RUHR-UNIVERSITÄT BOCHUM

hgi EMSEC

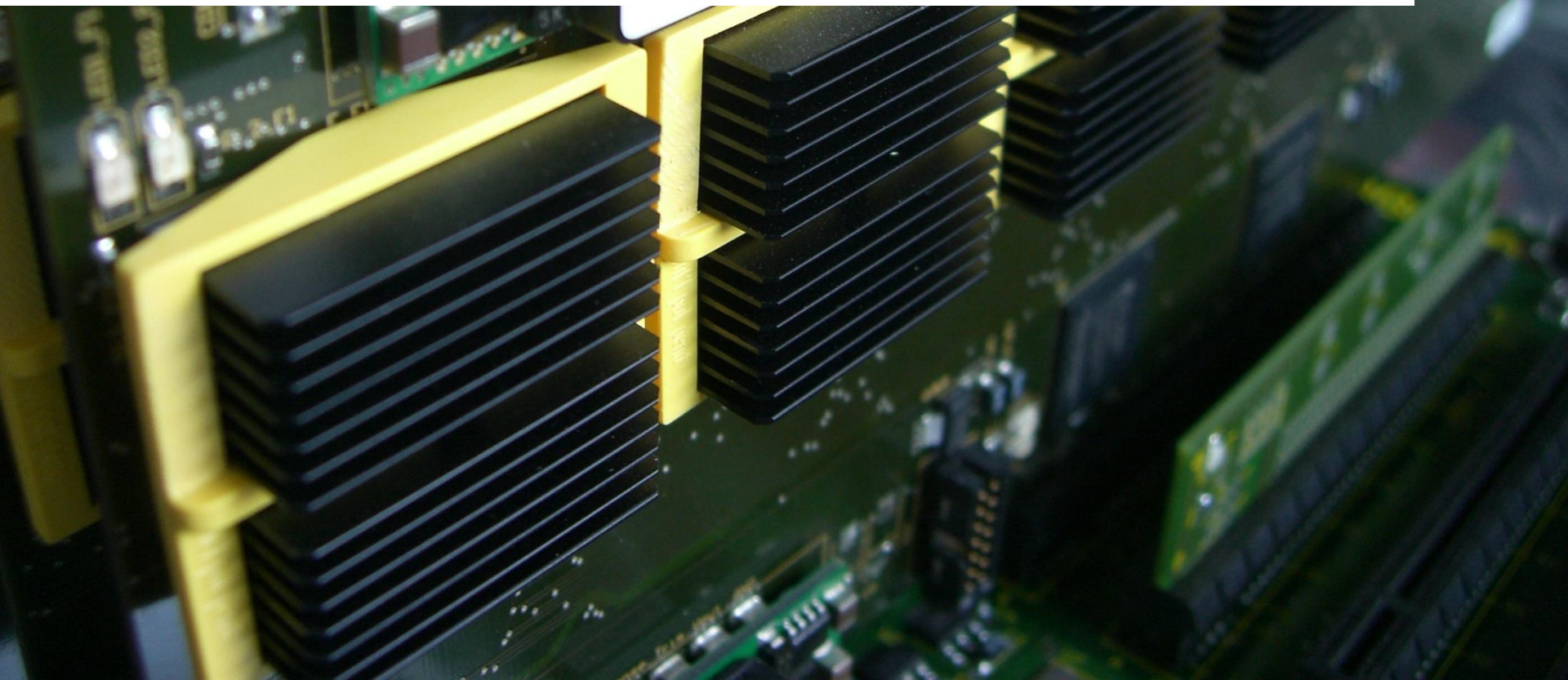
RUB

HIDING HIGHER-ORDER SIDE-CHANNEL LEAKAGE

− RANDOMIZING THRESHOLD IMPLEMENTATIONS IN RECONFIGURABLE HARDWARE −

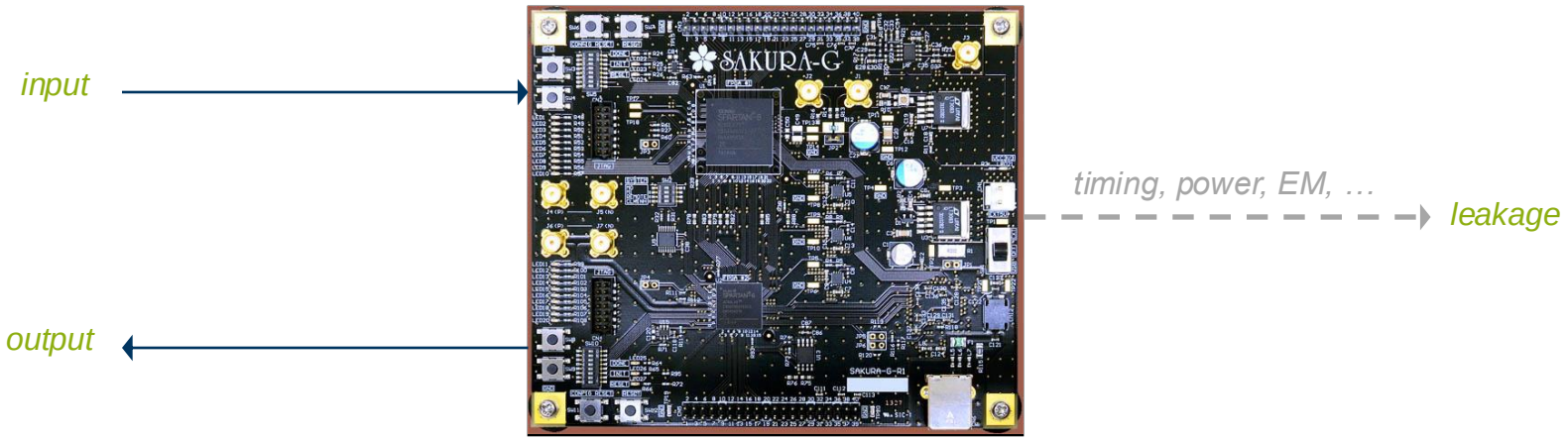
PASCAL SASDRICH, AMIR MORADI, TIM GÜNEYSU

FINAL CONFERENCE ON TRUDEVICE 2016, BARCELONA, SPAIN NOVEMBER 15, 2016



INTRODUCTION | SIDE-CHANNEL ANALYSIS (SCA)

ATTACKER MODEL



COUNTERMEASURES

masking

hiding

re-keying

INTRODUCTION | THRESHOLD IMPLEMENTATION

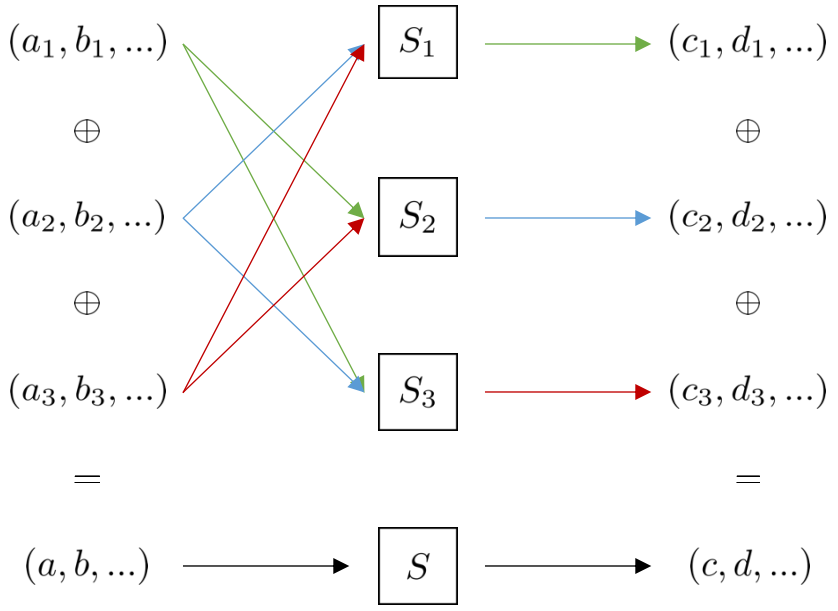
THRESHOLD IMPLEMENTATION:

- efficient *countermeasure in hardware* against (*first-order*) Side-Channel Analysis
- introduced in 2006 by Nikova et al. [1]
- provides provable security even in a glitch circuit

CONCEPT AND PROPERTIES:

- uniform masking
- non-completeness
- correctness
- uniform sharing of function outputs
  (each set of output pairs occurs with same probability)

NOTE: *The number of input and output shares depends on the function S.*



[1] S. Nikova, C. Rechberger, V. Rijmen, "Threshold Implementations Against Side-Channel Attacks and Glitches", ICICS, 2006

