RUHR-UNIVERSITÄT BOCHUM

hgi EMSEC

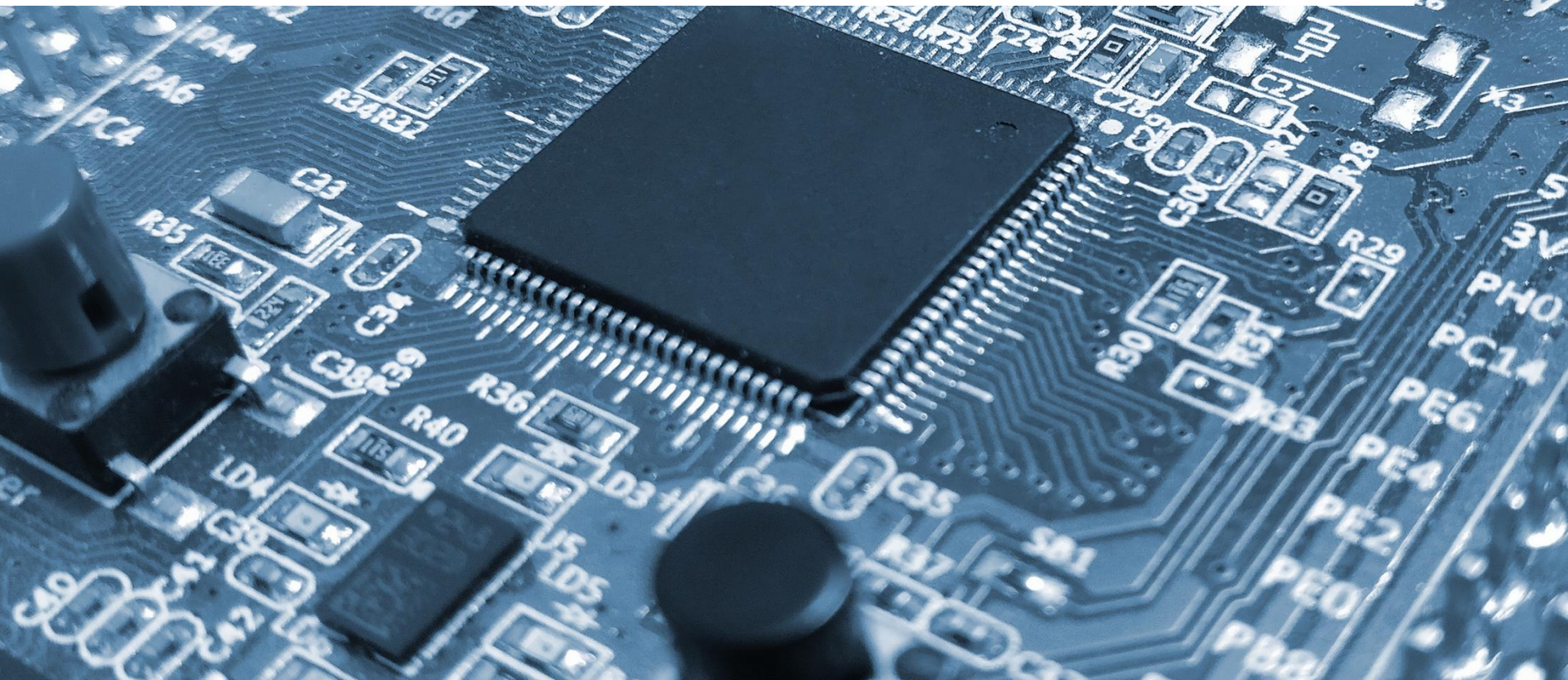
RUB

HIDING HIGHER-ORDER SIDE-CHANNEL LEAKAGE

− RANDOMIZING CRYPTOGRAPHIC IMPLEMENTATIONS IN RECONFIGURABLE HARDWARE −

PASCAL SASDRICH, AMIR MORADI, TIM GÜNEYSU

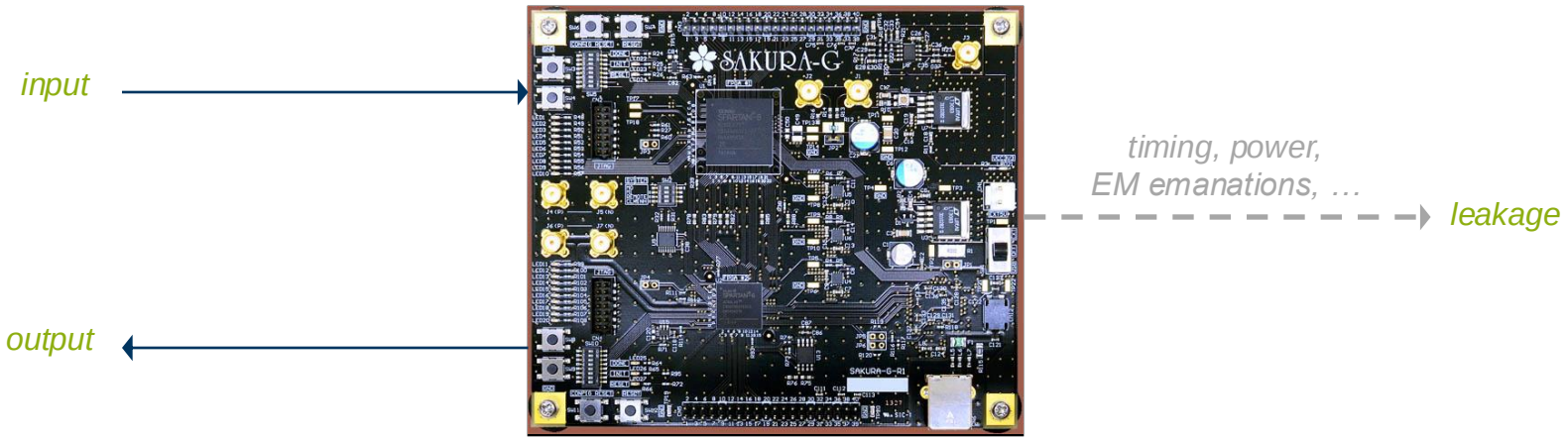
RSA CONFERENCE CRYPTOGRAPHERS' TRACK, SAN FRANCISCO, USA FEBRUARY 15, 2017



INTRODUCTION

INTRODUCTION | SIDE-CHANNEL ANALYSIS (SCA)

ATTACKER MODEL



COUNTERMEASURES

masking

hiding

re-keying

INTRODUCTION | MOTIVATION

BASICS:

- **Side-Channel Analysis (SCA):** attacks exploit information leakage of cryptographic devices
- **Side-Channel Protection:** countermeasures based on masking, hiding or re-keying (using random behavior)

PROBLEM:

Common countermeasures only protect against *first-order attacks*, but still are vulnerable to *higher-order attacks* (using higher-order statistical moments).

DIFFERENT APPROACHES TO ENCOUNTER THIS PROBLEM:

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

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

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

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

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

CONCEPT

CONCEPT | DYNAMIC HARDWARE MODIFICATION

- **OBSERVATIONS:**

  1. *Cryptographic implementations can be represented as sequence of atomic functions applied sequentially.*
  2. *Cryptographic implementations can be modeled by different but equivalent directed graphs.*

- **APPROACH:**
  - build a side-channel protected implementation using classical countermeasures (*masking*)
  - find directed graph representing the side-channel protected implementation
  - morph graph into different but equivalent representation using random encodings
  - update (randomize) protected implementation according to new representation

Algorithm 1: Morphing algorithm for cryptographic implementations

Input : $\mathcal{G} = (\mathcal{V}, \mathcal{A})$: digraph representing a cryptographic implementation.
Output: $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{A}^*)$: digraph representing an encoded cryptographic implementation.

