access to EEPROM upon authentication
- Recording of failed attempts (AACs)
- Commercial applications
  - Secure storage
    - Cryptographic keys, e-wallets, …
  - Anti-counterfeiting
    - Printer cartridges, …

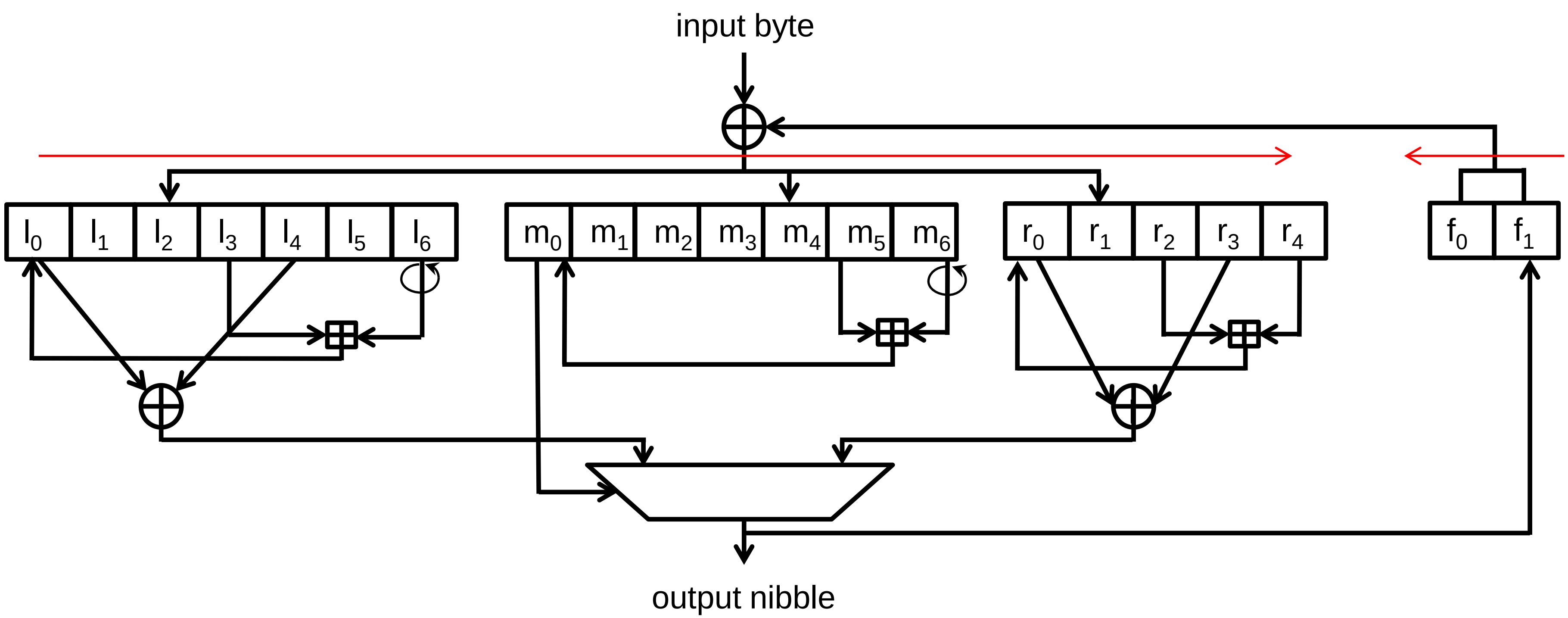


# CryptoMemory. Background (II)

- Related work
  - [GvRVS10] Reverse-engineered authentication protocol and stream cipher used in CryptoMemory
    - 2640 eavesdropped authentications, with $2^{52}$ cipher ticks
  - [BKZ11] Improved attack
    - 30 eavesdropped authentications, with $2^{50}$ cipher ticks
    - 2-6 days on a cluster with 200 cores
- Goals
  - Evaluate physical security of CryptoMemory devices
  - Can we find a more practical attack to extract the secret authentication keys?

