RUHR-UNIVERSITÄT BOCHUM

RUB

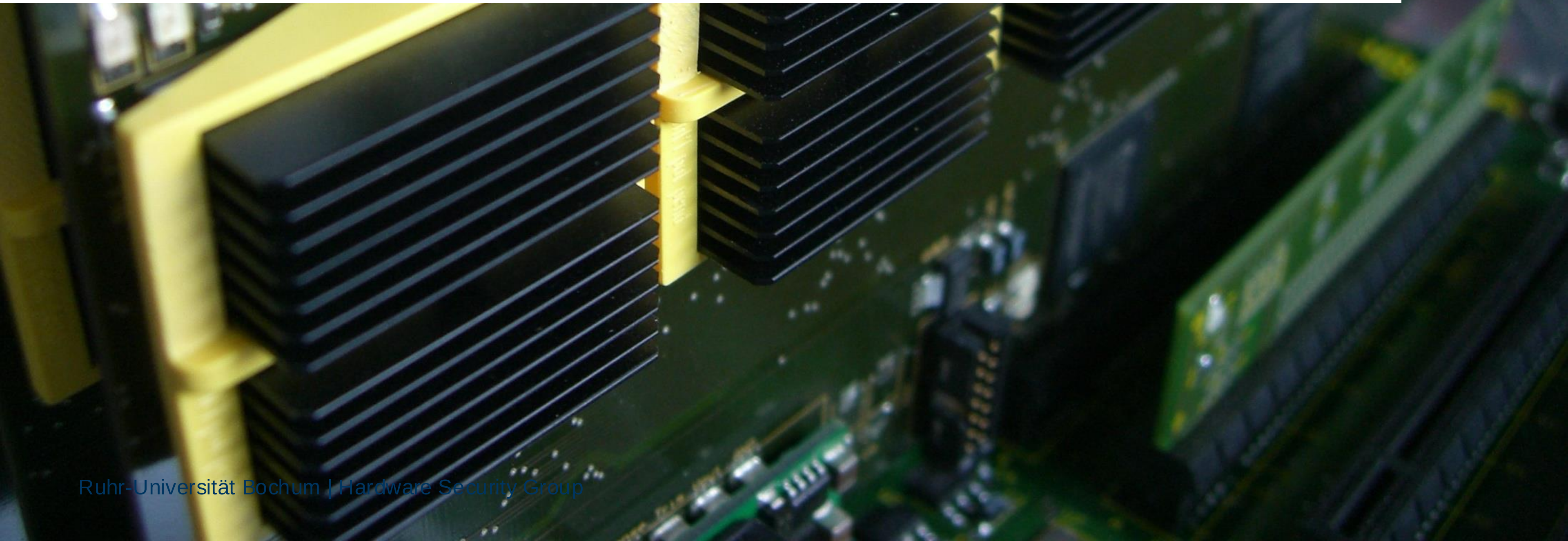
Side-Channel Protection by Randomizing Look-Up Tables on Reconfigurable Hardware

- Pitfalls of Memory Primitives -

COSADE 2015, Berlin, Germany

Pascal Sasdrich, Oliver Mischke,
Amir Moradi, and Tim Güneysu

26.03.2015



Outline

1. Motivation
2. Xilinx Memory Primitives
3. Block Memory Content Scrambling
4. Contribution
5. Randomized Look-Up Tables
6. Case Study and S-Box Designs
7. Evaluation and Results
8. Conclusion

Motivation

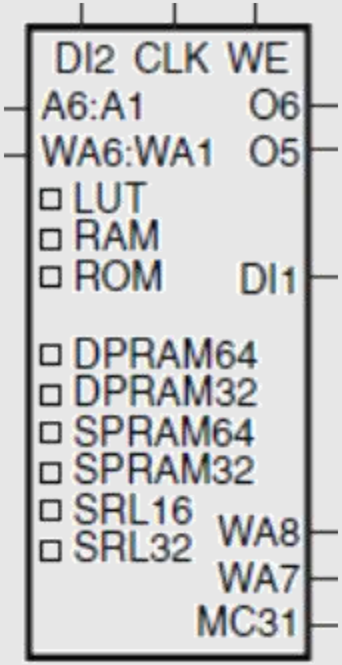
- At CHES 2011:

  Block Memory content Scrambling (BMS) was proposed as an effective way of 1st order side-channel protection.

