RUB

RUHR-UNIVERSITÄT BOCHUM

The University of Electro-Communications

Horst Görtz Institute for IT-Security

On the Power of Fault Sensitivity Analysis and Collision Side-Channel Attacks in a Combined Setting

Amir Moradi, Oliver Mischke, Christof Paar,
Yang Li, Kazuo Ohta, Kazuo Sakiyama

Nara, Japan, 30 September 2011

UEC TOKYO hgi EMSEC

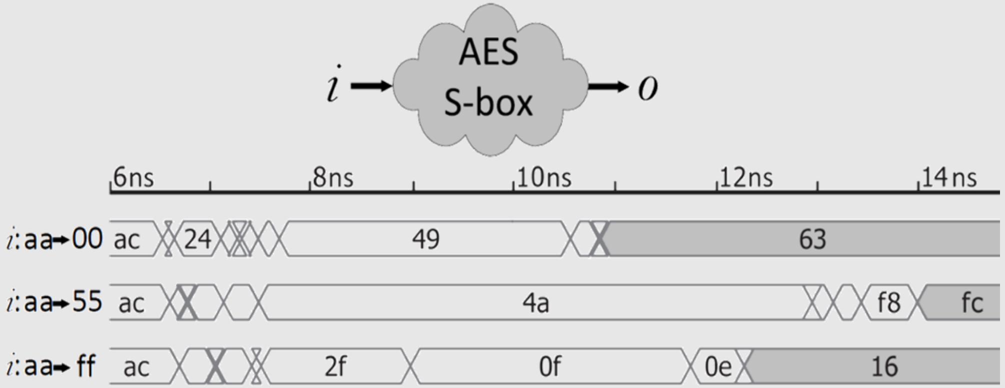
Outline

- Background
- Problems
- Solution of Bochum team
  - exploring colliding fault sensitivity information
- Solution of Tokyo team
  - examining distributions of faulty outputs

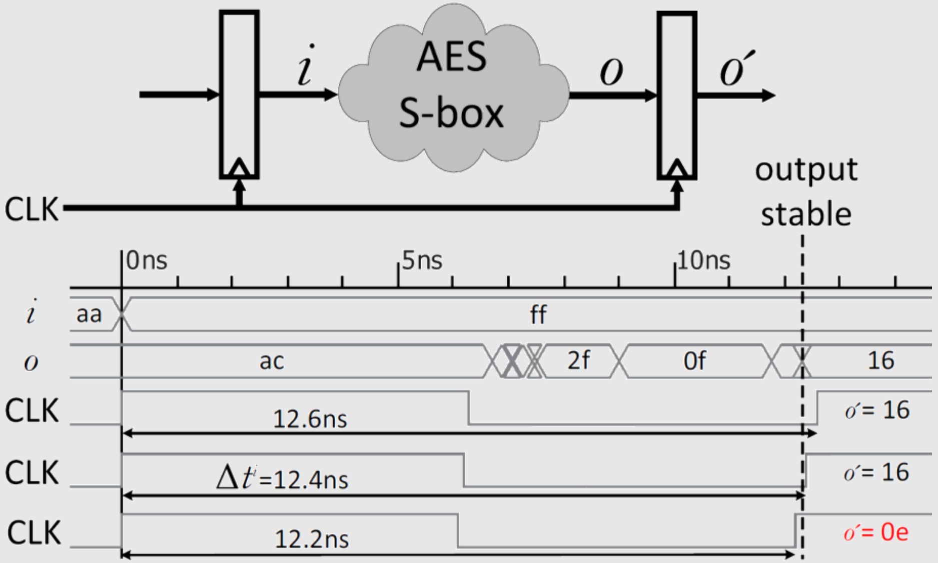
Background

- Fault Sensitivity Analysis by Yang Li (Tokyo team) at CHES 2010

- The main idea: *extracting the timing characteristics of a combinational circuit*