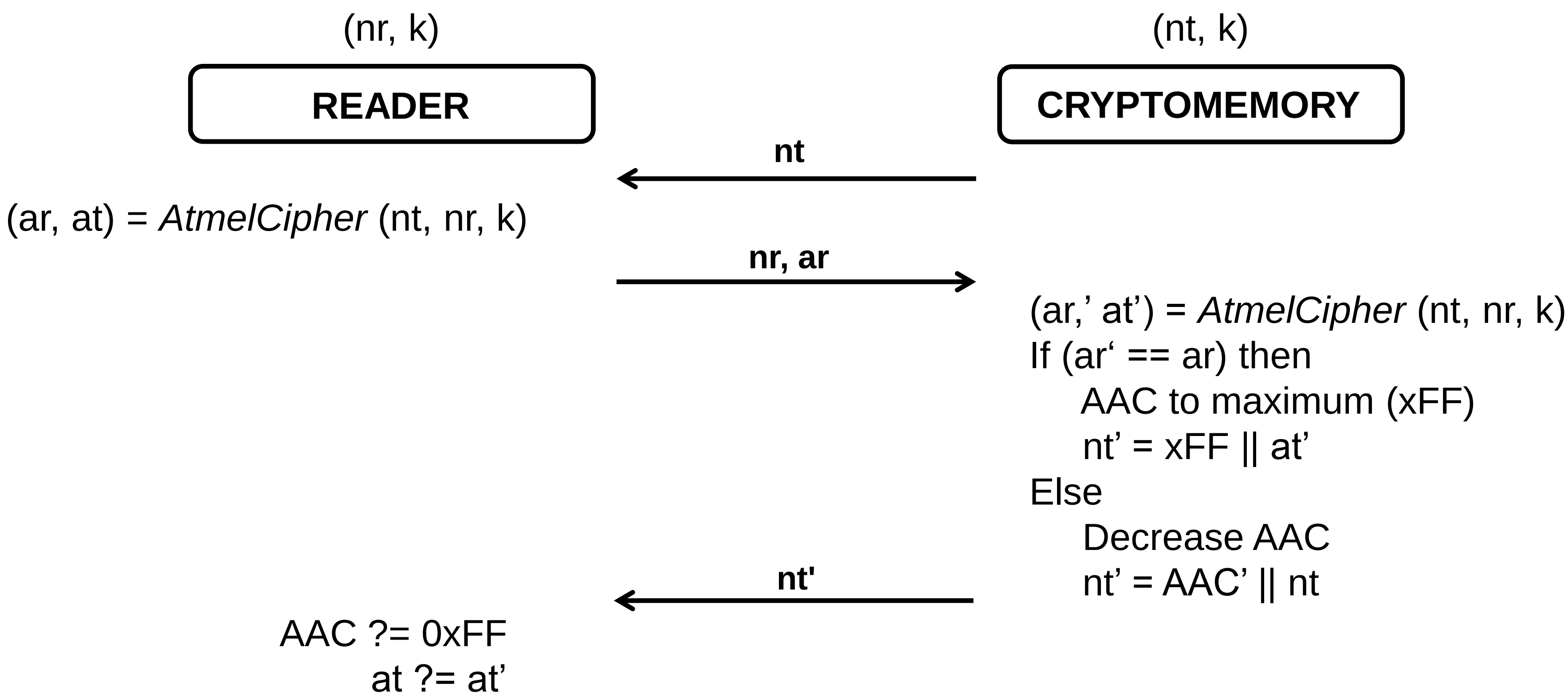
# CryptoMemory. Atmel stream cipher

- State: element of $F_2^{117}$ composed by 4 registers
- Each tick: 8 bits input → 4 bits output