INTRODUCTION | MOTIVATION

BASICS:

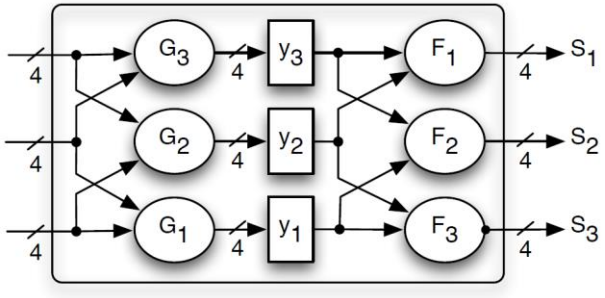
- **Side-Channel Analysis (SCA):** attacks exploit information leakage of cryptographic devices
- **Threshold Implementation (TI):** countermeasure based on Boolean masking and multi-party computation

PROBLEM:

TI only counteracts *first-order attacks*, but is vulnerable to *higher-order attacks* (using higher-order statistical moments).

DIFFERENT APPROACHES TO ENCOUNTER THIS PROBLEM:

- **Higher-order Threshold Implementations (HO-TI) [2]**
  - *might be* restricted to univariate settings
  - area overhead *might be* problematic
  - finding uniform representations *might be* challenging
- **Stay with 1st-order secure TI and make *higher-order attacks* harder**
  - reduce the signal (e.g., power equalization schemes, logic styles) [3]
  - increase the noise (e.g., shuffling) [4]



OUR CONTRIBUTION: General methodology (*dynamic hardware modifications*) to increase noise.

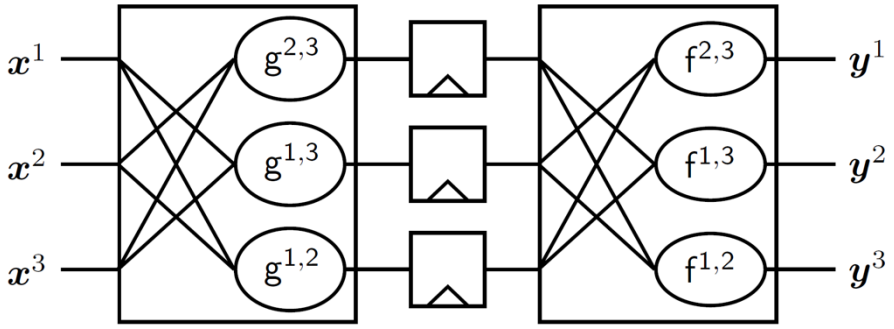
[2] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Higher-Order Threshold Implementations". ASIACRYPT 2014

[3] A. Moradi, A. Wild, "Assessment of Hiding the Higher-Order Leakages in Hardware – What are the Achievements versus Overheads?". CHES 2015

[4] P. Sasdrich, A. Moradi, T. Güneysu, "Affine Equivalence and its Application to Tightening Threshold Implementations". SAC 2015

CONCEPT | DYNAMIC HARDWARE MODIFICATION

- **THRESHOLD IMPLEMENTATION OF PRESENT:**
  - S-box decomposition into two quadratic functions *g* and *f* [5]
  - minimal number of shares (m = n = 3)
  - register stages to separate functions
  - linear permutation applied individually



- **RANDOM ENCODING:**
  - TI as network of look-up tables
  - each table updates 4 bit of internal state
  - use *White-Box Cryptography* [6] concepts:
    - apply random non-linear 4-bit encoding to every table output
    - apply inverse encoding to every adjacent table input (preserves correctness)

$$\mathsf{E}'_K = \underbrace{(\mathsf{f}^{r+1})^{-1} \circ \mathsf{E}^r_{k_r} \circ \mathsf{f}^r}_{table(s)} \circ \cdots \circ \underbrace{(\mathsf{f}^3)^{-1} \circ \mathsf{E}^2_{k_2} \circ \mathsf{f}^2}_{table(s)} \circ \underbrace{(\mathsf{f}^2)^{-1} \circ \mathsf{E}^1_{k_1} \circ \mathsf{f}^1}_{table(s)}$$