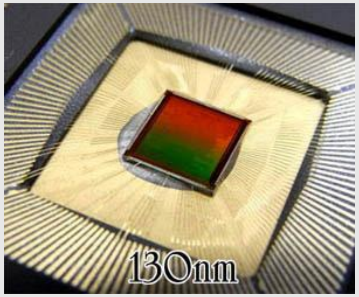
- How to extract?
    - Clock Glitch

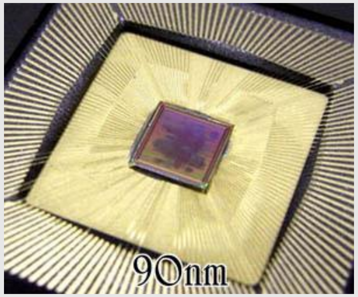


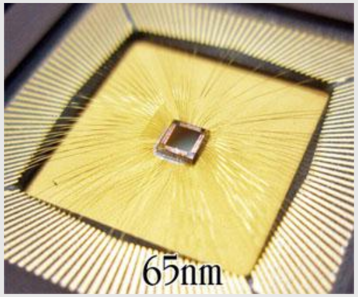
Background

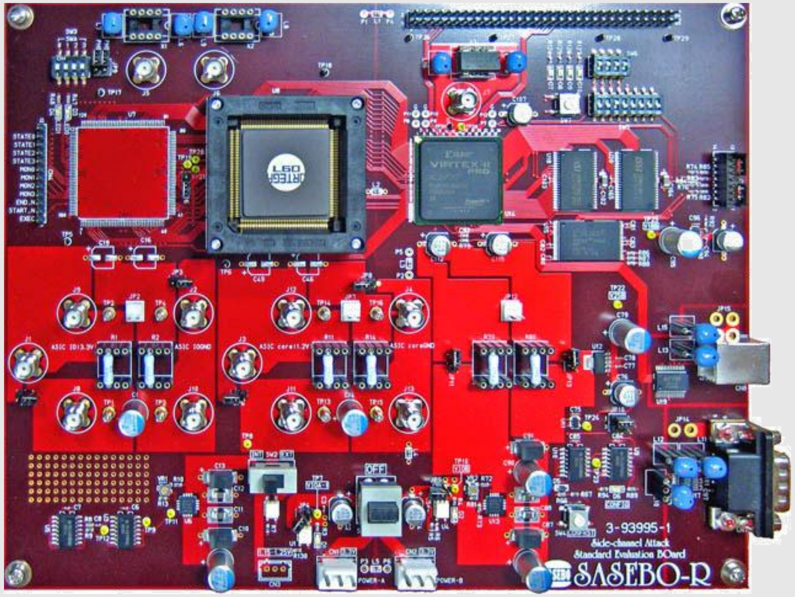
Target Platform

- SASEBO-R "ASIC version that has a socket to mount cryptographic LSIs"
- Three LSI chips *thanks to RCIS (Akashi Satoh)*

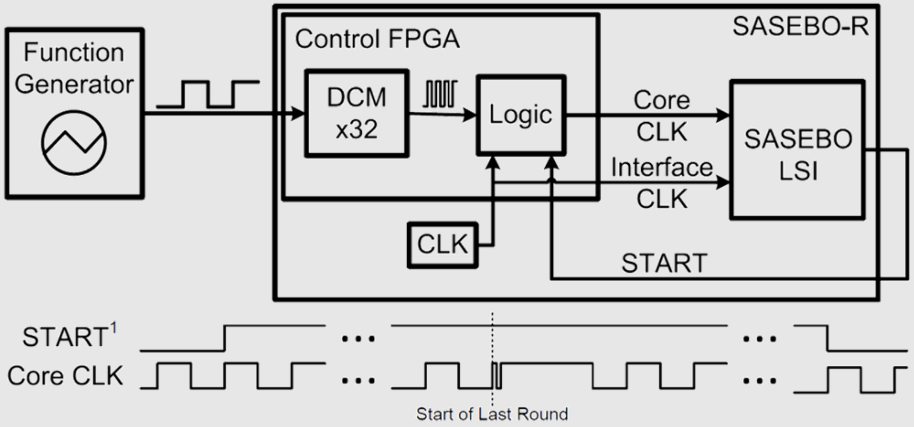






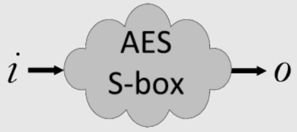


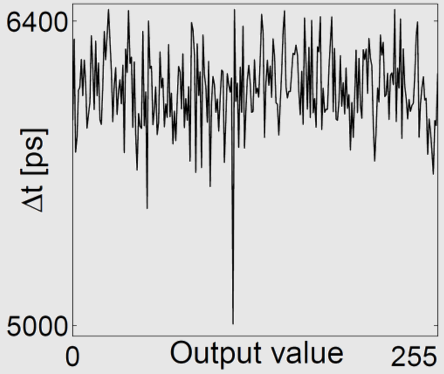
- Containing 14 AES cores
  - different S-boxes
  - DPA countermeasures
    - Masking
    - Logic style, …
  - Fault attack countermeasure



Problem

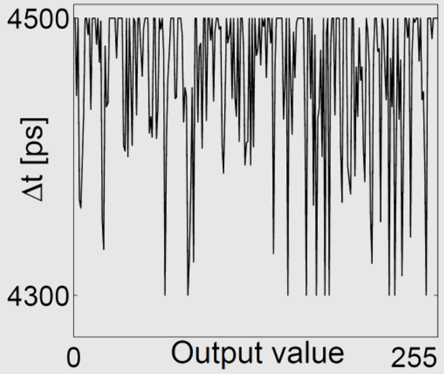
- Timing characteristics (fault sensitivities) are proportional to the processed values *collected from AES_Comp →*
  - An attack is possible knowing the underlying leakage model (HW/ZeroValue) *S-box*(0) *needs much less time →*





- What if the leakage model is not known?
- What if data randomization (masking) is involved?

