RUHR-UNIVERSITÄT BOCHUM

RUB

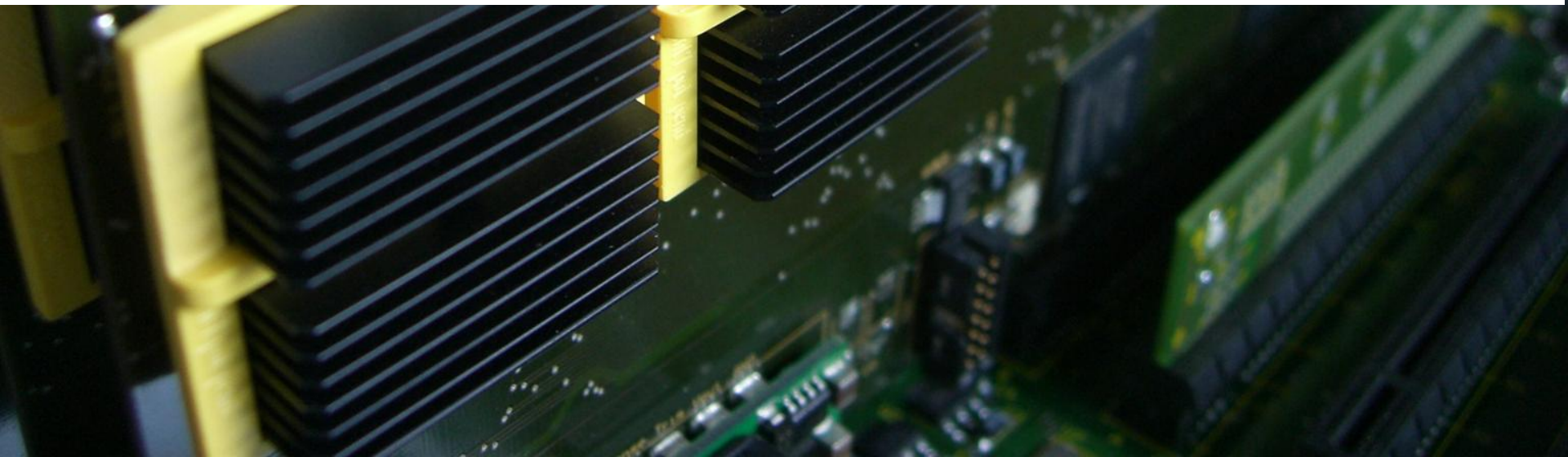
Glitch-Free Implementation of Masking in Modern FPGAs

Amir Moradi, Oliver Mischke
Horst Görtz Institute for IT-Security

escrypt
Embedded Security

EUROPÄISCHE UNION
Investition in unsere Zukunft
Europäischer Fonds
für regionale Entwicklung

Cryptarchi 2012, Chateau de Goutelas, France, 20/06/2012



Outline

- Background
- Problems
- Our Solution
- Evaluation
- Summary

Background SecureIP Project

- Objective is to develop and evaluate cryptographically and physically secure IP cores for FPGA-based systems
- Consortium of academic and industrial (SME) partners
  - Ruhr-University Bochum, Hardware Security Group
  - ESCRYPT – Embedded Security
- Duration of 3 years (Sep. 2011 – Aug. 2014)
- Publically funded by the European Commission and the German federal state NRW (IKT.NRW program)
- Further information under *https://www.secure-ip.org*

EUROPÄISCHE UNION
Investition in unsere Zukunft
Europäischer Fonds
für regionale Entwicklung