- Our goals:
  - Analyze different ways for 1st order protection using randomized look-up tables.
  - Find optimal choices for FPGAs and 8-bit S-boxes.

Xilinx Memory Primitives

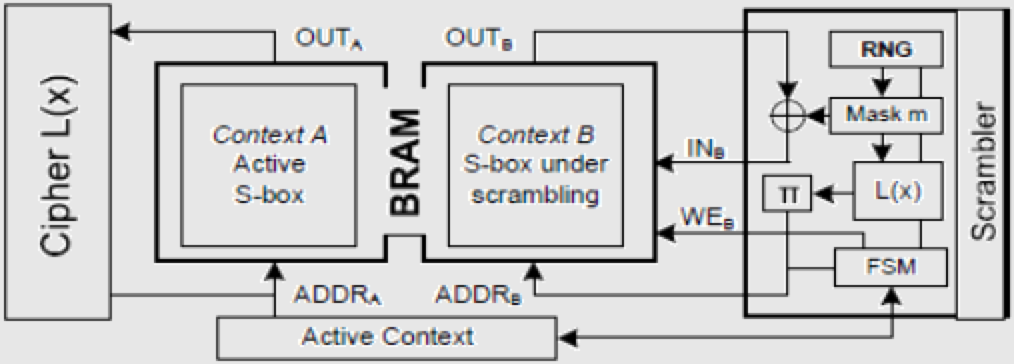
- Specific **slice logic** components can be implemented as distributed memory:
  - RAM32M (32x8bit SPRAM)
  - RAM64M (64x4bit SPRAM)
  - RAM256X1S (256x1bit SPRAM)
- Dedicated block memory primitives (RAMB8BWER) can be used as true dual-port block memory



Block Memory Content Scrambling

Main idea:

- Store 2 S-/T-Tables in one BRAM
- First table is active context and used for encryption.
- Second table is passive context and updated (scrambled) with fresh randomness.
- After update, contexts are switched.



T. Güneysu and A. Moradi. Generic Side-Channel Countermeasures for Reconfigurable Devices.

Disadvantages:

- Area overhead
- Latency
- Shared masks

Contribution

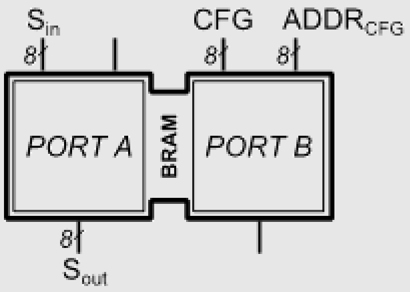
1. Analyzed **Xilinx FPGA memory** primitives to prevent 1st order side-channel leakage.

2. Built **randomized look-up tables** of different memory primitives.

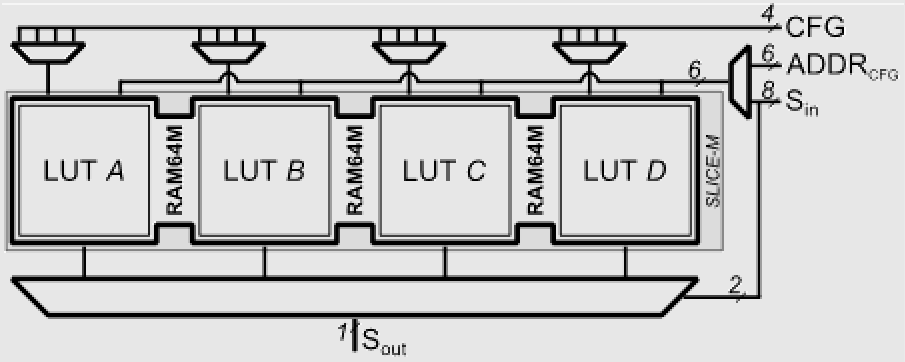
3. Evaluated designs using a state-of-the-art **leakage assessment methodology**.