$\mathcal{G}^* = (\mathcal{V}^*, \mathcal{A}^*)$: $\mathcal{V}^* \leftarrow \mathcal{V}$, $\mathcal{A}^* \leftarrow \mathcal{A}$
for $\forall v_i \in \mathcal{V}^*$ **do**
  $\mathcal{D} \leftarrow \emptyset$
  $s \leftarrow \mathsf{f}(v_i)$, $\mathcal{V}^* \leftarrow \mathcal{V}^* \setminus \{v_i\}$
  for $\forall v_j \in \mathcal{V}^*$ **do**
    if $a_{ij} \in \mathcal{A}^*$ **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup \mathsf{f}^{-1}(v_j)$
      $\mathcal{V}^* \leftarrow \mathcal{V}^* \setminus \{v_j\}$, $\mathcal{A}^* \leftarrow \mathcal{A}^* \setminus \{a_{ij}\}$,
    end
  end
  for $\forall d_i \in \mathcal{D}$ **do**
    $\mathcal{V}^* \leftarrow \mathcal{V}^* \cup \{s, d_i\}$, $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \langle s, d_i \rangle$,
  end
end
return $\mathcal{G}^*$

CASE STUDY

CASE STUDY | THRESHOLD IMPLEMENTATION

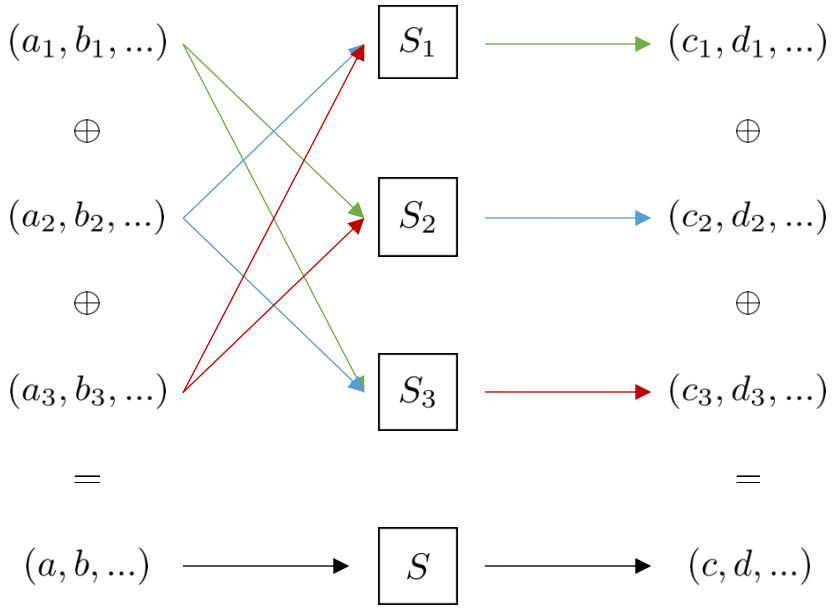
THRESHOLD IMPLEMENTATION:

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

CONCEPT AND PROPERTIES:

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

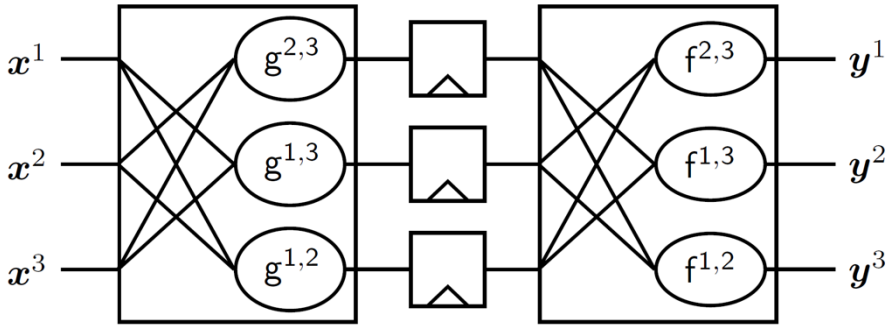
NOTE: *The number of input and output shares depends on the function S.*



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

CASE STUDY | CONCEPT

- **THRESHOLD IMPLEMENTATION OF PRESENT CIPHER:**
  - 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)