escrypt
Embedded Security

Background

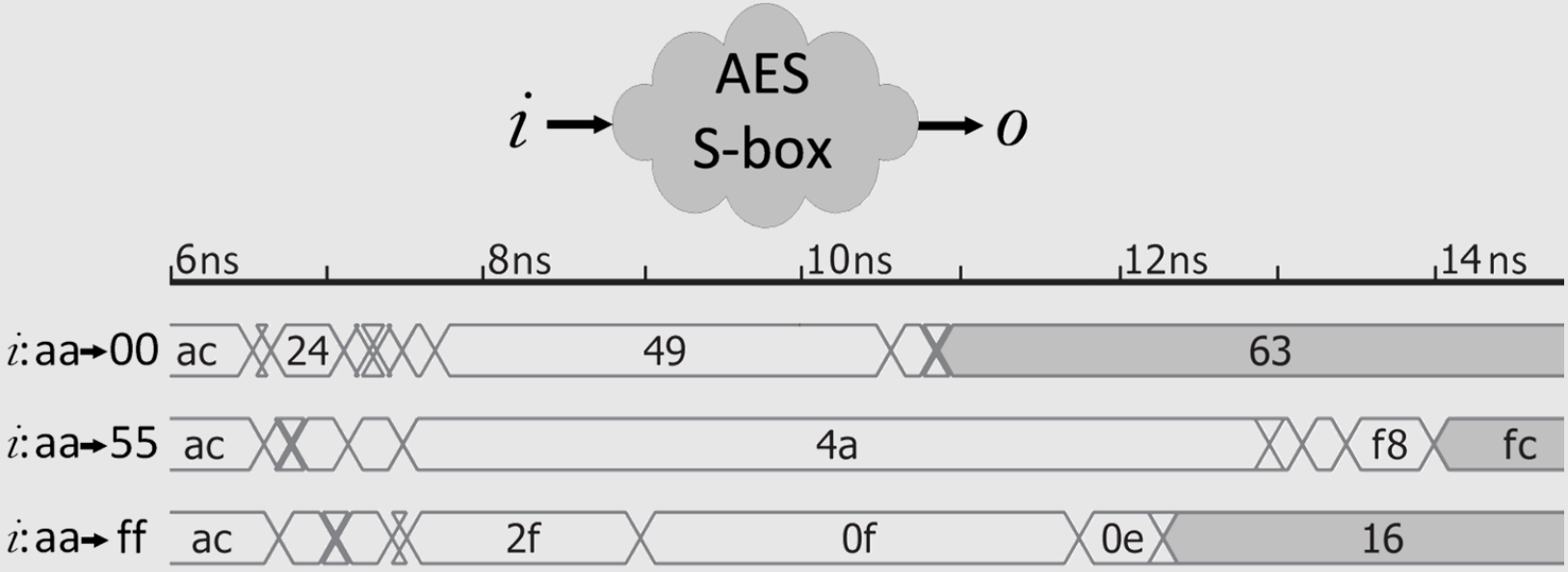
- Hardware countermeasures against power analysis
  - Many solutions from algorithmic countermeasures to special logic styles ((i)MDPL, WDDL, …)
  - But what about an FPGA?
    - Hardware is fixed -> mostly algorithmic countermeasures
    - Masking schemes: additive, multiplicative, affine
    - Problem of masking in hardware not solved!

      -> Glitches!

Problem: Glitches!

- Output of gates are not simply switching their signal level once
- Different arrival time causes multiple changes
- Glitches are passed to the next element

  -> even more glitches!



How to solve the problem of glitches?