4. We **revealed pitfalls** of using memory primitives for side-channel protection.

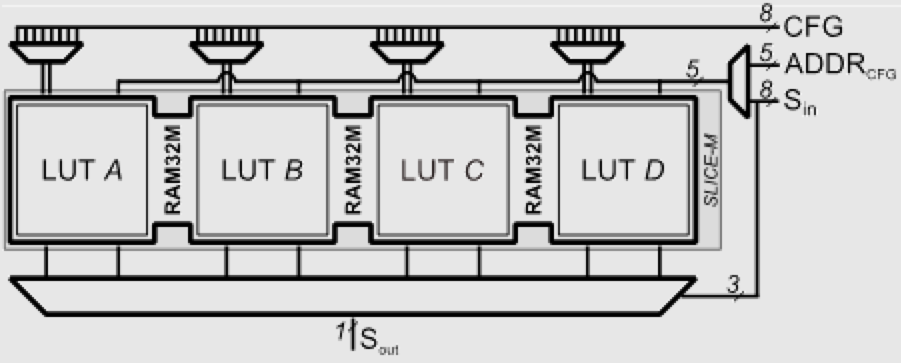
Randomized Look-Up Tables



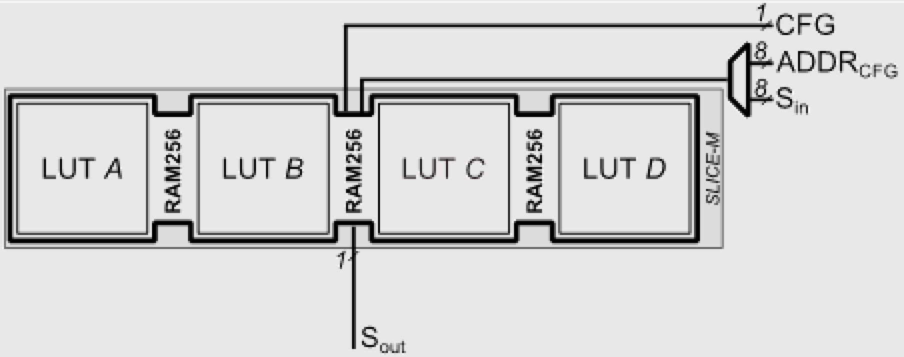
RAMB8BWER



RAM64M

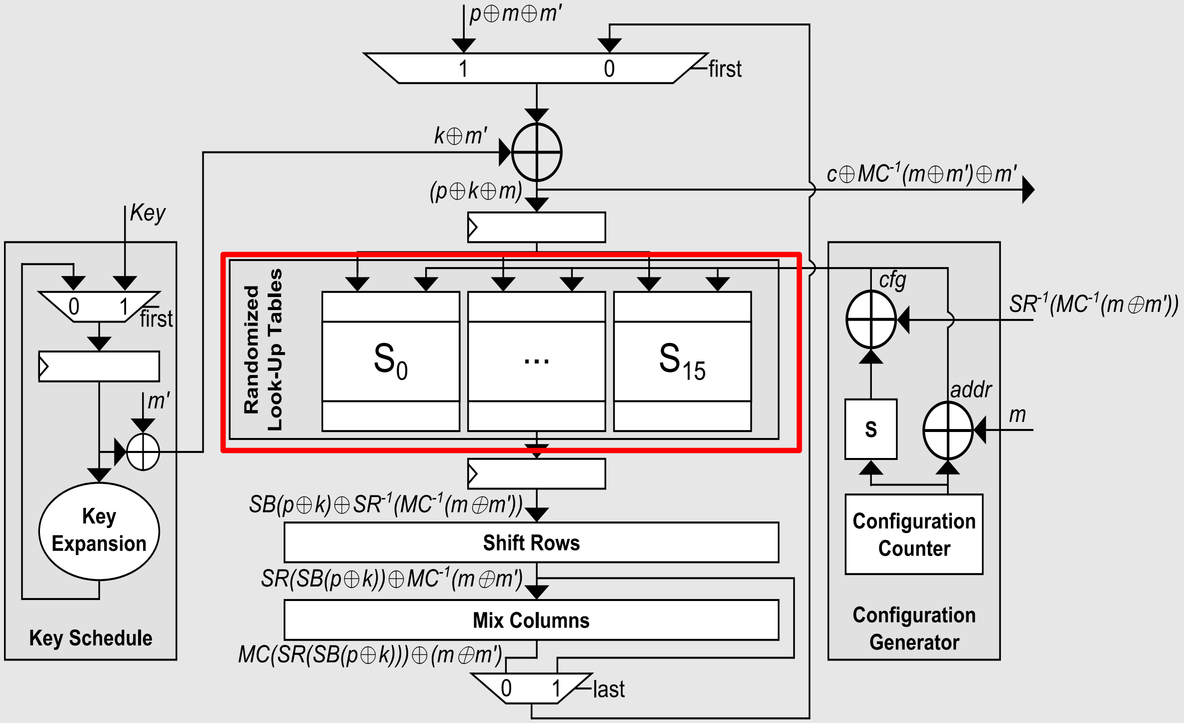


RAM32M



RAM256X1S

Case Study


$p \oplus m \oplus m'$
1
0
first
$k \oplus m'$
$(p \oplus k \oplus m)$
$c \oplus MC^{-1}(m \oplus m') \oplus m'$
Key
0
1
first
m'
Key Expansion
Key Schedule
Randomized Look-Up Tables
S_0
...
S_{15}
cfg
$SR^{-1}(MC^{-1}(m \oplus m'))$
addr
m
S
Configuration Counter
Configuration Generator
$SB(p \oplus k) \oplus SR^{-1}(MC^{-1}(m \oplus m'))$
Shift Rows