collected from AES_MAO →

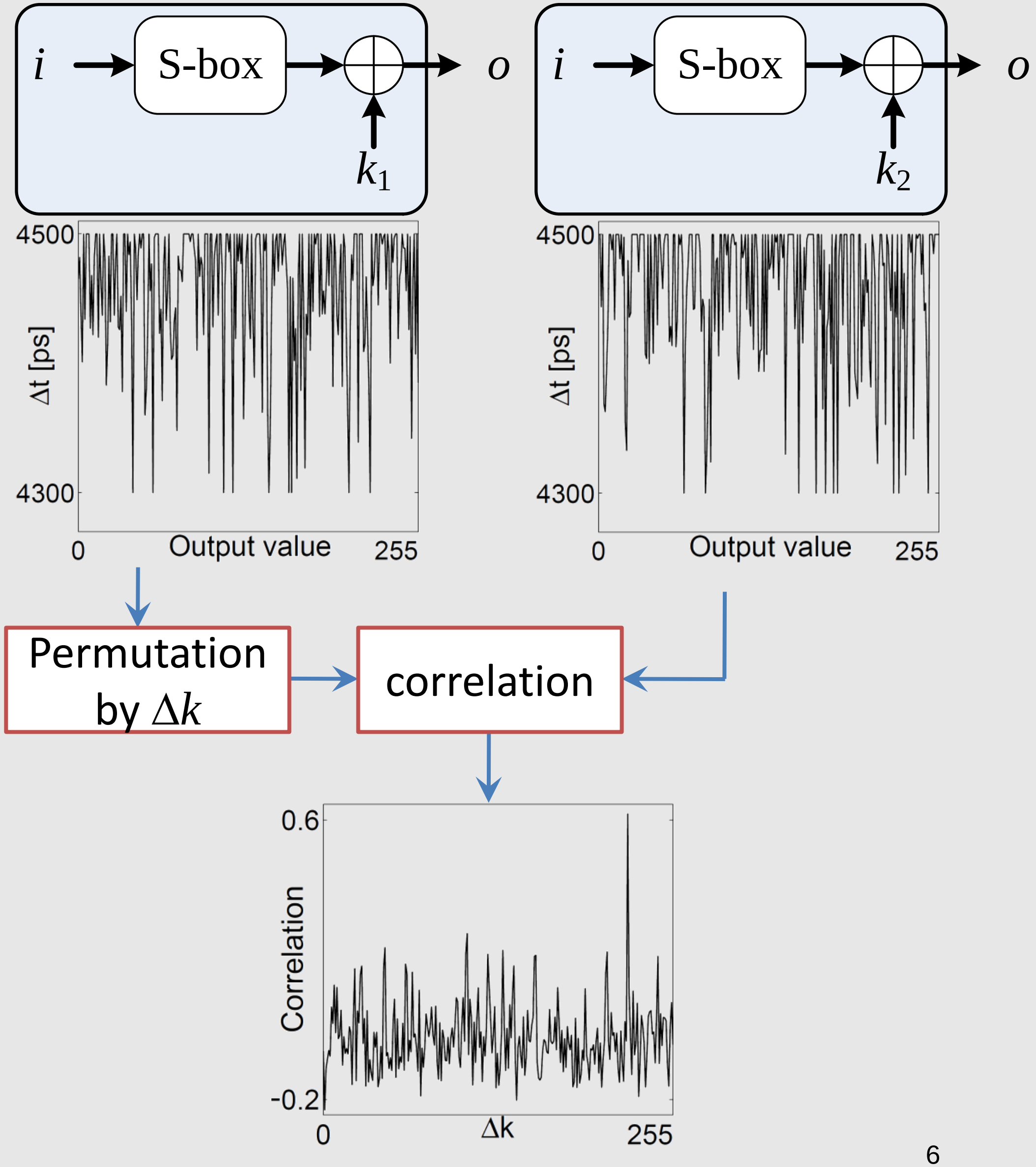


  - Template/profiling the device
  - Collision attacks
    - on timing characteristics (Bochum team)
    - on faulty output distributions (Tokyo team)

Collision Timing Attack (Bochum team)

Correlation Collision (CHES 2010)

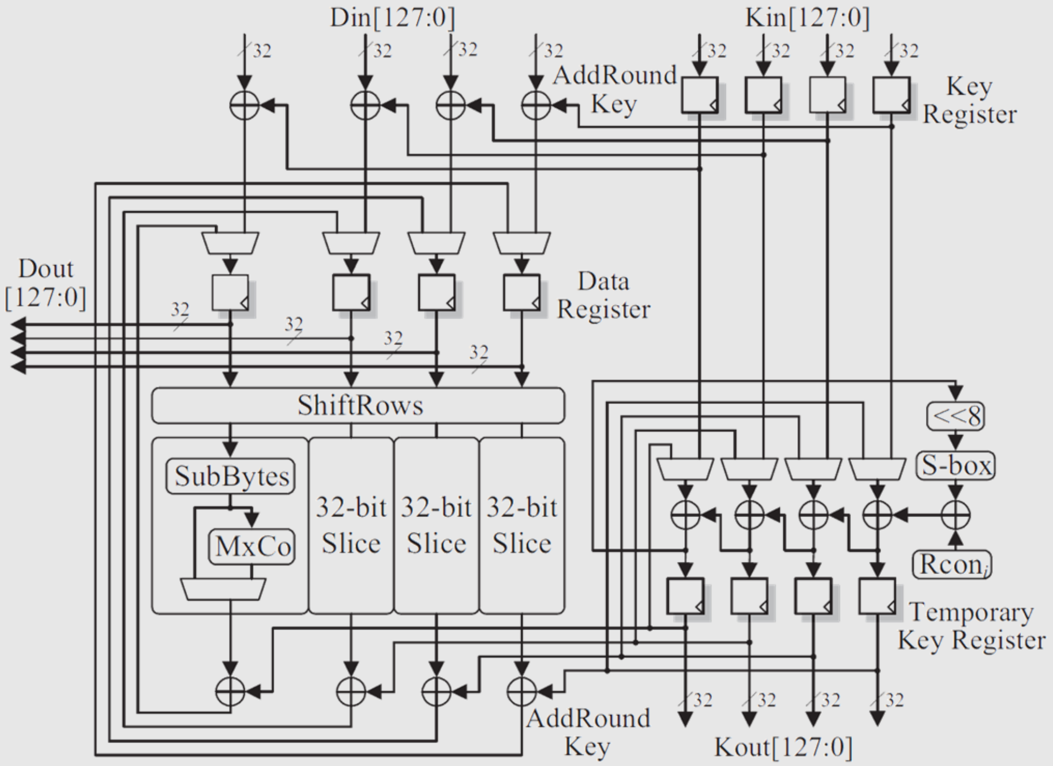
- Proposed to compare the side-channel leakage of two e.g., S-box instances
  - originally as a power analysis attack
- Here we use timing characteristics as side-channel leakage
- By means of correlation, $\Delta k = k_1 \oplus k_2$ is recovered
  - known as linear collision in AES