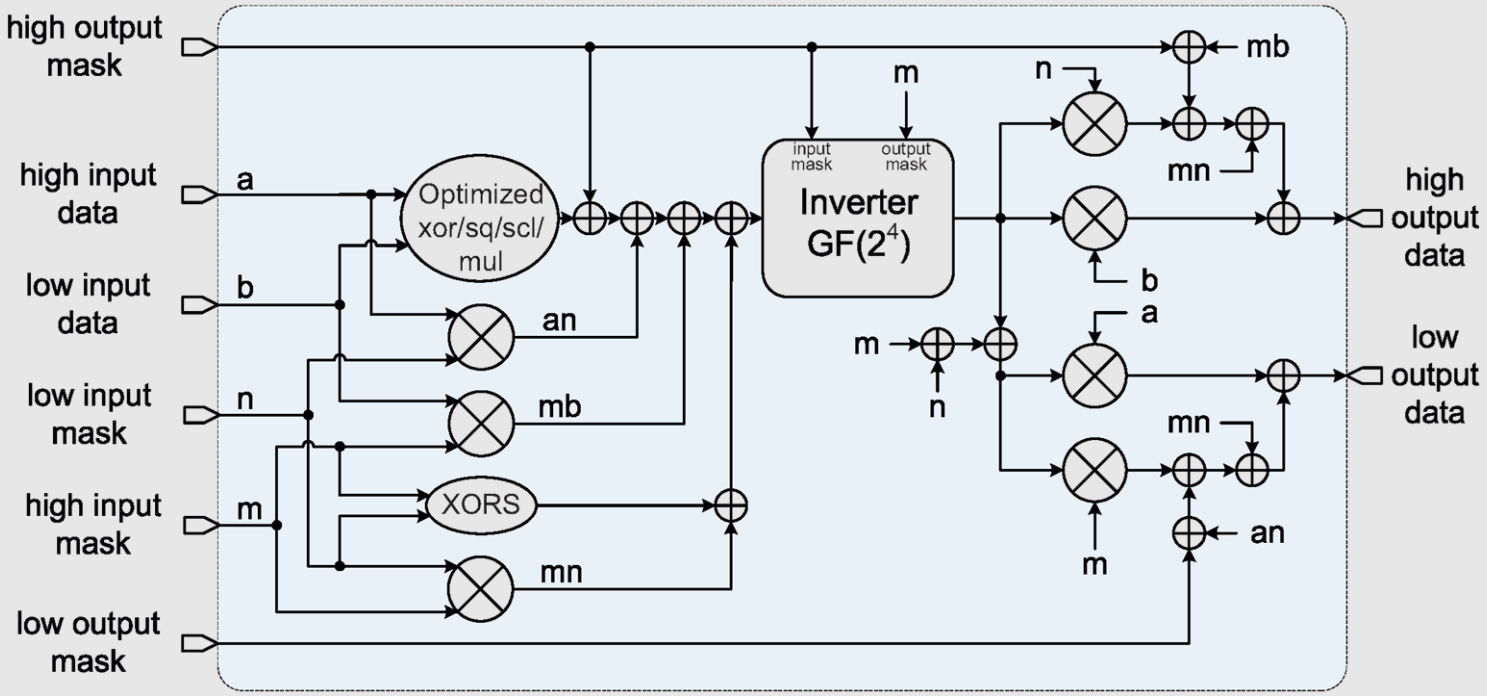
- Available so far:
  - Schemes which are resistant to glitches
    - TI (Nikova et al., ICICS 2006&2008/JoC 2/2011)
      - Not trivial, esp. for larger Sboxes
      - AES version could not yet be made
      - All 3x3 and 4x4 Sboxes at CHES2012
    - MPC (Prouff/Roche, CHES 2011)
      - Large time and area overhead
      - Not practically evaluated yet
- Is there another way for FPGAs? YES!

Our Solution

- Let‘s try a different approach:
  - Don‘t try to achieve glitch resistance, avoid glitches!
  - Use most compact masked AES Sbox to date (Canright/Batina)
  - Manually map modules to FPGA ressources
  - Add enable signal/registers to separate each LUT stage

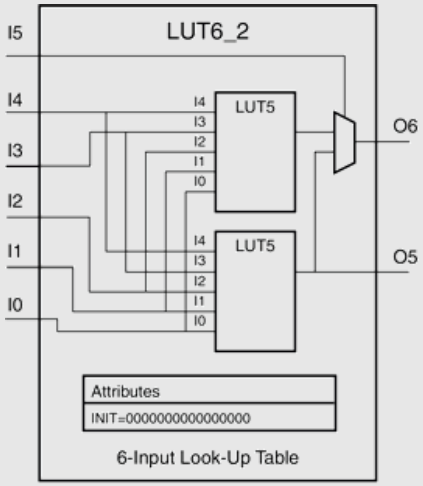
Masked Sbox

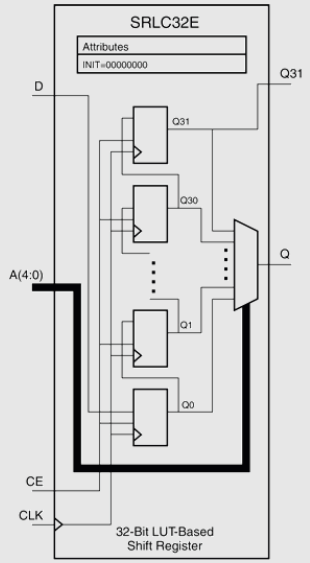
- Canright/Batina (ACNS 2008), corrected version: eprint Archive Report 2009/011
- Tower field approach, additive masking