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

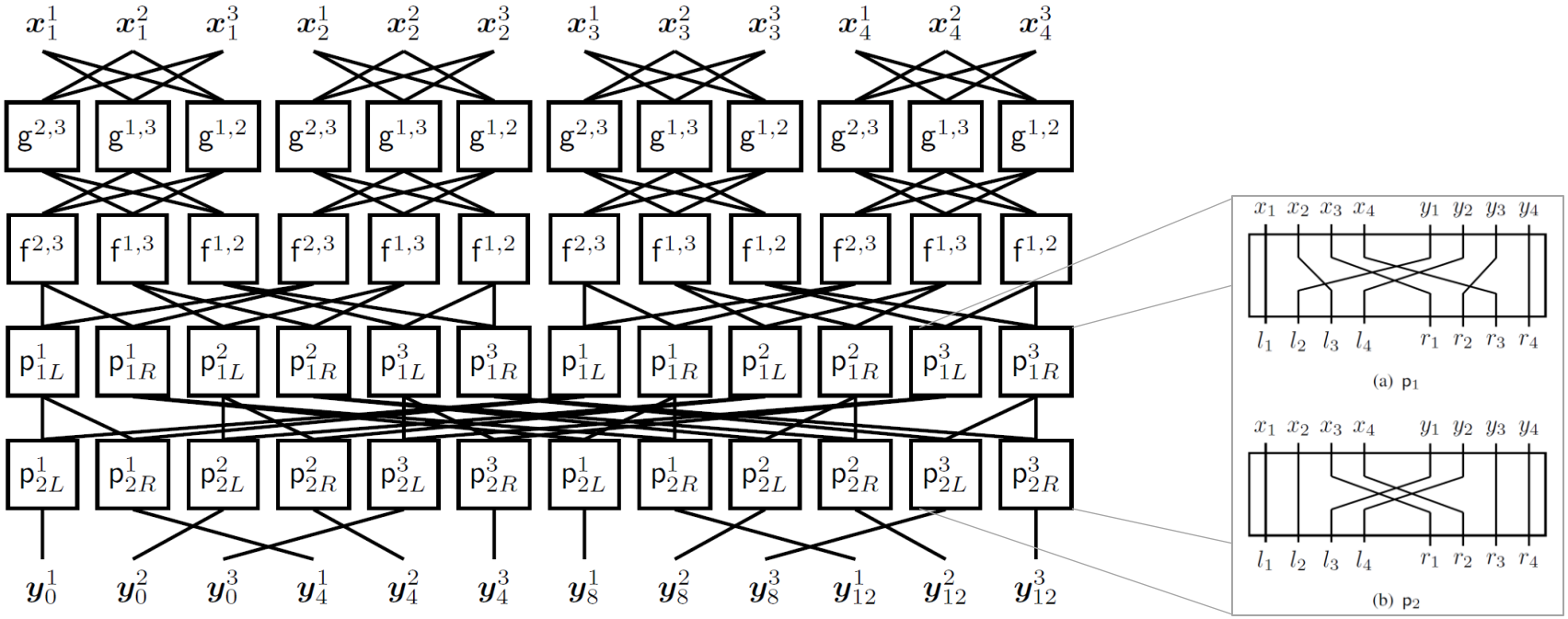
$$= (\mathsf{f}^{r+1})^{-1} \circ \mathsf{E}^r_{k_r} \circ \cdots \circ \mathsf{E}^2_{k_2} \circ \mathsf{E}^1_{k_1} \circ \mathsf{f}^1 = (\mathsf{f}^{r+1})^{-1} \circ \mathsf{E}_K \circ \mathsf{f}^1,$$

- **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 | IMPLEMENTATION (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
- 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)*	*(LUTRAM)*	*(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	3129	3222	1952	0	2373
Context Engine	22	44	32	0	18
Encoding Engine	2880	2880	1920	0	2258
Randomness Generator	136	256	0	0	40
Total	3236	3246	1952	192	2473

PRACTICAL IMPLEMENTATION:

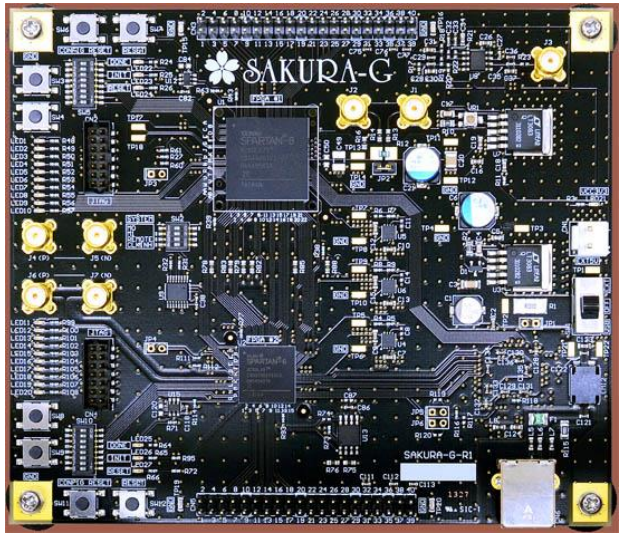
- post-place-and-route implementation on a Kintex-7 of SAKURA-X board
- basic architecture mainly implemented in Block RAM
- general purpose logic only required in order to perform dynamic hardware modification

SIDE-CHANNEL EVALUATION

SIDE-CHANNEL EVALUATION | SETUP

MEASUREMENT SETUP

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



EVALUATION SETUP

- high-performance measurement and evaluation setup
- two different measurement profiles
- leakage assessment methodology:
  non-specific t-test (for 1st, 2nd, 3rd order)

PROFILE 1:

- reference measurement
- PRNG off (countermeasure disabled)
- 1 000 000 power traces
- random vs. fix plaintexts

PROFILE 2:

- actual measurement
- PRNG on (countermeasure enabled)
- 100 000 000 power traces
- random vs. fix plaintexts

SIDE-CHANNEL EVALUATION | NON-SPECIFIC T-TEST

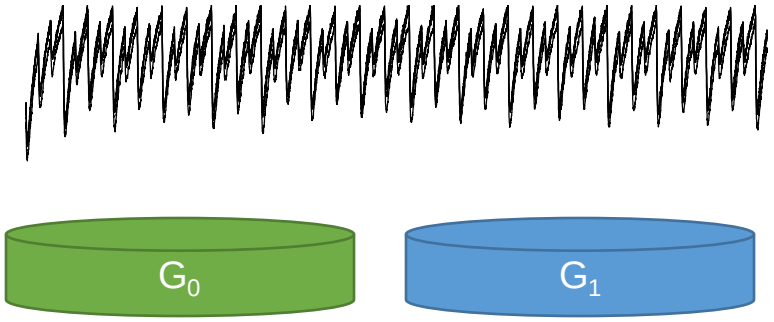
EVALUATION BASED ON WELCH'S t-TEST

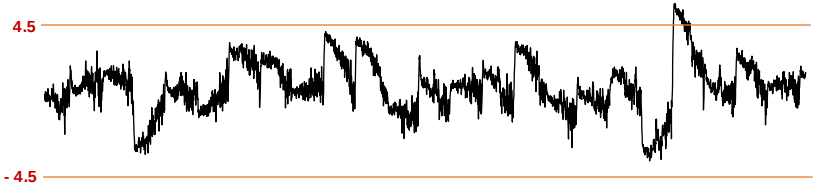
- measure (many) power traces with digital oscilloscope
- group traces depending on fix or randomly chosen plaintext (non-specific t-test)
- compute *sample mean* for each point in time
- compute *sample variance* for each point in time
- determine t-statistic for each point in time, according to:

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

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

VISUALIZATION



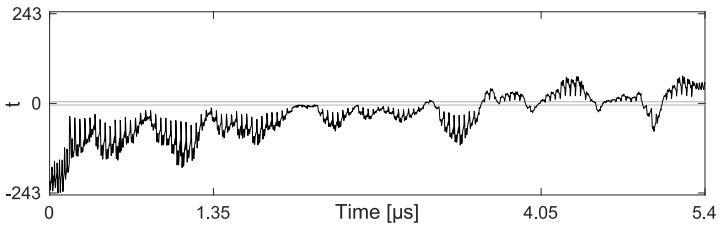


Fail/Pass Criteria: If there is any point in time for which the t-statistic exceeds a threshold of $\pm$ 4.5 the device under test fails.

