Evaluating the ROCKY Countermeasure for Side-Channel Leakage

K. Miteloudi[1] Ł. Chmielewski[1] L. Batina[1] N. Mentens[2,3]

[1]iCIS - Digital Security Group, Radboud University, The Netherlands

[2]imec-COSIC - ES&S, ESAT, KU Leuven, Belgium

[3]LIACS, Leiden University, The Netherlands

VLSI-SoC 2021 - 29th IFIP/IEEE International Conference on Very Large Scale Integration
October 4 - October 8, 2021

Outline

Physical attacks

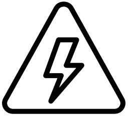
- exploit vulnerabilities in the implementation of cryptographic primitives
  - extract information on the secret key or
  - other internally processed data

Main classification of physicals attacks:

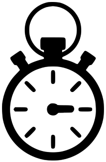
- Fault Injection (FI) attacks, where the attacker inserts faults (e.g. by glitching some parameters like voltage, power, clock etc.) in order to disrupt the normal behavior of the algorithm.
- Side-Channel Analysis (SCA) attacks, where the device under attack operates within specified conditions and the attacker observes the physical leakage.

Most common side channel resources:

Power



Timing



EM emission

Our Contribution

ROCKY countermeasure

- Recently introduced as a countermeasure against fault injection attacks.
- Efficient fault detection when combined with modular redundancy.
- It is based on the random rotation of the internal state of cryptographic primitives and can be applied to any symmetric cryptographic algorithm that is based on a shift invariant permutation.

In this work:

- Implementation of an unprotected architecture and three ROCKY-protected architectures of Xoodoo on an FPGA.
- Evaluation of the resistance against side-channel power analysis attacks of all architectures.

Algorithm 1: Definition of Xoodoo[n_r] with n_r the number of rounds

Parameters: Number of rounds n_r

for *Round index i from 1 - n_r to 0* **do**
  $A = R_i(A)$
end

Here R_i is specified by the following steps:

θ :

$$P \leftarrow A_0 \oplus A_1 \oplus A_2$$
$$E \leftarrow P \lll (1,5) \oplus P \lll (1,14)$$
$$A_y \leftarrow A_y \oplus E \quad for\ y \in \{0,1,2\}$$

ρ_{west} :

$$A_1 \leftarrow A_1 \lll (1,0)$$
$$A_2 \leftarrow A_2 \lll (0,11)$$

ι :

$$A_0 \leftarrow A_0 \oplus C_i$$

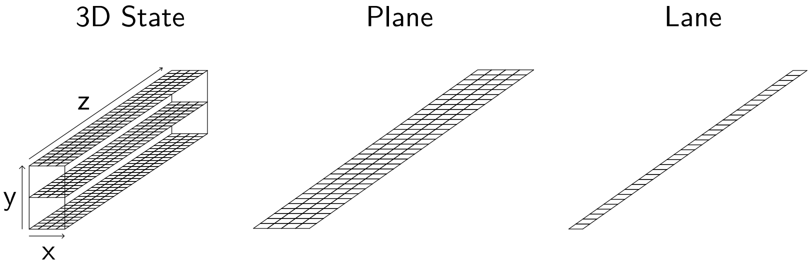
χ :

$$B_0 \leftarrow \overline{A_1} \bullet A_2$$
$$B_1 \leftarrow \overline{A_2} \bullet A_0$$
$$B_2 \leftarrow \overline{A_0} \bullet A_1$$
$$A_y \leftarrow A_y \oplus B_y \quad for\ y \in \{0,1,2\}$$

ρ_{east} :