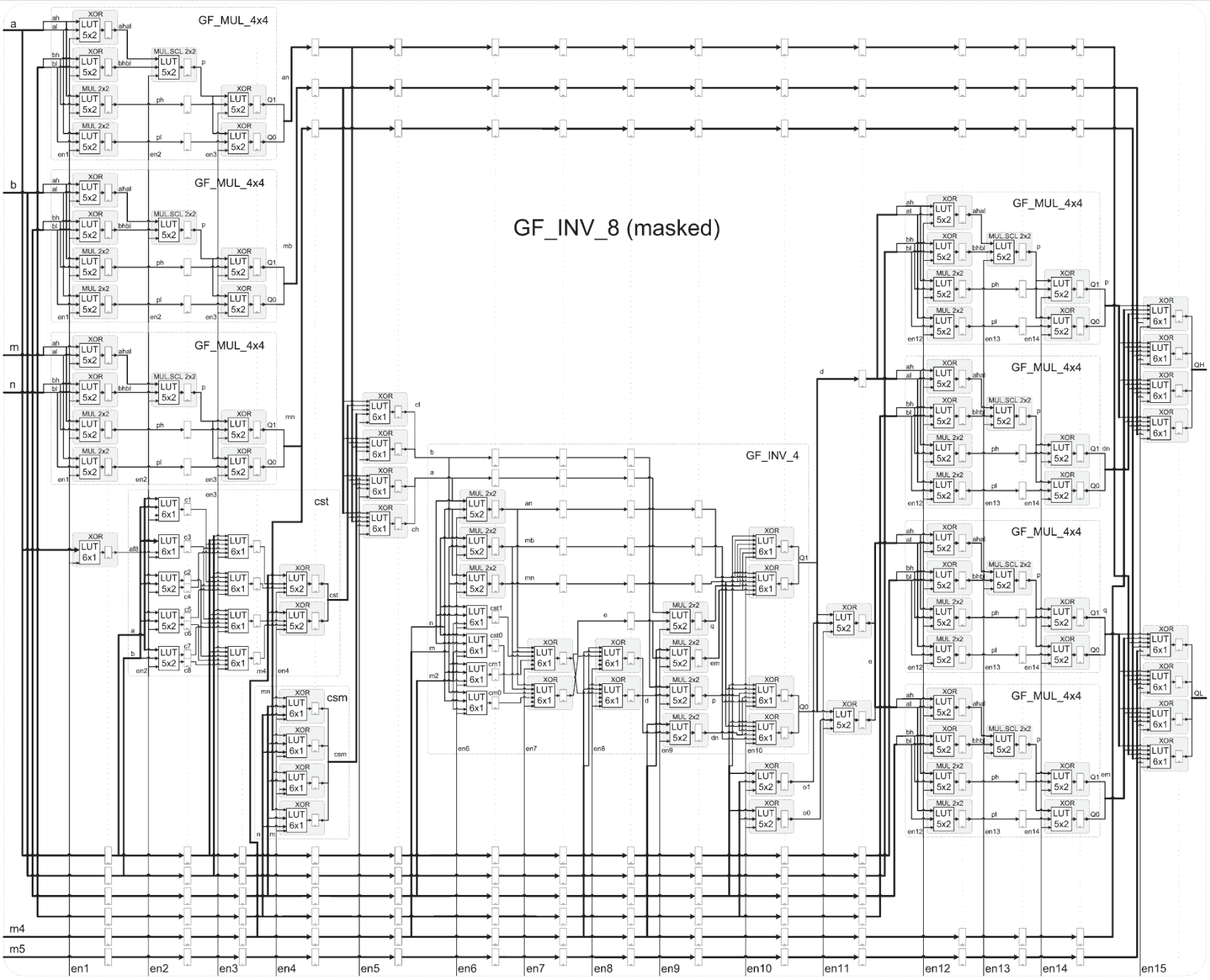


FPGA Ressources

- Many-to-one Look-Up-Tables (LUTs)
- Storage bit is selected by MUXs
- One input is used as enable
  - Output is forced to zero if bit is 1
  - Does it matter which bit is used?
    - Yes!
  - Use first MUX stage after storage bits
    - Bit is I0 (deduced from SRLC32E Arch.)
  - Fix PIN assignments by constraint!