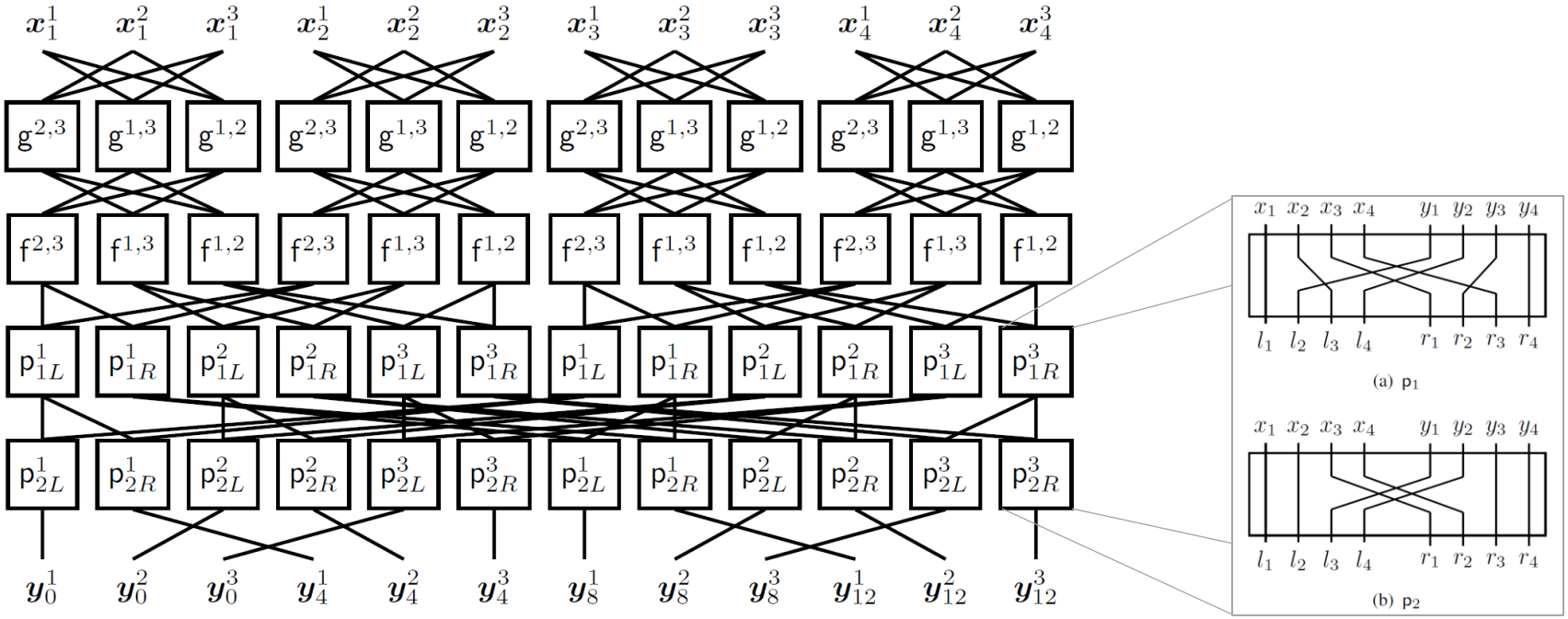
$$= (\mathsf{f}^{r+1})^{-1} \circ \mathsf{E}^r_{k_r} \circ \cdots \circ \mathsf{E}^2_{k_2} \circ \mathsf{E}^1_{k_1} \circ \mathsf{f}^1 = (\mathsf{f}^{r+1})^{-1} \circ \mathsf{E}_K \circ \mathsf{f}^1,$$

- **DYNAMIC UPDATE:**
  - find new random non-linear encodings using element swapping algorithm
  - update look-up tables using BRAM scrambling

[5] A. Poschmann, A. Moradi, K. Khoo, C. Lim, H. Wang, S. Ling, "Side-Channel Resistant Crypto for Less than 2300 GE". Journal of Cryptology, 2011

[6] S. Chow, P. A. Eisen, H. Johnson, P. C. van Oorschot, "White-Box Cryptography and an AES Implementation". SAC, 2002

CASE STUDY | PRESENT (QUARTER ROUND)



PRACTICAL FPGA IMPLEMENTATION:

- round-based architecture using look-up tables for TI S-box and permutation layer
- 4 quarter rounds in parallel, each using 48 BRAMs (as depicted)
- each BRAM can hold up to 32 different tables
  - store look-up tables for every round (31 rounds)
  - update tables using BRAM scrambling and remaining (empty) table entry
  - track context of active table positions