$$A_1 \leftarrow A_1 \lll (0,1)$$
$$A_2 \leftarrow A_2 \lll (2,8)$$

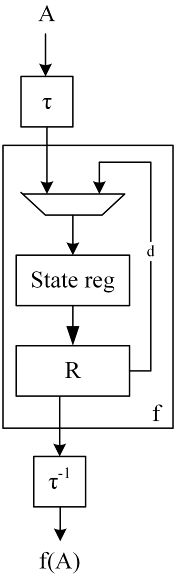
State representation of Xoodoo

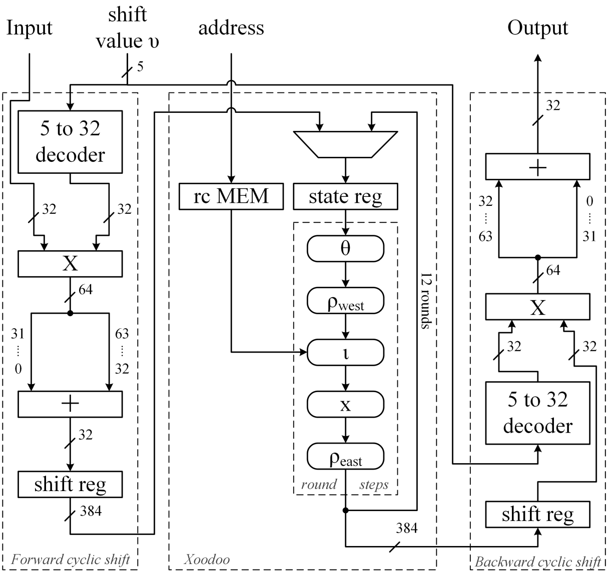


- $(x = 4\ ,\ y = 3\ ,\ z = 32) \Rightarrow$ 3 planes and 12 lanes of 32 bits
- Mapping from 3D State to 1D bit array : $i = z + 32(x + 4y)$.

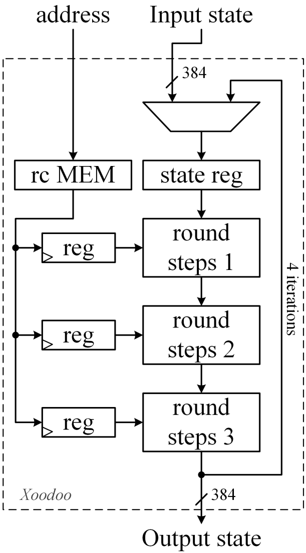
- A cryptographic permutation f can be applied with a shift-invariant round function to a shifted version of a state A.
- Let $f = \mathrm{R}^d$ with d the number of rounds and τ cyclist shift operations, then we have:

$$f(A) = \tau^{-1}(f(\tau(A))$$

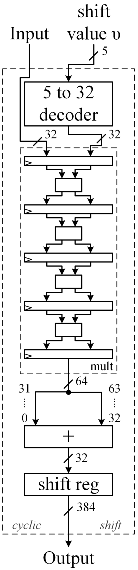




Basic Architecture 1



Architecture 2



Architecture 3

Oscilloscope

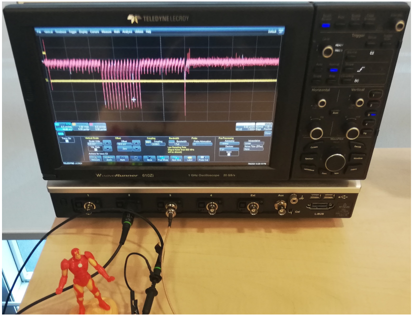
- Teledyne Lecroy Waverunner 8404M

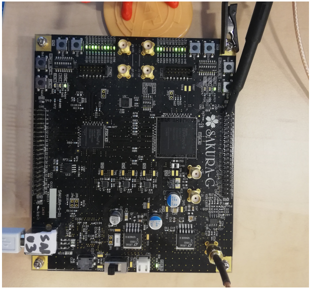
Sakura-G board

- Two Xilinx Spartan-6 FPGAs (xc6slx9 and xc6slx75)

PC

- Intel i7 3.4GHz processor and 64GB RAM





PC - Oscilloscope (Ethernet)

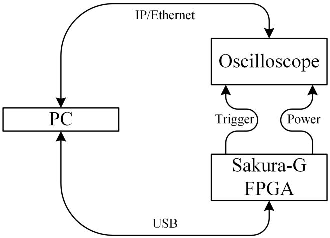
- parameters configuration (number of channels, trigger event and number of samples)
- download measurements and save on the disk.