$SR(SB(p \oplus k)) \oplus MC^{-1}(m \oplus m')$
Mix Columns
$MC(SR(SB(p \oplus k))) \oplus (m \oplus m')$
0
1
last

- AES as case study
- Build randomized look-up tables using different memory primitives
- Replaced SubBytes with different designs.

S-Box Designs

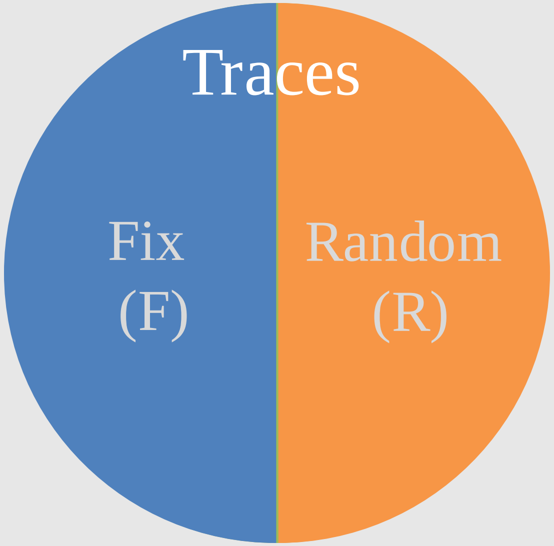
Memory		**SubBytes**		**Configuration**		**Max.**
Primitive	*Logic*	*Dist. Mem.*	*Block Mem.*	*Logic*	*Memory*	**Throughput**
	(LUT)	*(LUT)*	*(BRAM16)*	*(LUT)*	*(FF)*	*(Mbit/s)*
BMS [CHES11]	-	-	16	1706	1169	35.4
RAMB8BWER	-	-	8	298	8	68.6
RAM256X1S	128	512	-	298	8	77.0
RAM64M	768	512	-	727	6	247.3
RAM32M	1920	512	-	1222	5	363.3