Target Architecture

Encryption module

- 128-bit datapath
  - one round per clock cycle
- The last enc. round is our target
  - since MxCo is not in the path
- no fault effect by *key schedule*
  - when clock glitch in last round
- every S-box faulty output can be seen (bitwise)

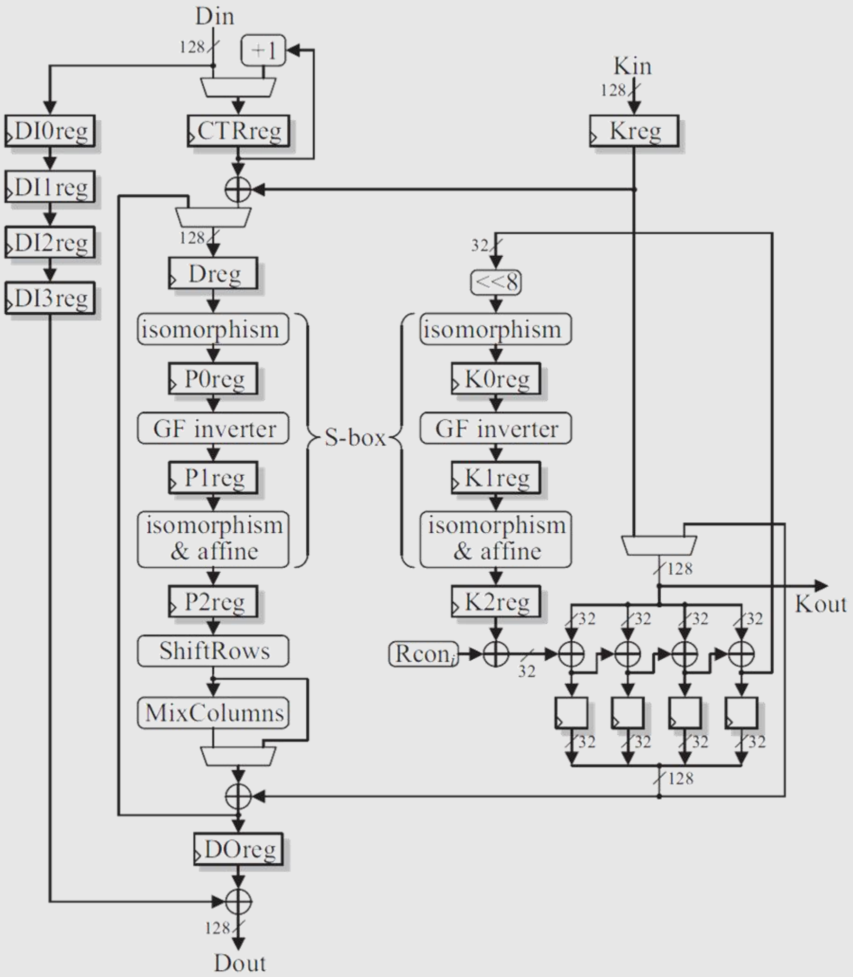


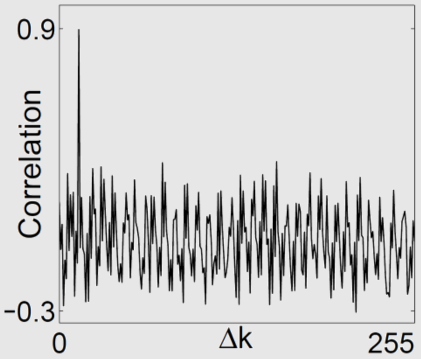
- Timing characteristics of every S-box instance can be extracted simultaneously
- The collision attack can run now

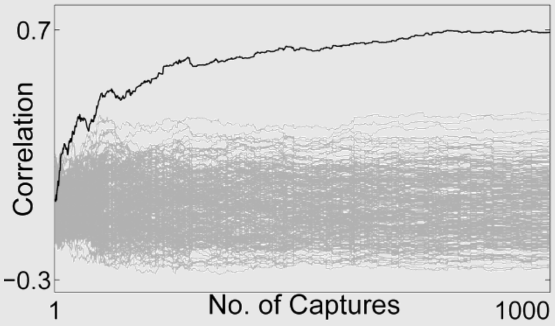
Results

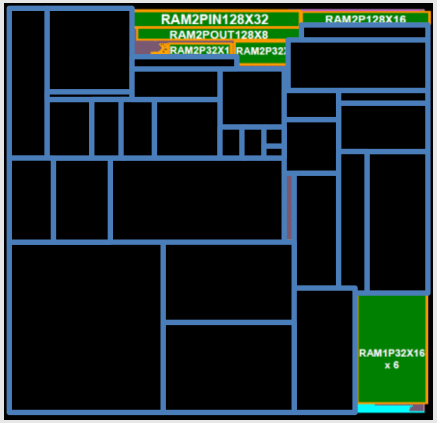
Unprotected

- All unprotected cores in all 3 technologies (except AES_TBL)
  - need a few (~100) captures to be completely broken
- AES_TBL
  - look-up tables
  - around 1 million!
- AES_CTR
  - counter mode
  - pipeline arch.
  - P=0, IV=rand
  - C=Cipher(IV)
  - ~100 captures