CASE STUDY | IMPLEMENTATION RESULTS

module/ component	resource utilization				
	logic	*memory*			*area*
	(LUT)	*(FF)*	*(DistRAM)*	*(BRAM)*	*(Slices)*
control logic	11	24	0	0	13
round function	96	0	0	192	87
g-layer	0	0	0	48	0
f-layer	0	0	0	48	0
p_1-layer	0	0	0	48	0
p_2-layer	0	0	0	48	0
reconfiguration	5081	3222	1952	0	2373
context engine	54	44	32	0	18
encoding engine	4800	2880	1920	0	2258
randomness generator	136	256	0	0	40
Total	5188	3246	1952	192	2473

PRACTICAL IMPLEMENTATION:

- post-place-and-route implementation on a Kintex-7 of SAKURA-X board
- basic architecture mainly implemented in Block RAM
- general purpose logic only required in order to perform dynamic hardware modification

PRACTICAL EVALUATION | NON-SPECIFIC T-TEST

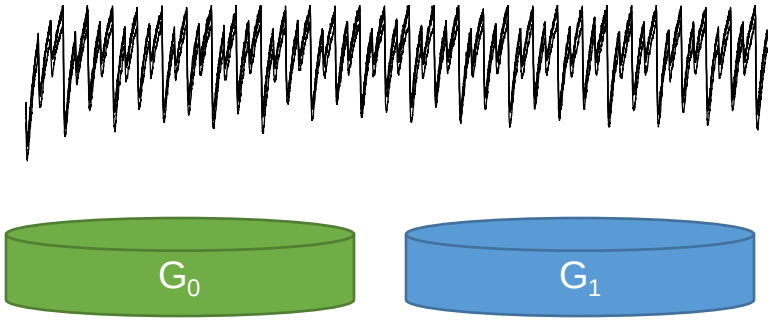
EVALUATION BASED ON WELCH'S t-TEST

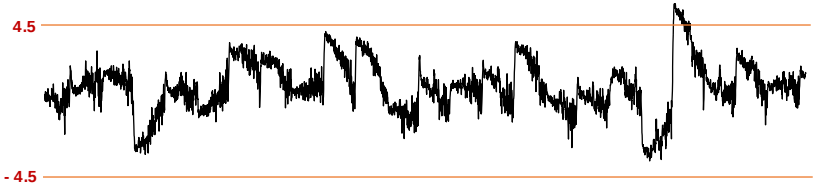
- measure (many) power traces with digital oscilloscope
- group traces depending on fix or randomly chosen plaintext (non-specific t-test)
- compute *sample mean* for each point in time
- compute *sample variance* for each point in time
- determine t-statistic for each point in time, according to:

$$t = \frac{\mu(T \epsilon G_1) - \mu(T \epsilon G_0)}{\sqrt{\frac{\delta^2(T \epsilon G_1)}{|G_1|} + \frac{\delta^2(T \epsilon G_0)}{|G_0|}}}$$

where μ denotes the *sample mean* and δ denotes the *sample variance*.

VISUALIZATION





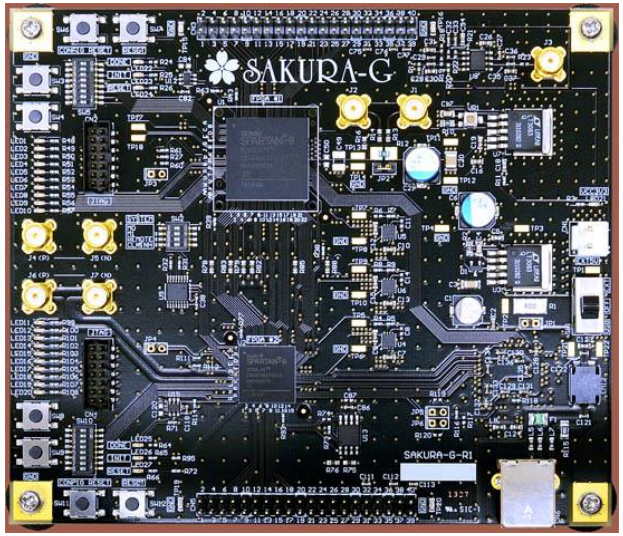
Fail/Pass Criteria: If there is any point in time for which the t-statistic exceeds a threshold of $\pm$ 4.5 the device under test fails.