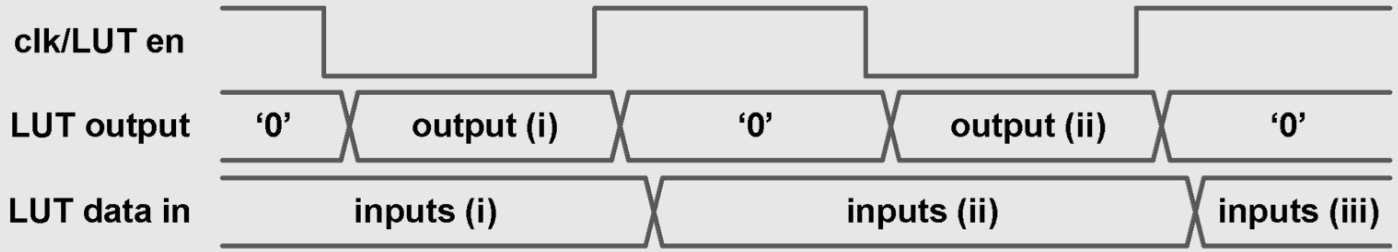

GF_INV_8 (masked)
GF_MUL_4x4
GF_INV_4
cst
csm
XOR
LUT
5x2
6x1
MUL 2x2
MUL_SCL 2x2
a
b
m
n
m4
m5
QH
QL
en1
en2
en3
en4
en5
en6
en7
en8
en9
en10
en11
en12
en13
en14
en15

Design Profiles

- 6 exemplary designs on SASEBO GII:
  - (1) Original with synthesizer optimization
  - (2) Original without optimizations
  - (3) Our modified design, no enables, no regs/pipelining
  - (4) No regs/pipelining, but enabling all stages sequentially
  - (5) Regs/pipelining, but always enabled
  - (6) Regs/pipelining and active low enable
    - Utilizes special routing of the clock tree

Timing of Profile 6

- Timing behaviour of LUT in- and output for Profile 6



  - Active low enable tied to clock signal
  - Clock high: output is forced to 0, input becomes stable
  - Clock low: stable input is distributed
  - Rising edge clock: output forced to zero before new inputs arrive at the LUT input

Synthesis results

Profile	Max. Freq.	#LUTs	#FFs	Latency (#clocks)	Throughput (16 Inv. /s)
1	105.519MHz	99	0	0	6 594 937
2	56.504MHz	244	0	0	3 531 500
3	88.300MHz	100	0	0	5 518 750
4	641.026MHz	100	0	30	1 335 471
5	641.026MHz	100	649	15 (*pipe'd*)	20 678 258
6	320.513MHz	100	649	15 (*pipe'd*)	10 339 129

- Comparable LUT requirements with unsecure versions
- Higher troughput of secure profile 6 is paid by large amount of registers