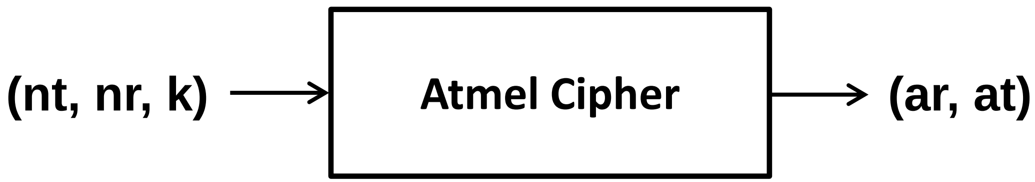


# CryptoMemory. Authentication

- Mutual authentication protocol with counters



# CryptoMemory. Computing authenticators

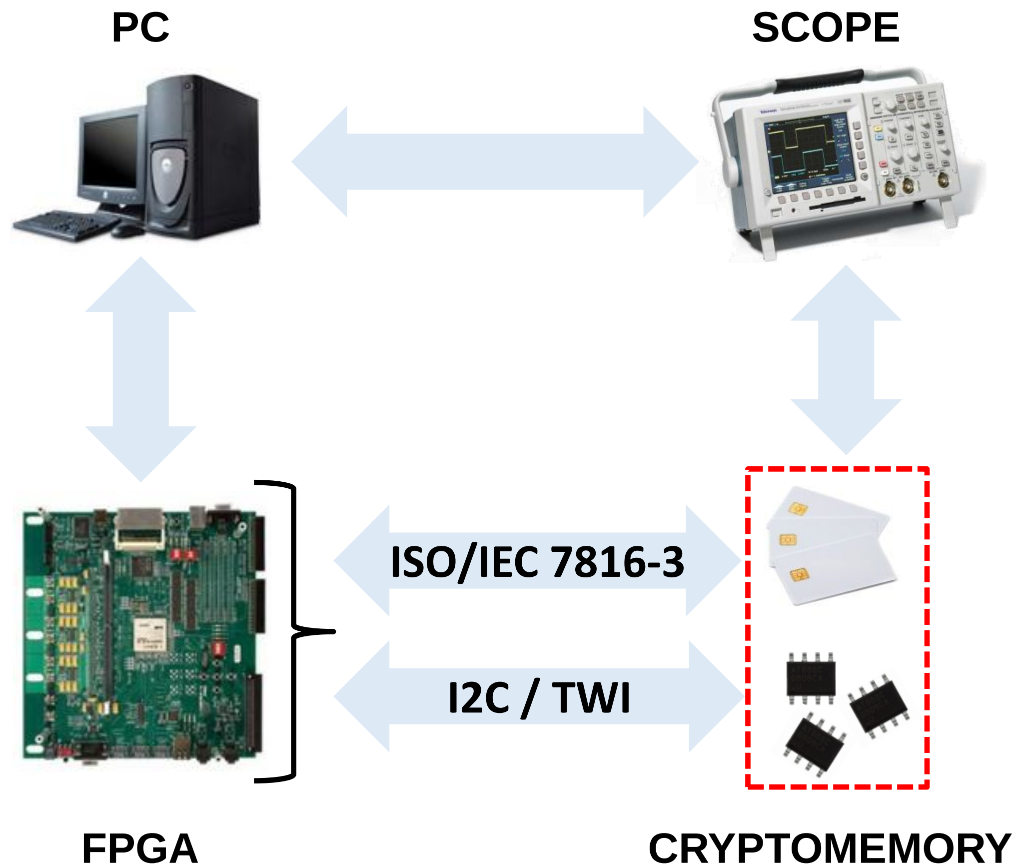



| TICKS | INPUT | | | | | | | OUTPUT |
|---|---|---|---|---|---|---|---|---|
| **0 to 6** | nt0 | nt0 | nt0 | nt1 | nt1 | nt1 | nr0 | - |
| **7 to 13** | nt2 | nt2 | nt2 | nt3 | nt3 | nt3 | nr1 | - |
| **14 to 20** | nt4 | nt4 | nt4 | nt5 | nt5 | nt5 | nr2 | - |
| **21 to 27** | nt6 | nt6 | nt6 | nt7 | nt7 | nt7 | nr3 | - |
| **28 to 34** | k0 | k0 | k0 | k1 | k1 | k1 | nr4 | - |
| **35 to 41** | k2 | k2 | k2 | k3 | k3 | K3 | nr5 | - |
| **42 to 48** | k4 | k4 | k4 | k5 | k5 | k5 | nr6 | - |
| **49 to 55** | k6 | k6 | k6 | k7 | k7 | k7 | nr7 | - |
| **56 to 125** | 0 | 0 | 0 | 0 | … | | 0 | ar, at |

- Ticks 0 to 55
  - Scramble nonces and key
- Ticks 56 to 125
  - Generate authenticators

# Experimental Setup

- FPGA as central element
- Communication with <u>any</u> CryptoMemory
- Accurate control over all external signals
  - I/O, Vcc, Rst, Clk, …
- Scope collects power measurements

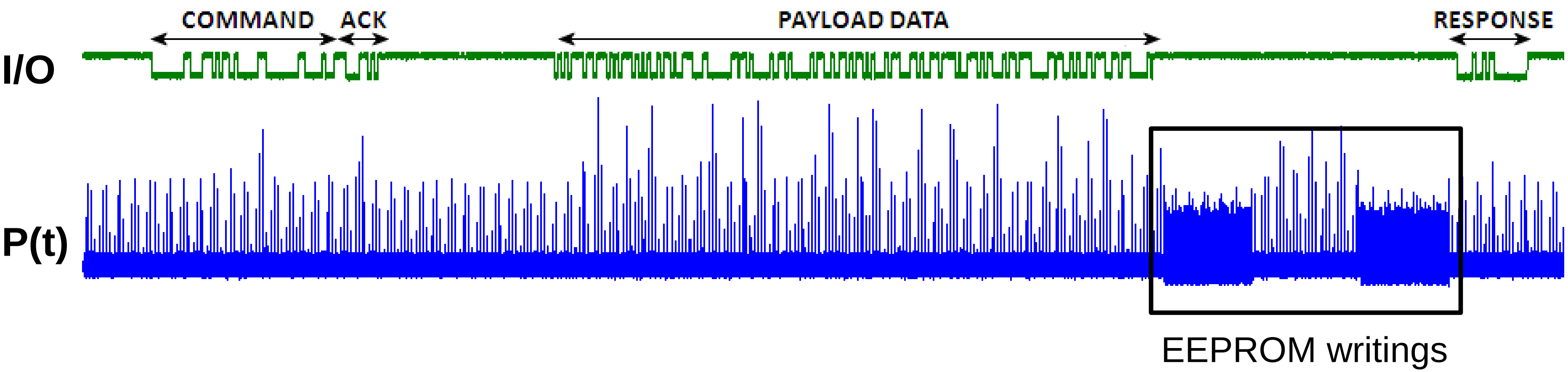



In the following all experiments carried out with <u>smart card</u>

# Analyzing power traces (I)

- Successful authentication
  - Before: $nt = AAC \| nt_1 \ldots nt_7$
  - After: $nt' = xFF \| at'_0 \ldots at'_6$