More info: "Leakage Assessment Methodology - a clear roadmap for side-channel evaluations", CHES 2015, ePrint: 2015/207

PRACTICAL EVALUATION | SETUP

MEASUREMENT SETUP

- SAKURA-X Side-Channel Evaluation Board
- designs running @ 24 MHz
- power measurements using digital oscilloscope @ 500 MS/s



EVALUATION SETUP

- non-specific t-test (1st, 2nd, 3rd order)
- several million traces
- two different measurement profiles

PROFILE 1:

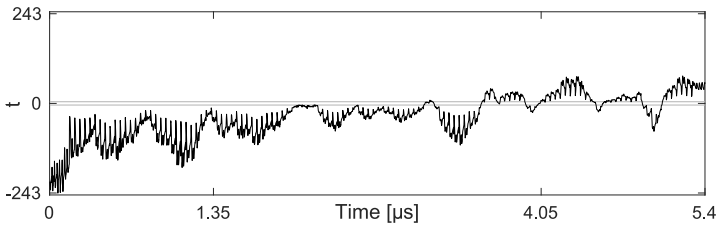
- reference measurement
- PRNG off
- 1 000 000 power traces

PROFILE 2:

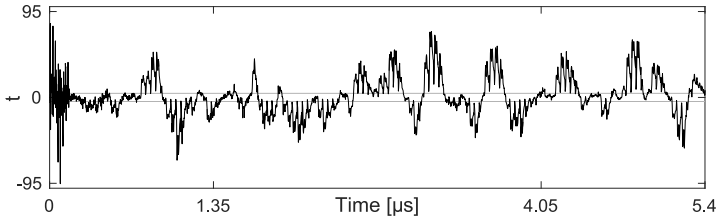
- actual measurement
- PRNG on
- 100 000 000 power traces

PRACTICAL EVALUATION | PROFILE 1 (PRNG OFF)

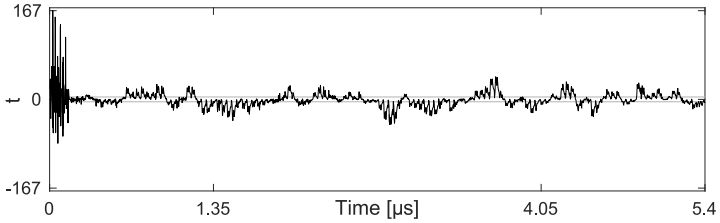
NON-SPECIFIC T-TEST (1 MILLION TRACES)