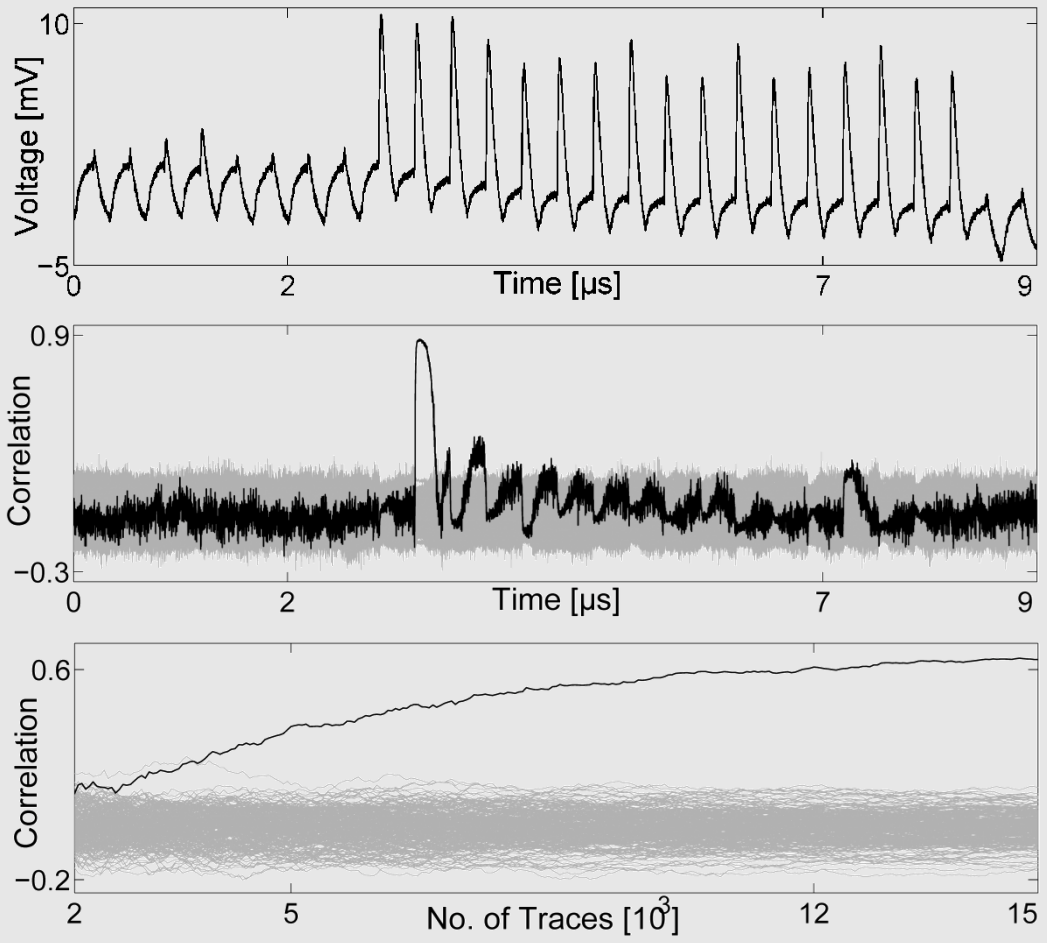
Evaluation

- Single re-used Sbox instance

  -> perfect for correlation collision attacks (Moradi et al., CHES 2010/Eurocrypt 2012)
- Focus on the 3 most interesting profiles:
  - (3) Our modified design, no enables, no regs/pipelining -> big combinational circuit
  - (5) Regs/pipelining, but always enabled -> hinder glitch propagation
  - (6) Regs/pipelining and active low enable -> no glitches and fast execution