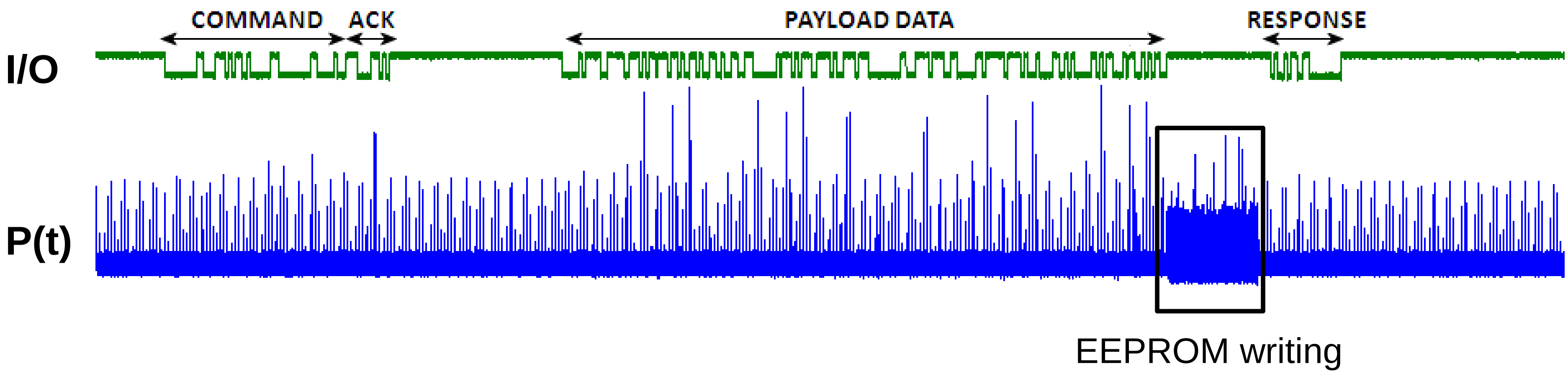
- Areas of interest



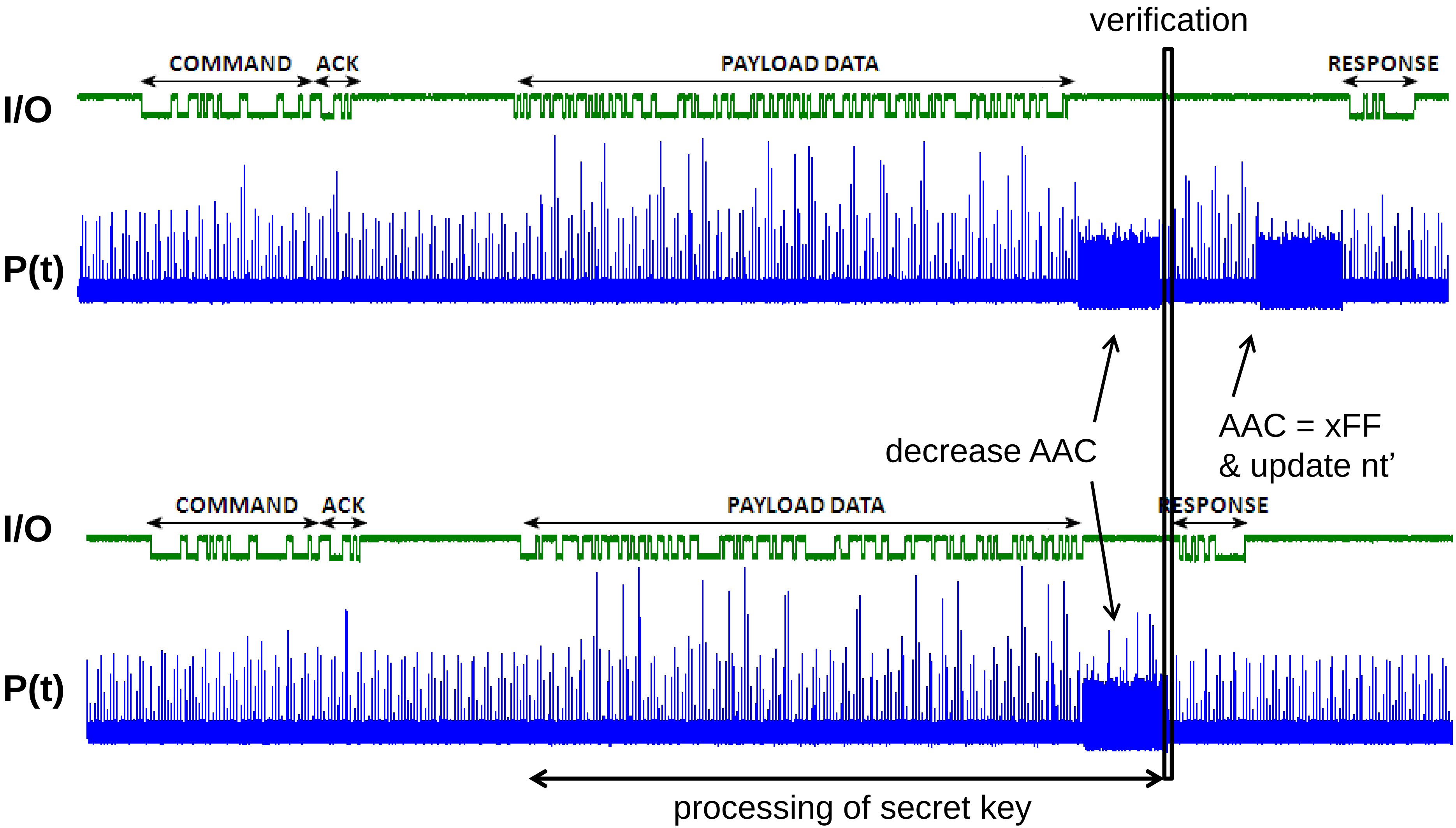
# Analyzing power traces (II)

- Unsuccessful authentication
  - Before: $nt = AAC \,||\, nt_1 \ldots nt_7$
  - After: $nt' = AAC' \,||\, nt_1 \ldots nt_7$