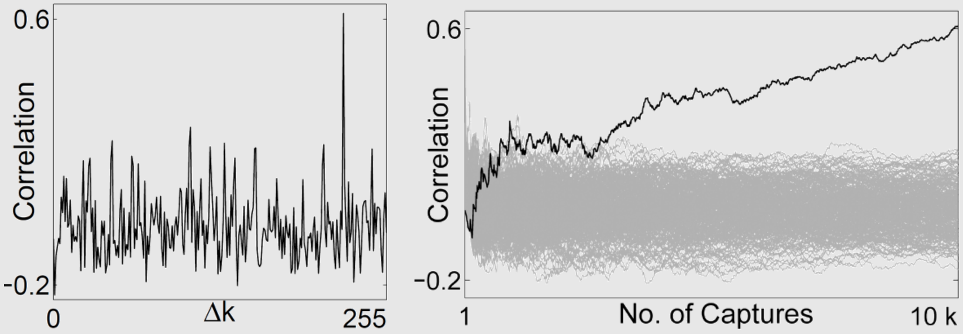




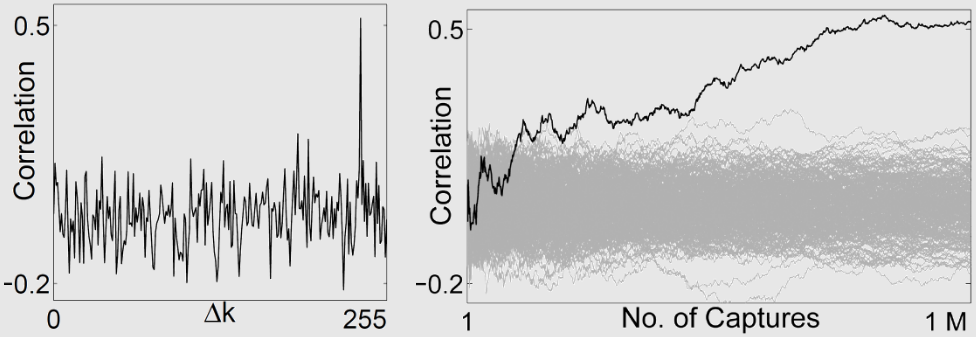
Results

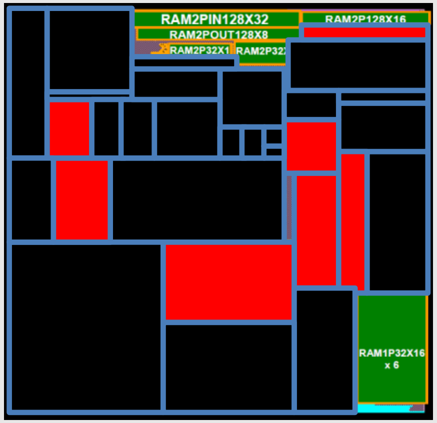
DPA-protected

- AES_MAO (masked S-box)
  - the same ach. as unprotected cores
  - the same attack scenario works
  - needs more captures ~4 k



- AES_TI (threshold implementation)
  - the same ach. as unprotected cores
  - not fulfilling all the requirements
  - 4 shares
    - (3 random mask bytes for each plaintext byte)
  - Needs much more captures ~500 k

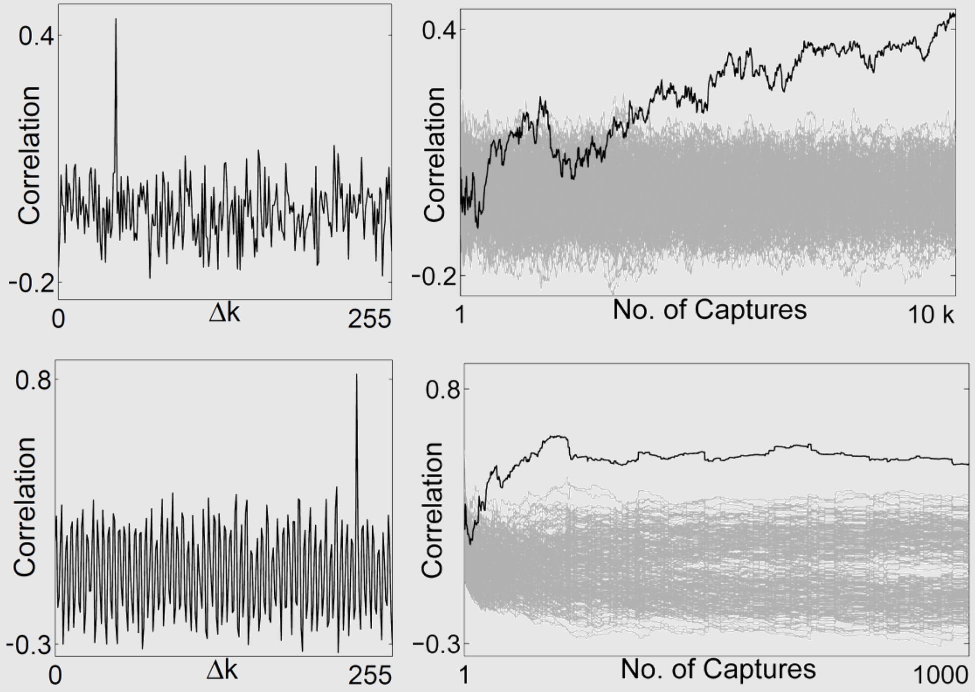




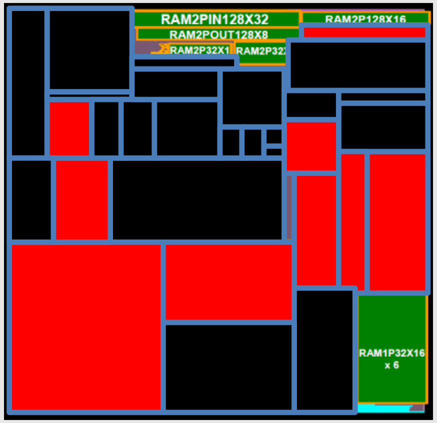
Results

DPA-protected (cont'd)