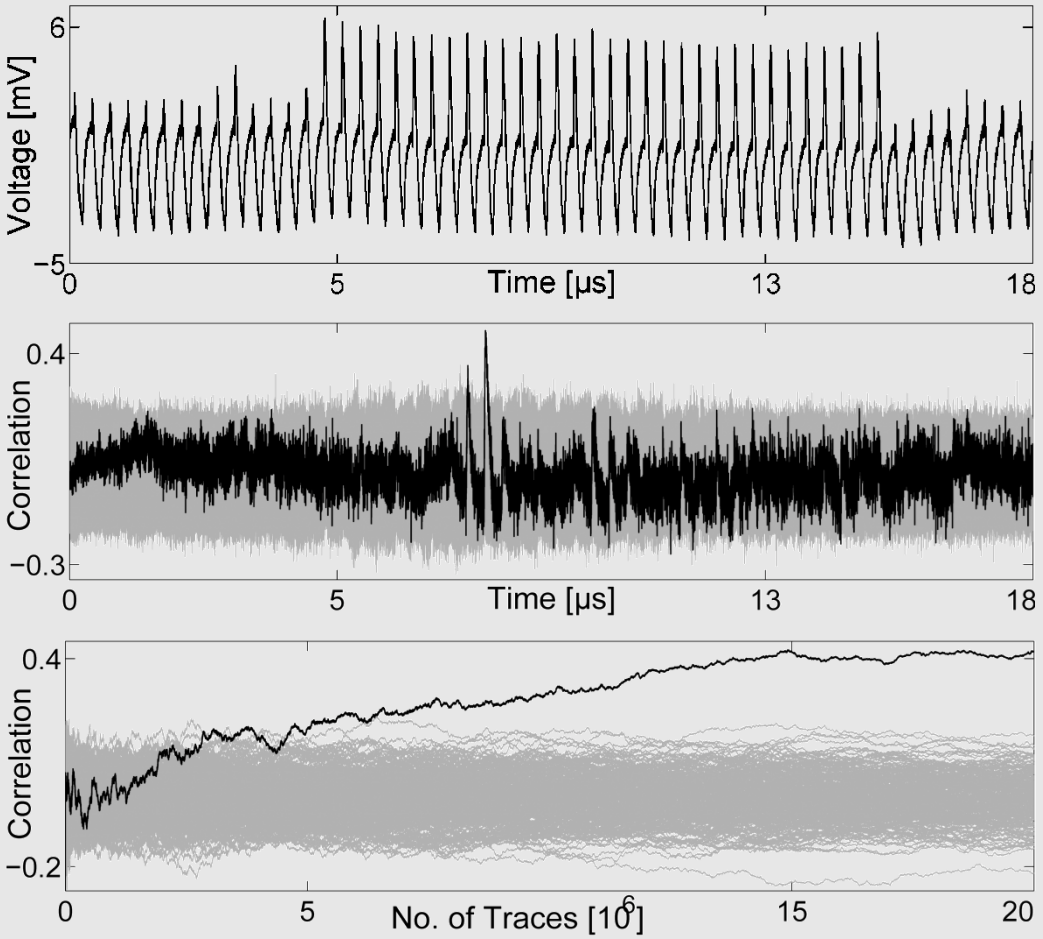
Profile 3: Combinational Circuit

- Full SubBytes takes 16 clock cycles
- Very high correlation after 50k measurements
- 5k measurements are sufficient



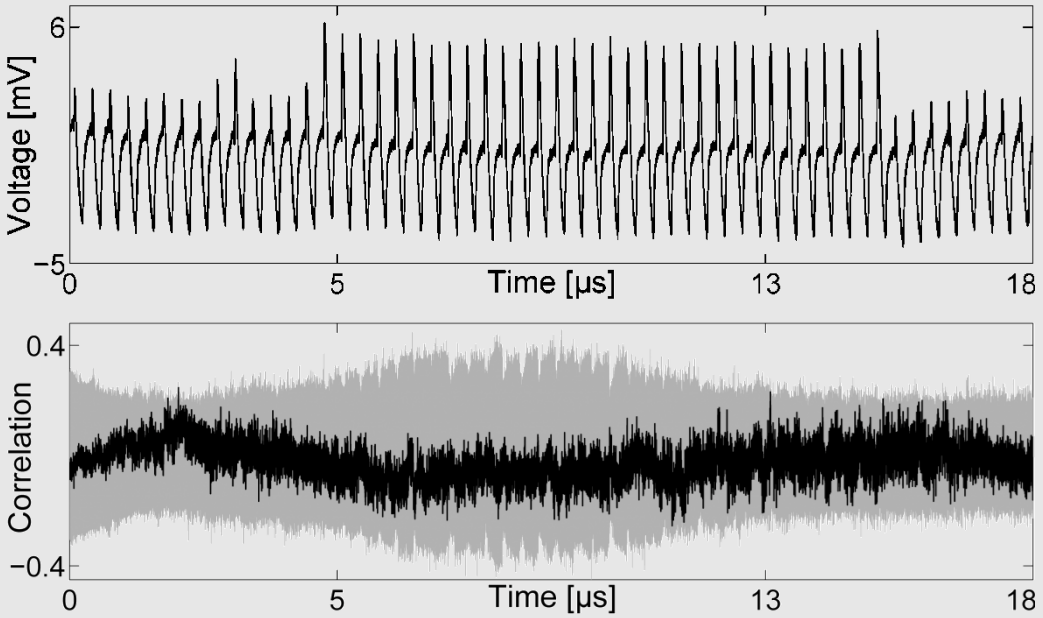
Profile 5: Pipelining

- Full SubBytes takes 31 clock cycles
- Depicted successful result after 20M measurements
- Already 8M measurements are required