- Areas of interest

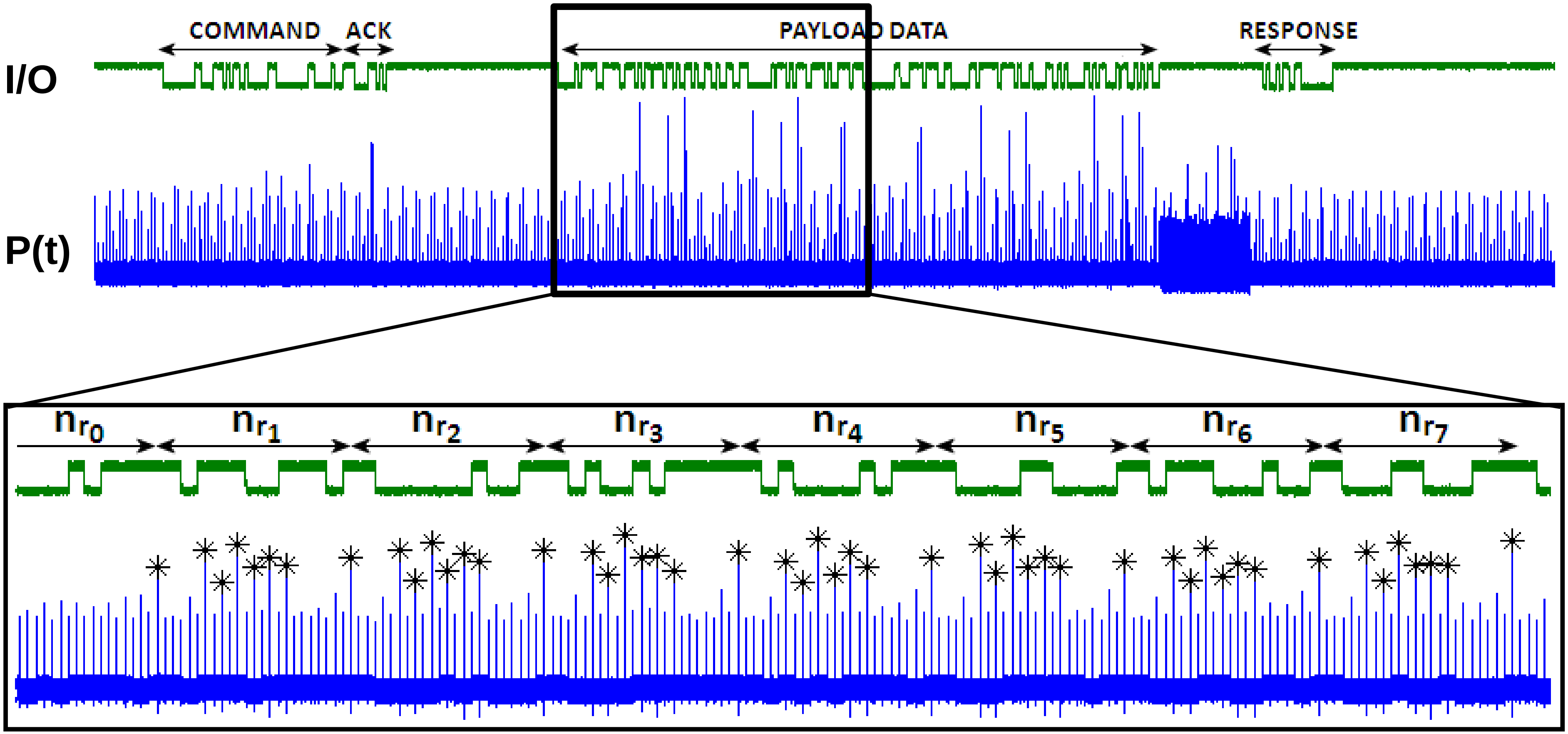


# Analyzing power traces (III)



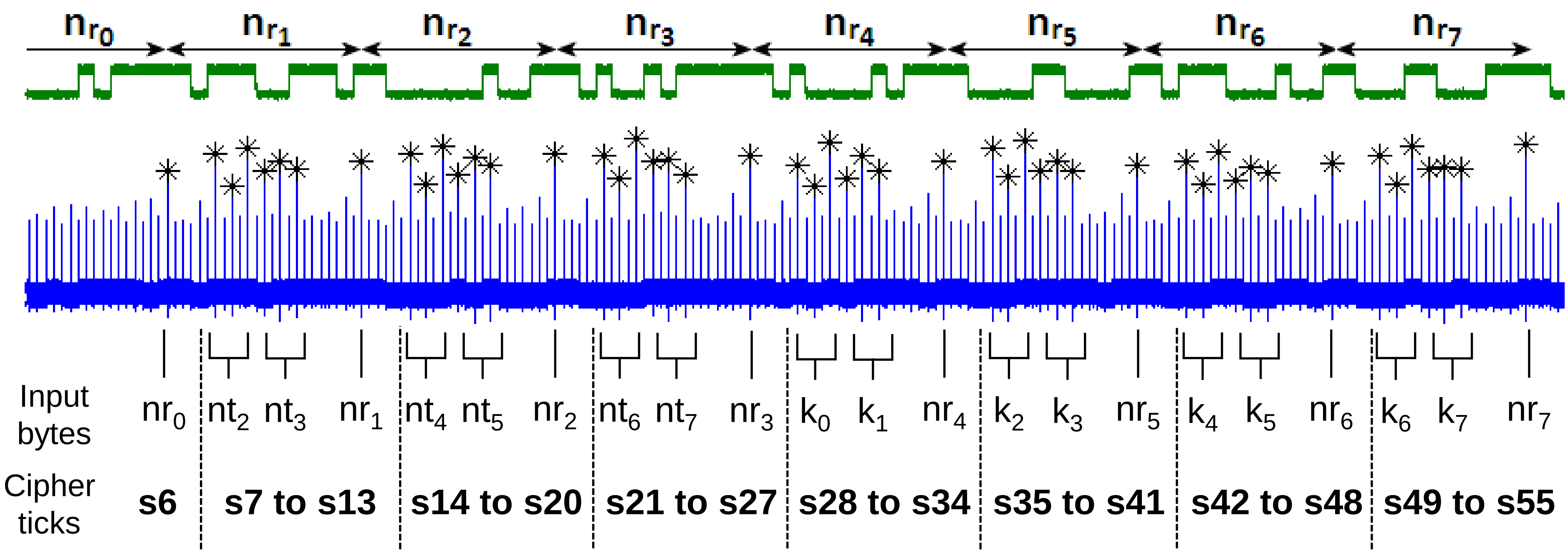
# Analyzing power traces (IV)