PC - FPGA (USB)

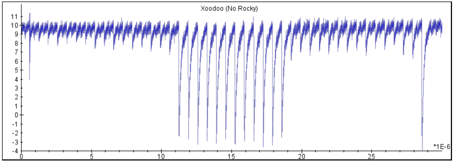
- send shift value υ and the Xoodoo state (48 bytes)
- signal to start
- verification of result

FPGA - Oscilloscope

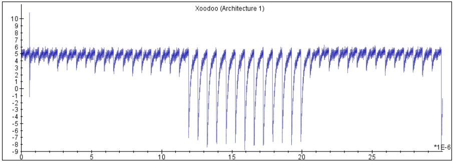
- trigger to start
- power trace acquisition



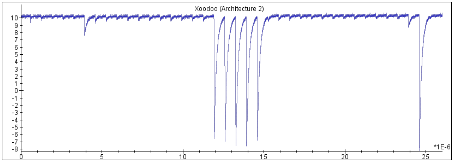
Unprotected



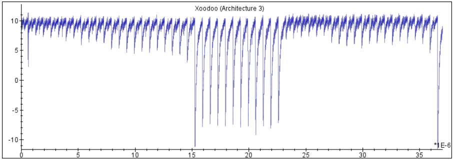
Architecture 1



Architecture 2



Architecture 3



Test Vector Leakage Assessment (TVLA)

- Proposed as an alternative leakage evaluation methodology against the complexity and amount of different side-channels attacks.

The core idea of TVLA

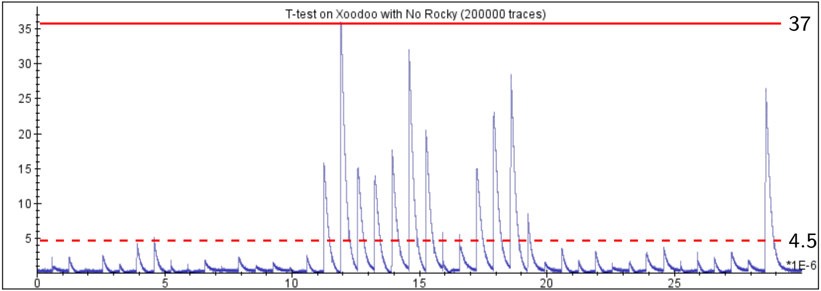
- Compute the t-test statistic between two sets of measurements:

$$\frac{|\mu_A - \mu_B|}{\sqrt{\frac{\sigma_A^2}{N_B} + \frac{\sigma_A^2}{N_A}}},$$

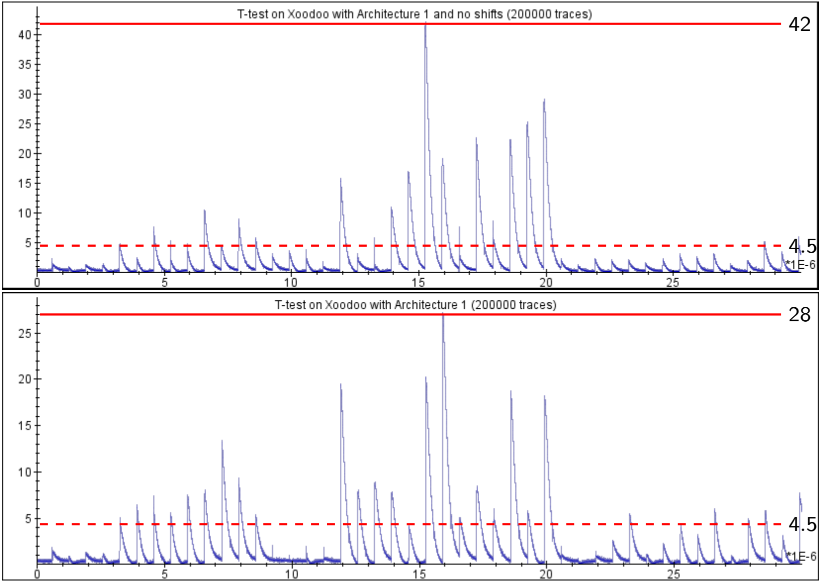
  where μ_x is the average of all the traces, σ_x the standard deviation and N_x the number of traces in each group x.
- One set is initialized with one or more fixed inputs and the other set with random inputs.
- Traditionally, the threshold of t-test value that indicates leakage is 4.5.