- AES_PR (pseudo RSL)
  - the same ach. as unprotected cores
  - S-box is divided into small parts
    - nonlinear parts by RSL
    - linear parts by CMOS
    - each part is enable controlled
  - the same attack works
    - needs high # of captures ~100 k



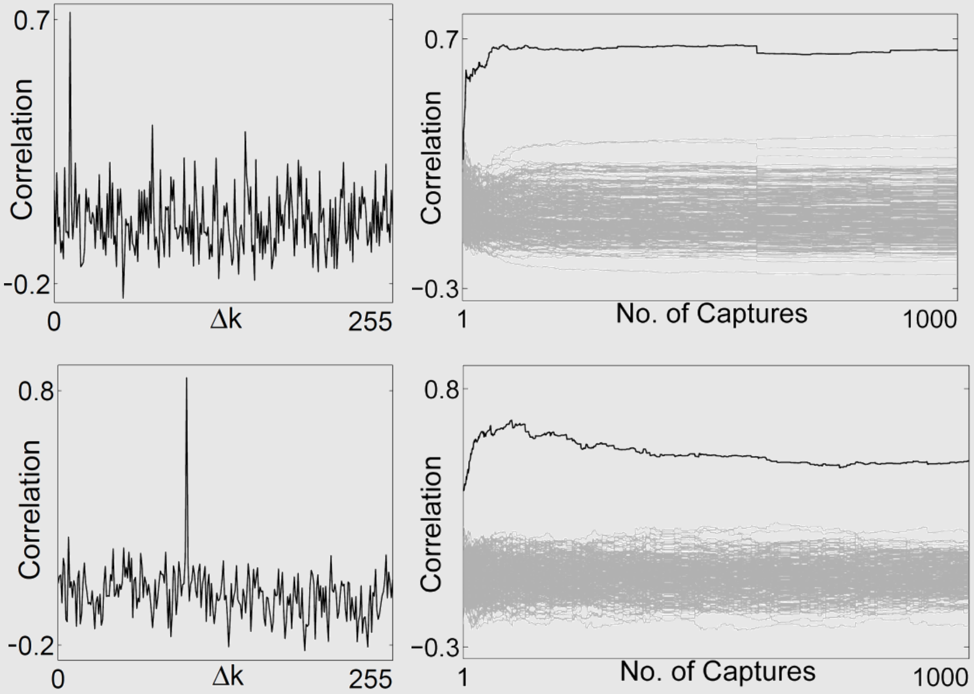
- AES_WO (similar to AES_PR but for evaluation purposes!)
  - shorter critical path
  - the attack works similar to unprotected cores
    - ~100 captures

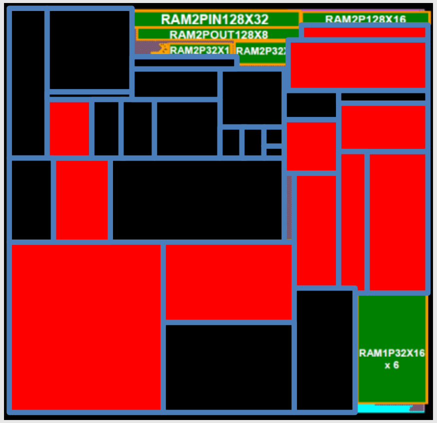


Results

DPA-protected (cont'd)

- AES_WDDL
  - the same arch. (128-bit datapth)
  - master-slave FF
    - two clock cycle per round
  - no fault (0->1) can be injected
    - because of the precharge phase
    - also reported by Yang Li (Tokyo team) at HOST 2011
- AES_MDPL
  - completely the same as AES_WDDL
- The same attack works on both
  - with less # of captures than unprotected cores < 100