- Bytes of *nr* are fed into the cipher upon reception



# Analyzing power traces (V)

- Each power peak corresponds to a cipher tick
  - Nonces and key are scrambled into the cipher state during ticks 0 to 55

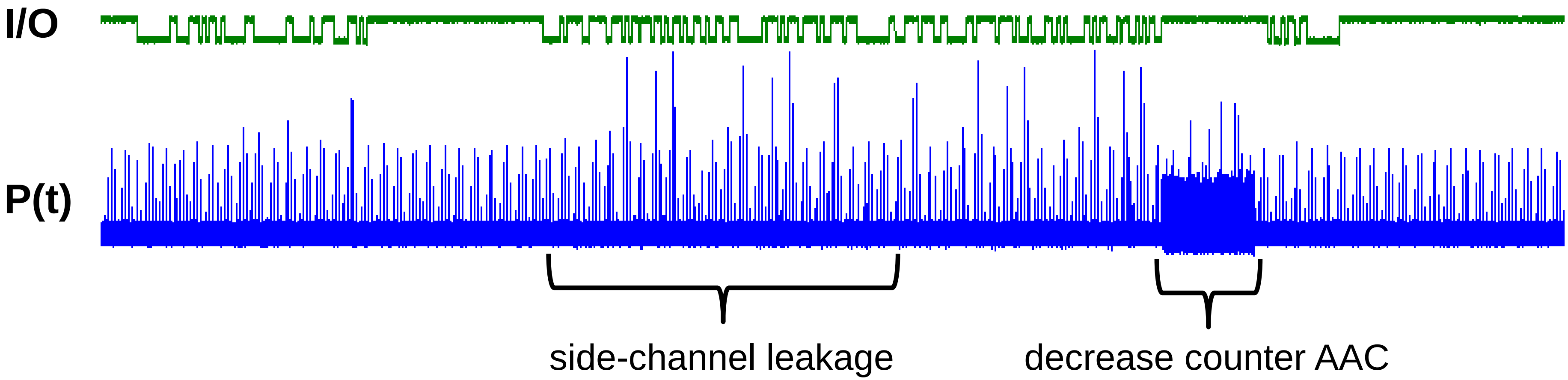


# Power Analysis

- Goal: use information leaked via power measurements to extract secret keys
- No countermeasures documented
  - But high claims on physical security
- Perhaps not needed?
  - Secrecy of cipher and authentication protocol
  - AAC limits the number of power traces to 3 before permanently locking the device
- Question
  - Is it possible to overcome the AAC counter?

# Enabling Power Analysis (I)

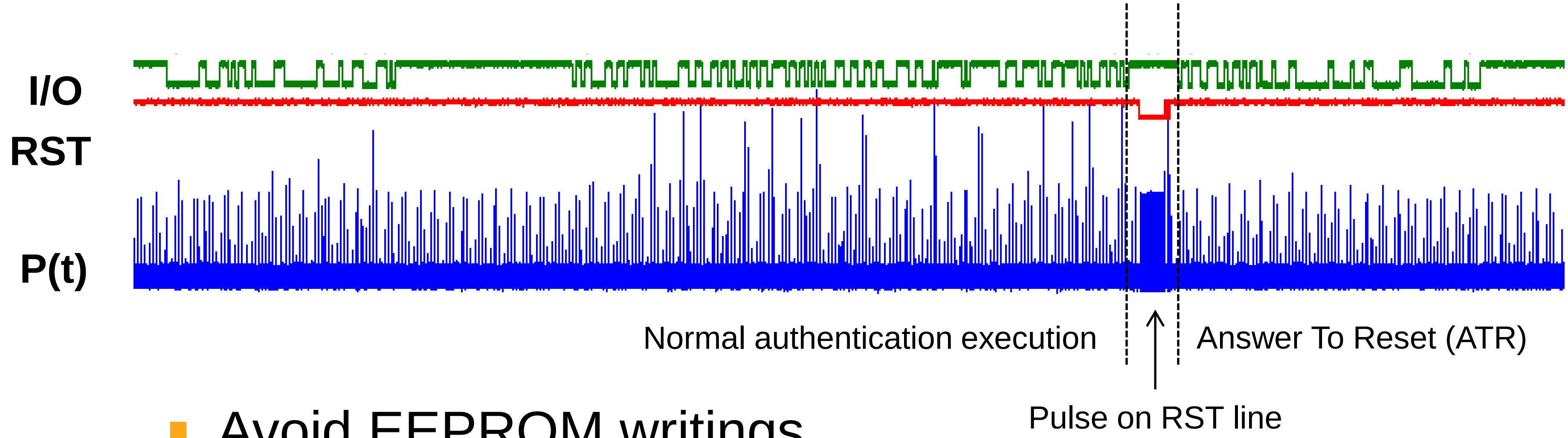
- Key observation




- Possible to collect the leakage information and prevent the counter from decreasing?