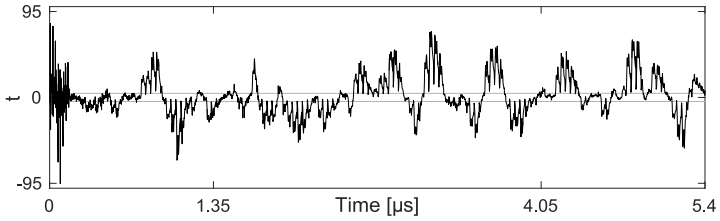
More info: "Leakage Assessment Methodology - a clear roadmap for side-channel evaluations", CHES 2015, ePrint: 2015/207

SIDE-CHANNEL EVALUATION | PROFILE 1 (PRNG OFF)

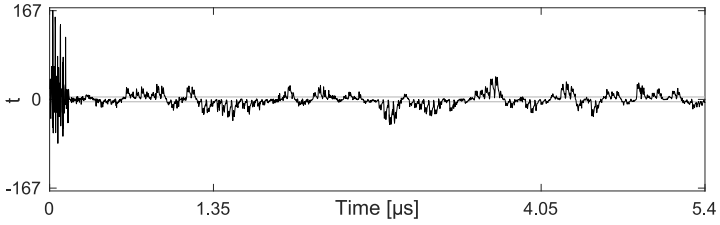
NON-SPECIFIC T-TEST (1 MILLION TRACES)