TVLA results (unprotected)

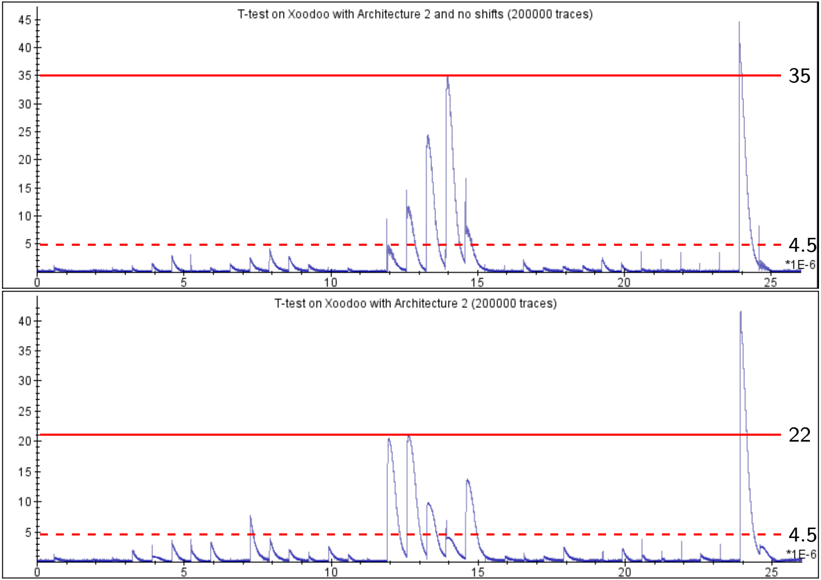
- 200K power traces
  - half with fixed input and half with random input State
- analysis with Riscure software



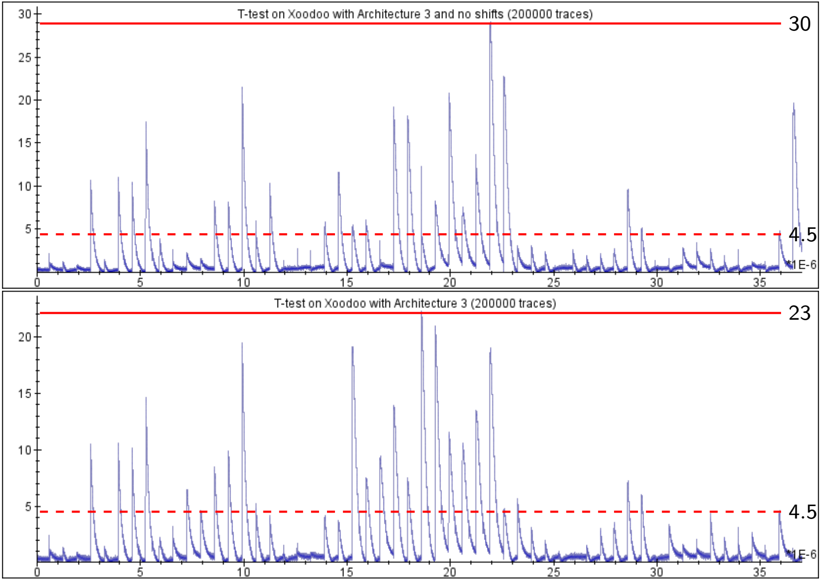
TVLA results (Architecture 1)



TVLA results (Architecture 2)



TVLA results (Architecture 3)



Summary

- All architectures show significant first-order leakage (t-test value threshold 4.5).
- We focus the analysis on the highest of the peaks.

	Disabled	Enabled	*	**
Unprotected	37	-	-	-
Architecture 1	42	28	33%	24%
Architecture 2	35	22	37%	40%
Architecture 3	30	23	23%	37%

* % difference between Disabled and Enabled ROCKY architectures
** % difference between Unprotected and Enabled ROCKY architectures

Conclusions

- We implemented an unprotected and three ROCKY-protected FPGA architectures of Xoodoo and perform a TVLA analysis.
- The results show that ROCKY (with 5-bit randomness) improves the side-channel resistance of the implemented cipher above 20% with no additional overhead.
- TVLA analysis limitations:
  - a negative test for leakage does not mean that the device is secure.
  - a positive indication of leakage, does not imply that the leakage can be exploited by an adversary.
- Therefore, more extensive power analysis and more sophisticated attacks will be performed in the future to determine the resistance of ROCKY against both type of attacks FI and SCA.

Rocky

Thanks you all!