# Enabling Power Analysis (II)

- Sending a reset signal to the device




  - Avoid EEPROM writings
    - Counter AAC not decreased

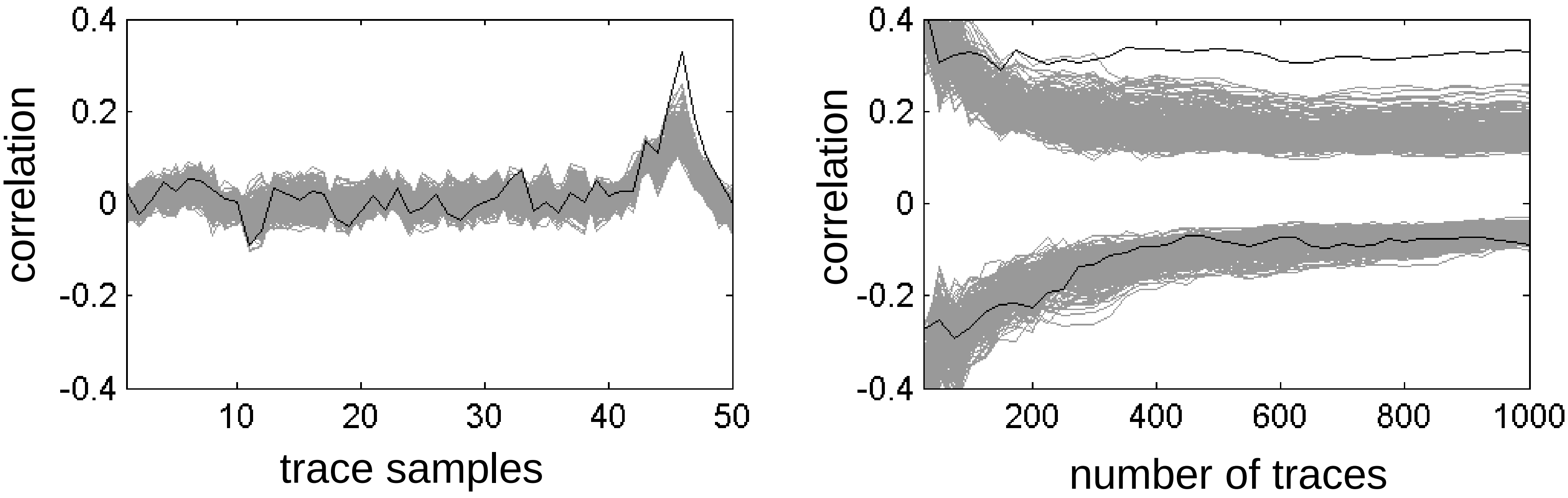
CryptoMemory in packaged ICs does not provide a RST line, but the same result is achieved by switching off VCC

# Power Analysis. Attack (I)

- Collect a set of 1000 power traces
  - Provide known random values for *nr*
  - RST pulse before EEPROM writings
- Peak extraction of cipher states
  - Compressed traces (only 50 points, states 6 to 55)
  - No need to align
- Power model: Hamming distance
  - Bit flips in cipher state between cipher transitions
- Distinguisher: Pearson's correlation coefficient

# Power Analysis. Attack (II)

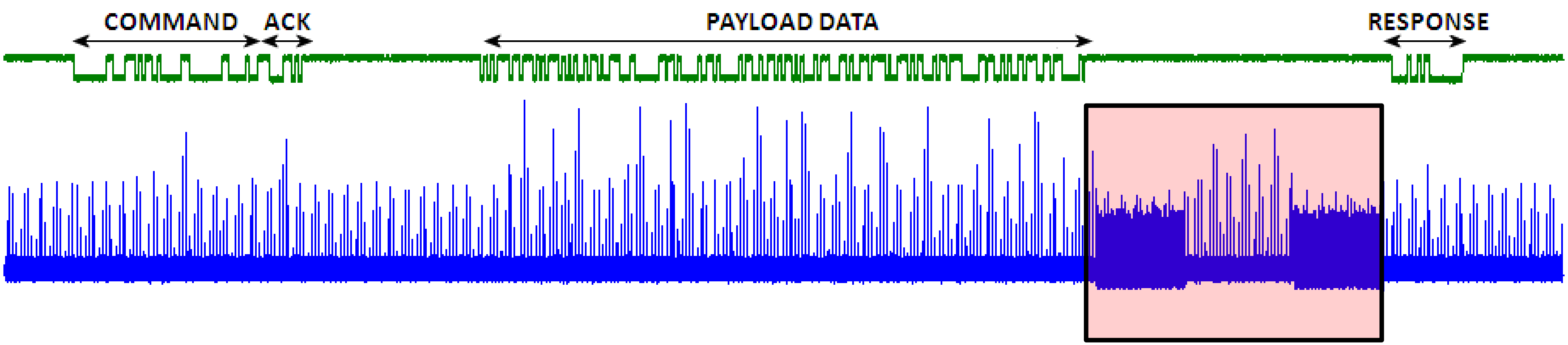
- Example for $k_6$ (required most traces)