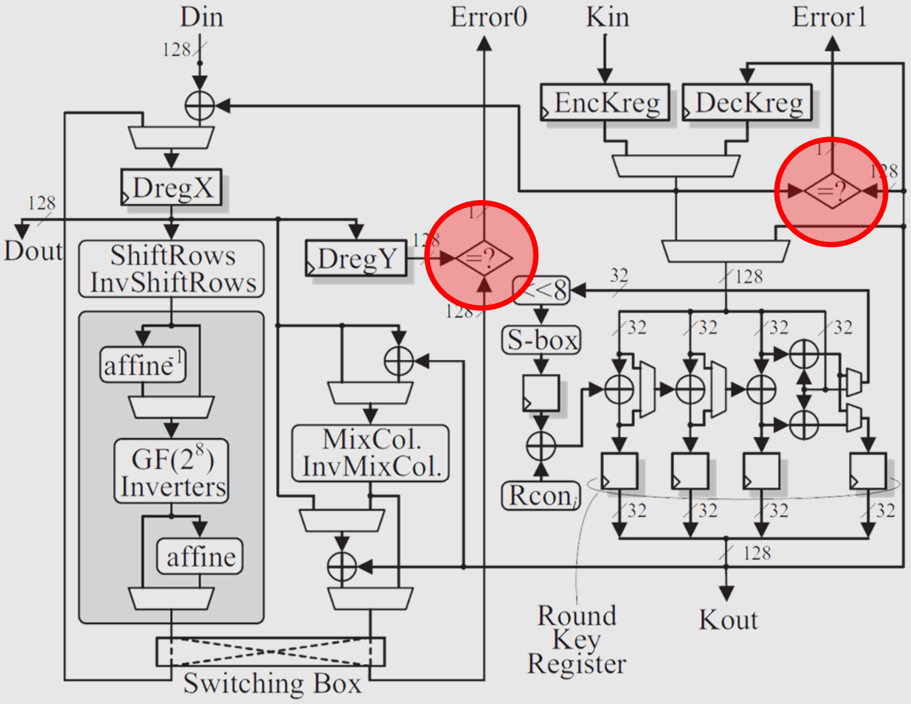


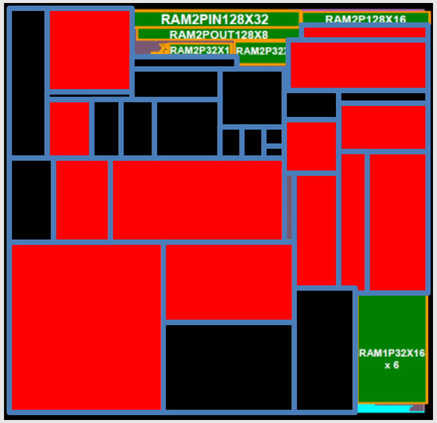


Results

Fault detection unit

- AES_FA (high-performance error detection scheme of CHES 2008)
- needs two clock cycles per round
- the performance is altered by comparison
- extraction of timing characteristics is not easy as before, we selected the first round
- bitwise and accurate timing characteristics cannot be obtained (there is only a fault bit)
  - the attacks work the same
    - of course using high # of captures ~50 k
    - and all key relations cannot be recovered
- In contrary to other cores, it can be extended to the next round
- final message: it can be completely broken by ~50 k captures





eprint.iacr.org/2011/162

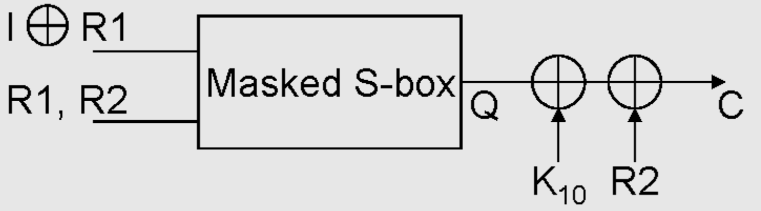
Results

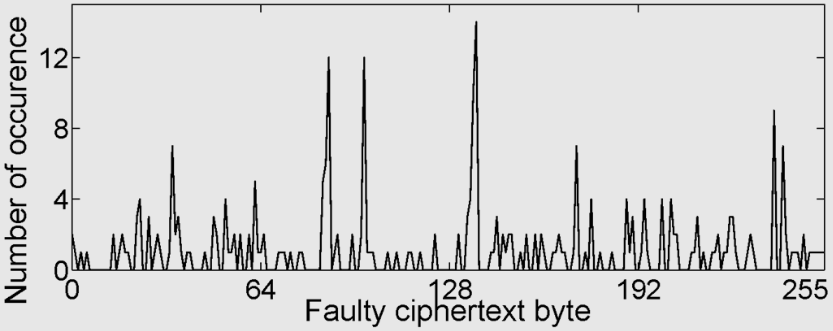
Difficulties

- Precise timing characteristics required [some times]
    - changing the clock glitch width by steps of ~5ps [not for all cores]
- A tons of engineering hours (~6 months to handle all cores in all technologies)
    - the clock glitch is canceled out by internal filters [PCB, FPGA, ASIC]
        - modifying the resistive/capacitive load of the clock signal
- and more
- Most of the problems can be softened by decreasing the core voltage
- In short, attacking the 65nm chip was easier than the others (different library)

Colliding Faulty Output Distributions (Tokyo team)