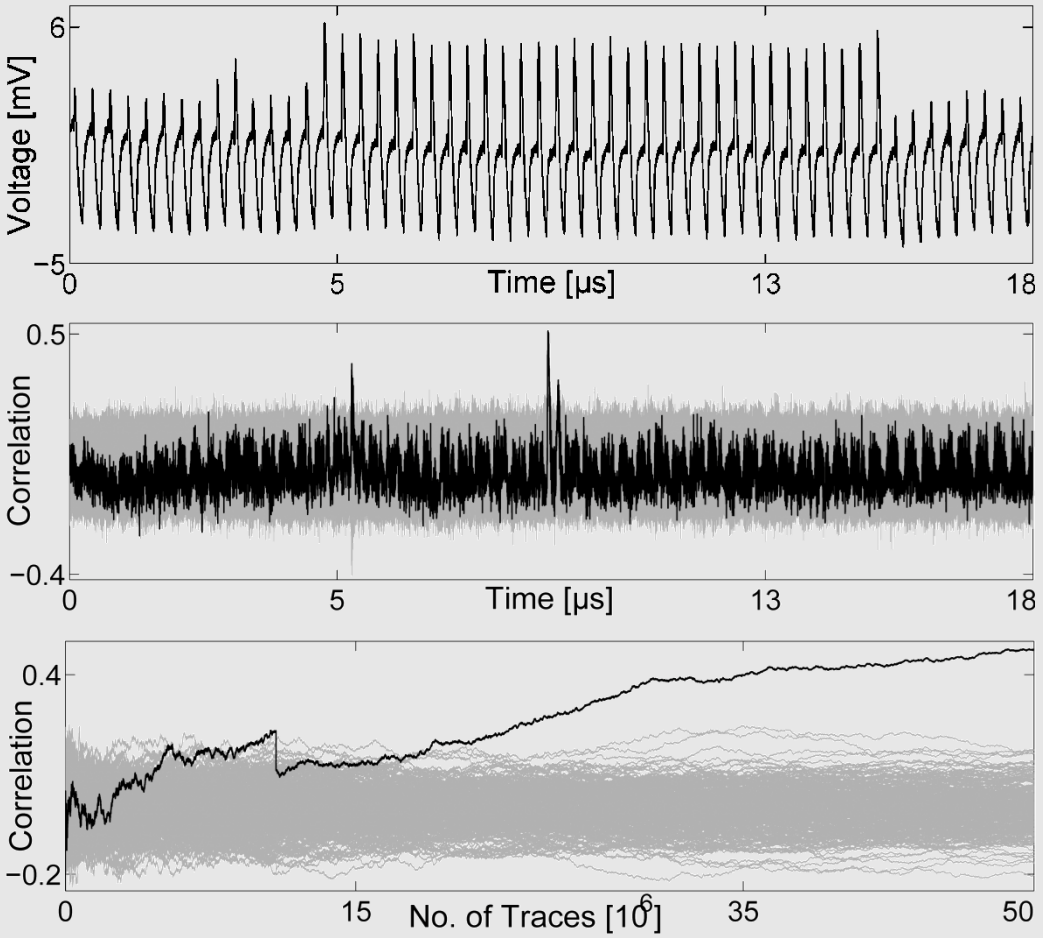
Profile 6: Pipelining and Special Enable

- Full SubBytes takes 31 clock cycles again
- First order attack no longer successful
- Even 50M measurements are not giving any results



Profile 6: 2nd Order Attack

- Full SubBytes also takes 31 clock cycles
- Depicted successful result after 50M measurements
- Leakage exploitable after 25M measurements



Summary

- What have we done:
  - Mapped highly optimized compact masked ASIC Sbox to efficiently use the available FPGA resources
  - Elimination of glitches by specially routed enable signal
  - High throughput by pipelining
  - Applicable to all modern Xilinx FPGAs (Virtex 5 onwards, 6 Input LUTs)
  - No first-order leakage after 50 million measurements
  - Second order leakage exploitable using 25 million traces

    -> similar 2nd order leakage as in reported TI implementations but much smaller
  - Source code available for evaluation: http://www.emsec.rub.de/research/publications

RUHR-UNIVERSITÄT BOCHUM

RUB

Thanks!
Any questions?

Amir Moradi, Oliver Mischke
Horst Görtz Institute for IT-Security

Cryptarchi 2012, Chateau de Goutelas, France, 20/06/2012