- Improved attack requires only 100 traces
  - More details in the paper

# How to prevent bypassing of counters (I)

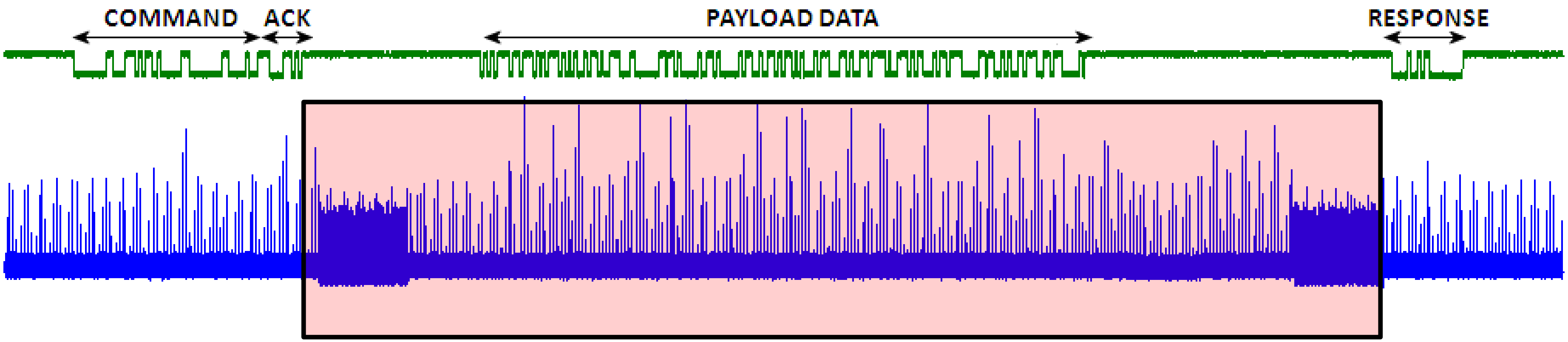
- Currently the comparison is protected…
  - Similar to SIM cards during PIN verification
- … but the processing of the secret *k* is not




protected operations

# How to prevent bypassing of counters (II)

- Solution
  - Decrease AAC upon authentication request
  - No major changes required (backwards compatible)



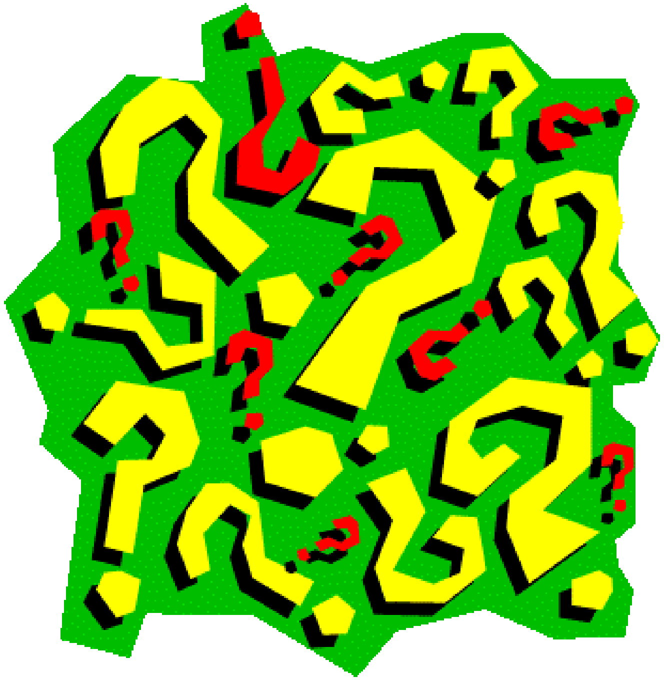

protected operations

# Conclusions

- Evaluation of CryptoMemory devices to non-invasive physical attacks (power analysis)
- High-level countermeasures
  - Secrecy of cryptographic tools
  - AAC counter to limit collection of power traces
- Reported flaw in handling of AAC counters
  - Key extraction in 20 minutes
  - Can be fixed while keeping backward compatibility

# Thanks for your attention!

- Questions?




[GvRVS10] F.D. Garcia, P. van Rossum, R. Verdult, and R.W. Schreur, “Dismantling SecureMemory, CryptoMemory and CryptoRF". In Proceedings of ACM CCS 2010, pp. 250-259. ACM Press, 2010.

[BKZ11] A. Biryukov, I. Kizhvatov, and B. Zhang, “Cryptanalysis of the Atmel Cipher in SecureMemory, CryptoMemory and CryptoRF. In Proceedings of ACNS 2011, pages 91-109. Springer, 2011.

# Support slides

- CryptoMemory. Security claims
  - “Tamper proof, metal shield layers, encrypted internal buses, defenses against timing and power supply attacks”

  - “Secure place for storage of sensitive information”
  - “Truly secure means of preventing counterfeiting and piracy”
  - “Can secure data against the most sophisticated attacks […], including physical attacks”
  - “[…] guarantee these values [authentication keys] can never be read”
  - “[…] designed to keep contents secure, whether operating in a system or removed from the board and sitting in the hacker’s lab”

*all quotes from publicly available documents*

# Support slides

- CryptoMemory device with AAC cleared
  - Authentication command is refused
  - Reader cannot send payload data (nr, ar)