Evaluation

Welch‘s T-Test

fix vs. random:

- 1 fix plaintext

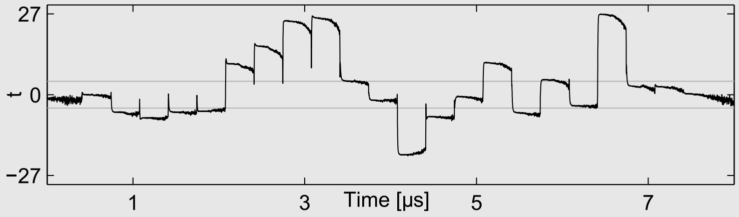


$\mathrm{N_F} = \text{Size of F}$
$\mathrm{X_F} = \text{Mean of F}$
$\mathrm{S_F} = \text{Std. deviation of F}$

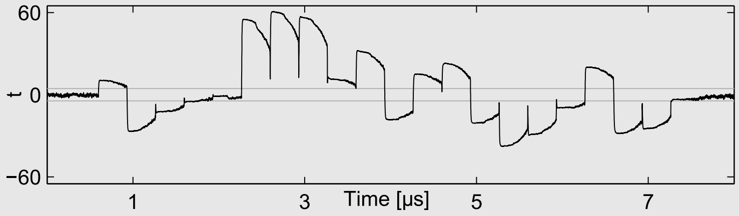
$\mathrm{N_R} = \text{Size of R}$
$\mathrm{X}_R = \text{Mean of R}$
$\mathrm{S}_R = \text{Std. deviation of R}$

$$\mathrm{T} = \frac{X_F - X_R}{\sqrt{\frac{S_F^2}{N_F} + \frac{S_R^2}{N_R}}}$$

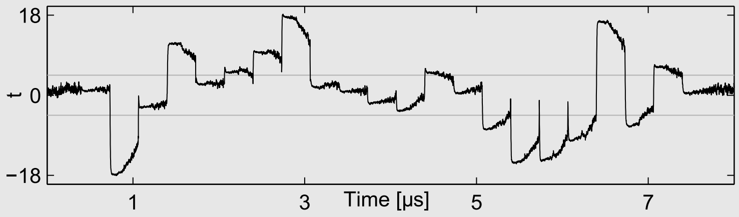
Results



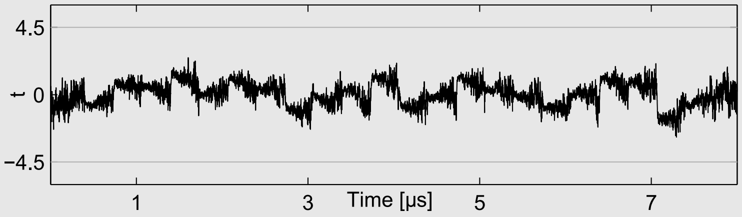
RAM32M (1 million traces)



RAM64M (1 million traces)



RAM256X1S (1 million traces)



RAMB8BWER (10 million traces)

- leakage is detectable for distributed memory primitives
- assume that leakage is due to internal slice architecture
- BRAM primitives exhibit no detectable leakage (even for larger trace numbers)

Conclusion

Our results infer the pitfall of using distributed memory primitives:

- Distributed memory causes a secure scheme to exhibit 1st order leakage.
- Replaced with Block Memory 1st order leakage is not detectable.

Besides, our designs achieve higher throughput and require less randomness than original BMS scheme.

RUHR-UNIVERSITÄT BOCHUM

RUB

Side-Channel Protection by Randomizing Look-Up Tables on Reconfigurable Hardware

- Pitfalls of Memory Primitives -

COSADE 2015, Berlin, Germany

Pascal Sasdrich, Oliver Mischke,
Amir Moradi, and Tim Güneysu

26.03.2015

Thank you for your attention!

Any Questions?