first-order

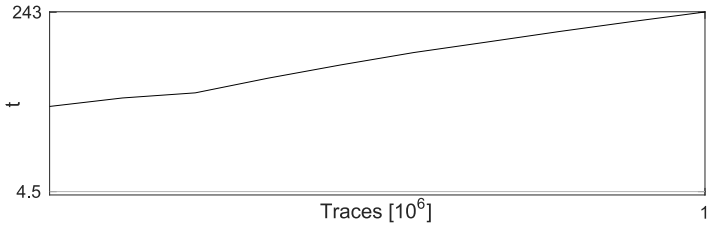


second-order

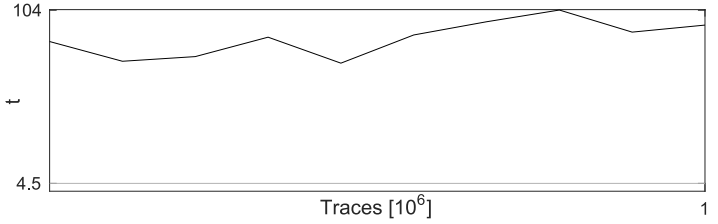


third-order

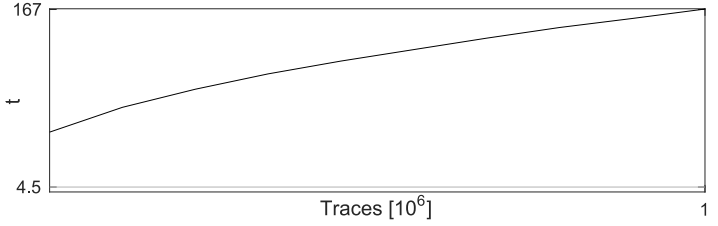
EVOLUTION OF ABSOLUTE T-TEST MAXIMUM



first-order



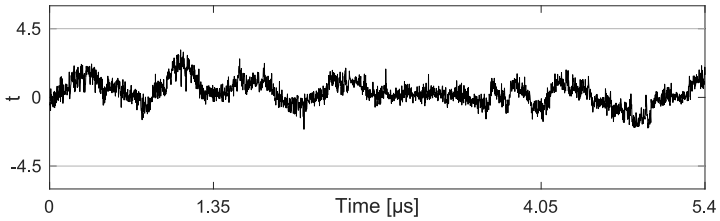
second-order



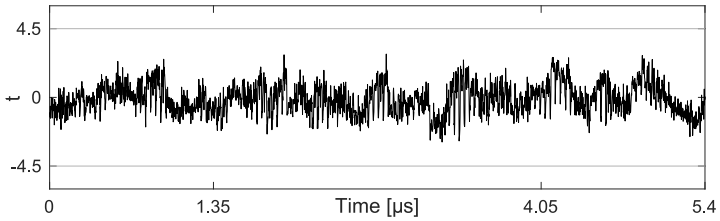
third-order

SIDE-CHANNEL EVALUATION | PROFILE 2 (PRNG ON)

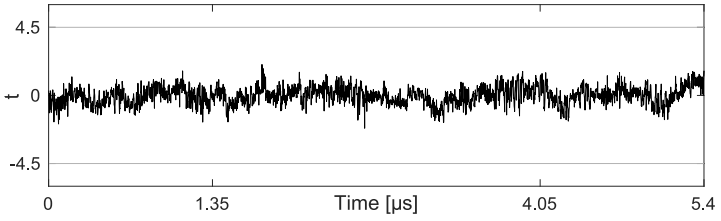
NON-SPECIFIC T-TEST (100 MILLION TRACES)



first-order

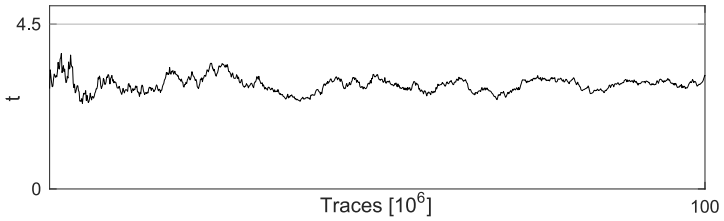


second-order

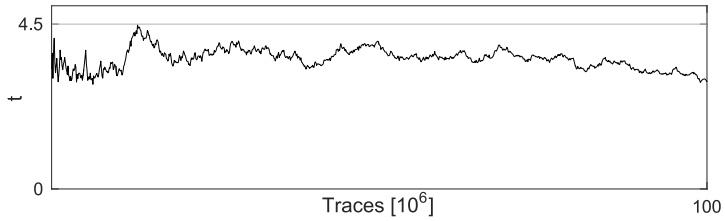


third-order

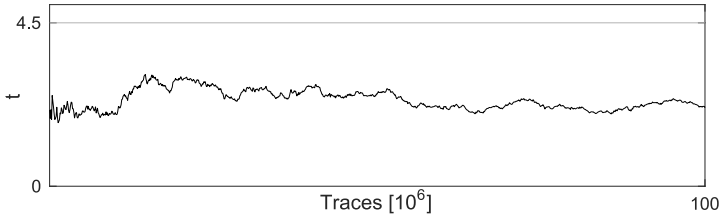
EVOLUTION OF ABSOLUTE T-TEST MAXIMUM



first-order



second-order



third-order

CONCLUSION

CONCLUSION

CONCEPT:

- success of higher-order attacks depends on noise-level
- combining *hiding countermeasures* (noise addition) with classical approaches (e.g. first-order secure TI)
- *dynamic hardware modifications* (inspired by white-box cryptography) as generic hiding approach

RESULTS:

- proposing a generic approach and methodology called *dynamic hardware modifications*
- case study: FPGA implementation combining *dynamic hardware modifications* with PRESENT TI
- providing power measurements and leakage assessment (using non-specific t-test)
- case study implementation is (practically) secure against higher-order attacks (2^{nd} and 3^{rd} order)

Dynamic hardware modifications are an alternative approach achieve higher-order protection 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

RSA CONFERENCE CRYPTOGRAPHERS' TRACK, SAN FRANCISCO, USA FEBRUARY 15, 2017

Thank you for your attention!
Any questions?