Concept



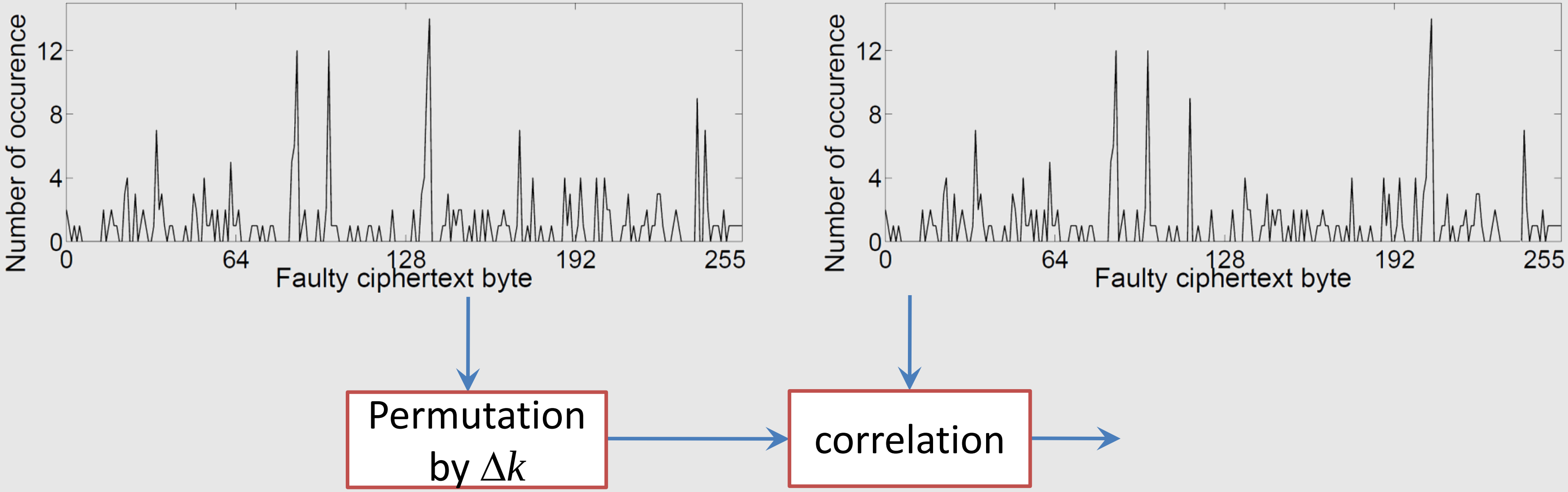


- Let's have a look at a masked S-box
- Fixing unmasked input (I) during clock glitch
  - faulty ciphertext bytes are not uniformly distributed →
- R2 and K_{10} are faster than "Masked S-box"
  - can be seen as fixed inverters/buffers
  - the distribution belongs to (I, K_{10}), therefore belongs to $Q \oplus R2$
    - Indeed a dependency between the distribution and unmasked data
- How to use this dependency in an attack?

Attack Scheme

Step by Step

- Set fault intensity so that ~50% of executions are faulty
  - Guess $\Delta k = k_1 \oplus k_2$ and select an appropriate plaintext so that $c_1 \oplus c_2 = \Delta k$
  - Collect two distributions of the faulty outputs at c_1 and c_2



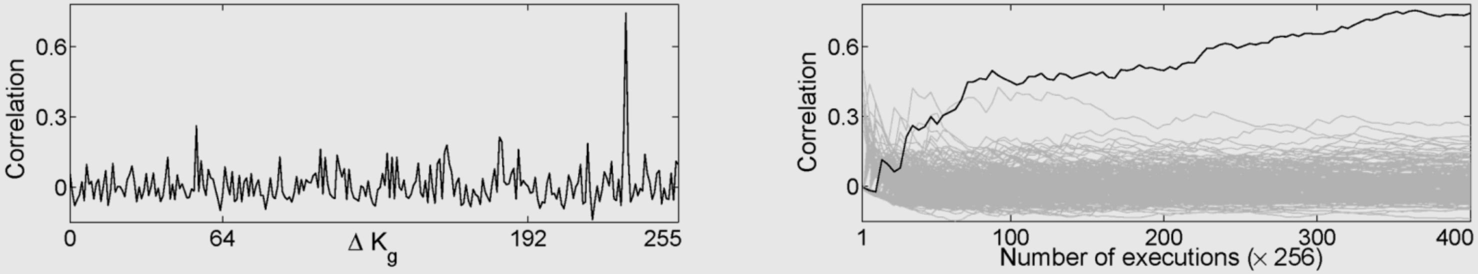
Permutation by Δk

correlation

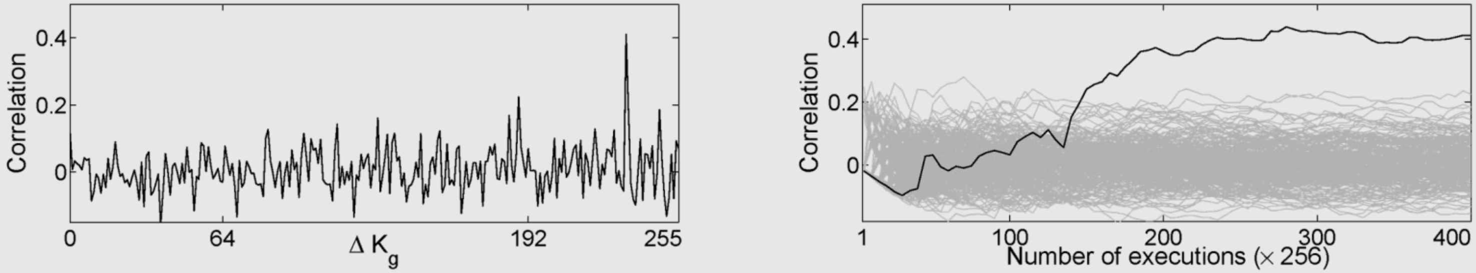
Attack Results

Two cores (130nm)

- AES_MAO
  - ~40 k (150 for each selected plaintext) executions are sufficient



- AES_TI
  - a bit more executions (~50 k) are required



Attack Details

Additional Observations

- How accurate should be the fault rate?
  - still worked if between 40-60%
- Attack works nonetheless with a very low amount of executions
  - lower requirements compared to DFA/FSA
  - It can be still reduced!
    - The goal is to have 256 pairs of distributions corresponding to all 256 linear differences between the ciphertext bytes
    - can be done by special ciphertexts (corres. plaintexts must be found)
      - one byte as 0x00, 0x01, ... , 0x0F
      - one byte as 0x00, 0x10, ... , 0xF0

RUB

RUHR-UNIVERSITÄT BOCHUM

Horst Görtz Institute for IT-Security

hgi EMSEC

moradi@crypto.rub.de

The University of Electro-Communications

Department of Informatics

UEC TOKYO

liyang@ice.uec.ac.jp