first-order

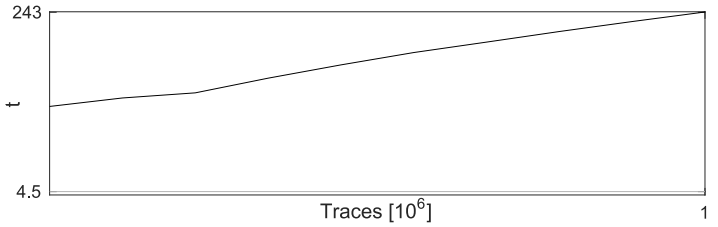


second-order

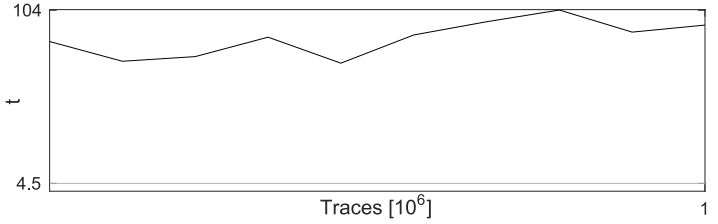


third-order

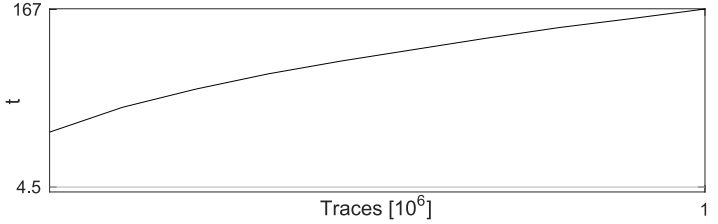
EVOLUTION OF ABSOLUTE T-TEST MAXIMUM



first-order



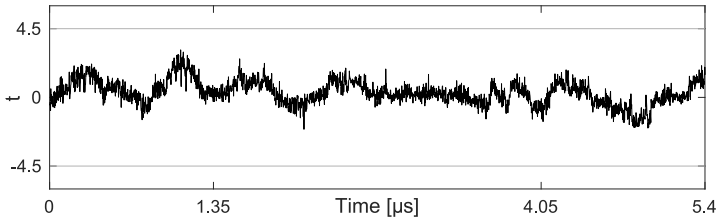
second-order



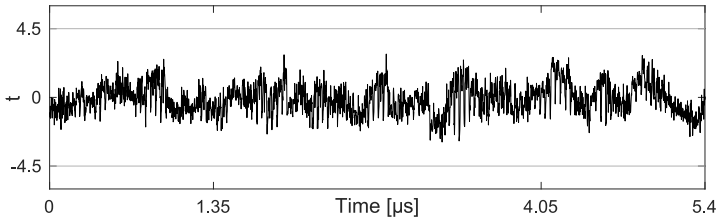
third-order

PRACTICAL EVALUATION | PROFILE 2 (PRNG ON)

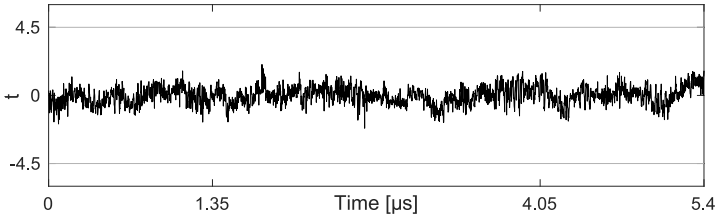
NON-SPECIFIC T-TEST (100 MILLION TRACES)



first-order

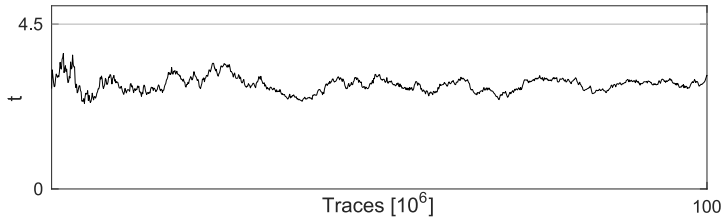


second-order

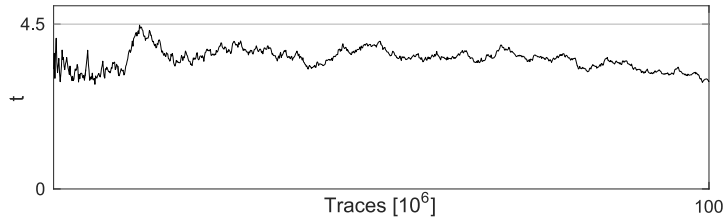


third-order

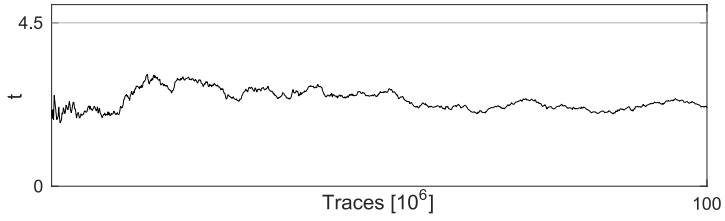
EVOLUTION OF ABSOLUTE T-TEST MAXIMUM



first-order



second-order



third-order

CONCLUSION

CONCEPT:

- success of higher-order attacks depends on noise-level
- combining *hiding countermeasures* (noise addition) with provable secure first-order TI
- *dynamic hardware modification* (inspired by white-box cryptography) as generic hiding approach

RESULTS:

- FPGA implementation combining dynamic hardware modification approach with PRESENT TI
- power measurements and leakage assessment (non-specific t-test)
- case study implementation is (practically) secure against higher-order attacks (2nd and 3rd order)

***Dynamic hardware modifications* form an alternative to Higher-Order Threshold Implementations providing *generality* and *scalability*.**

RUHR-UNIVERSITÄT BOCHUM

hgi EMSEC

RUB

HIDING HIGHER-ORDER SIDE-CHANNEL LEAKAGE

− RANDOMIZING THRESHOLD IMPLEMENTATIONS IN RECONFIGURABLE HARDWARE −

pascal.sasdrich@rub.de

FINAL CONFERENCE ON TRUDEVICE 2016, BARCELONA, SPAIN NOVEMBER 15, 2016

Thank you for your attention!
Any questions?