Indian Institute of Technology Kharagpur

Telecom ParisTech

From Theory to Practice: Private Circuit and Its Ambush

Debapriya Basu Roy, Shivam Bhasin, Sylvain Guilley, Jean-Luc Danger and Debdeep Mukhopadhyay

Introduction

- Side Channel: Information leakage from the implementation

Introduction

- Side Channel: Information leakage from the implementation
- Probing Attack

Introduction

- Side Channel: Information leakage from the implementation
- Probing Attack
- Power attack and EM radiation attack

Introduction

- Side Channel: Information leakage from the implementation
- Probing Attack
- Power attack and EM radiation attack
- t-Private Circuit: Countermeasure design with sound theoretical proof

Introduction

- Side Channel: Information leakage from the implementation
- Probing Attack
- Power attack and EM radiation attack
- t-Private Circuit: Countermeasure design with sound theoretical proof
- Overhead: $O(nt^2)$, n is the number of gates in the circuit.

Related Works

- Masking: by product of private circuit to protect against first order differential attacks.

Related Works

- Masking: by product of private circuit to protect against first order differential attacks.
- Reducing Overhead: From $O(nt^2)$ to $O(nt)$, further reducing it to $\lceil t/2 \rceil$.

Related Works

- Masking: by product of private circuit to protect against first order differential attacks.
- Reducing Overhead: From $O(nt^2)$ to $O(nt)$, further reducing it to $\lceil t/2 \rceil$.
- Designing block ciphers with reduced number of *AND* operations, for example: *PICARO*.

Related Works

- Masking: by product of private circuit to protect against first order differential attacks.
- Reducing Overhead: From $O(nt^2)$ to $O(nt)$, further reducing it to $\lceil t/2 \rceil$.
- Designing block ciphers with reduced number of *AND* operations, for example: *PICARO*.
- Modifying private circuit for efficient FPGA implementation.

Motivation

- Private circuit is based on sound theoretical proof but with some inherent assumptions which may not be valid in practical scenario.

Motivation

- Private circuit is based on sound theoretical proof but with some inherent assumptions which may not be valid in practical scenario.
- Theoretical analysis of private circuit for power analysis in presence of glitches has been studied.

Motivation

- Private circuit is based on sound theoretical proof but with some inherent assumptions which may not be valid in practical scenario.
- Theoretical analysis of private circuit for power analysis in presence of glitches has been studied.
- However, no practical evaluation of private circuit is present in the literature

Contribution

- We identify the practical scenarios in which private circuit may fail to provide us the desired security.

Contribution

- We identify the practical scenarios in which private circuit may fail to provide us the desired security.
- We actually try to identify the *lazy engineering* practices which can rattle the security of private circuit.

Contribution

- We have implemented a lightweight block cipher *SIMON* using private circuit methodology on SASEBO-GII board.

Contribution

- We have implemented a lightweight block cipher *SIMON* using private circuit methodology on SASEBO-GII board.
- The implemented private circuits are analyzed against SCA using EM traces and correlation power analysis.

Contribution

- We have implemented a lightweight block cipher *SIMON* using private circuit methodology on SASEBO-GII board.
- The implemented private circuits are analyzed against SCA using EM traces and correlation power analysis.
- Moreover, we have used *Test Vector Leakage Assessment (TVLA)* methodology based leakage detection to classify our design as side channel secure or not.

Contribution

- *Optimized SIMON*: SIMON is implemented according to the ISW scheme, but the design tool is free to optimize the circuit. This is an example of *lazy engineering* approach,

Contribution

- *Optimized SIMON*: SIMON is implemented according to the ISW scheme, but the design tool is free to optimize the circuit. This is an example of *lazy engineering* approach,
- *2-input LUT based SIMON*: Here, to mimic the private circuit methodology exactly on the FPGA, we have constrained the design tool to map each two-input gate to a single LUT. In other words, though a LUT has six inputs, it is modeled as two-input gate and gate-level optimization is minimized.

Contribution

- *Optimized SIMON*: SIMON is implemented according to the ISW scheme, but the design tool is free to optimize the circuit. This is an example of *lazy engineering* approach,
- *2-input LUT based SIMON*: Here, to mimic the private circuit methodology exactly on the FPGA, we have constrained the design tool to map each two-input gate to a single LUT. In other words, though a LUT has six inputs, it is modeled as two-input gate and gate-level optimization is minimized.
- *Synchronized 2-input LUT based SIMON*: This is nearly similar to the previous methodology. The only difference is that each gate or LUT is preceded and followed by flip-flops so that each and every input to the gates is synchronized and glitches are minimized .

Preliminaries

SIMON

In 2013, NSA had introduced two ultra-lightweight block cipher SIMON and SPECK with a Feistel construction. Out of the two block ciphers, SIMON is more suited for hardware implementations. SIMON can encrypt a block of $2k$ bits, with a key of $m \cdot k$ bits.

Preliminaries

SIMON

In 2013, NSA had introduced two ultra-lightweight block cipher SIMON and SPECK with a Feistel construction. Out of the two block ciphers, SIMON is more suited for hardware implementations. SIMON can encrypt a block of $2k$ bits, with a key of $m \cdot k$ bits.

TVLA

TVLA consists in operating the device under test with *a fixed and chosen key*. Then, a T-test is applied on both sets of measurements. Similar difference testing can be performed on intermediate values of the block cipher and also on each bit of that intermediate value.

t-Private Circuit

- Input Encoding: A vector of $(a_1, a_2, ..., a_{2t}, a_{2t+1})$

$$a_{2t+1} = a \oplus \bigoplus_{i=1}^{2t} a_i \ .$$

t-Private Circuit

- Input Encoding: A vector of $(a_1, a_2, ..., a_{2t}, a_{2t+1})$

$$a_{2t+1} = a \oplus \bigoplus_{i=1}^{2t} a_i \ .$$

- *NOT gate*: $\grave{a} = (a_1, a_2, ..., a_{2t+1})$. $\grave{\grave{a}} = (a_1, a_2, ..., \overline{a_{2t+1}})$.

t-Private Circuit

- Input Encoding: A vector of $(a_1, a_2, ..., a_{2t}, a_{2t+1})$

$$a_{2t+1} = a \oplus \bigoplus_{i=1}^{2t} a_i \ .$$

- *NOT gate*: $\grave{a} = (a_1, a_2, ..., a_{2t+1})$. $\grave{\grave{a}} = (a_1, a_2, ..., \overline{a_{2t+1}})$.
- *Xor gate*:

$$c_i = a_i \oplus b_i, 1 \leq i \leq 2t \ .$$

t-Private Circuit

- *AND gate*: Inputs $\grave{a} = (a_1, a_2, ..., a_{2t+1})$ and $\grave{b} = (b_1, b_2, ..., b_{2t+1})$, output $\grave{c} = (c_1, c_2, ..., c_{2t+1})$, which is calculated by following steps:
  1. Generate random bits $r_{i,j}$, where $i \neq j$ and $1 \leq i \leq j \leq 2t+1$.
  2. Compute $r_{j,i} = (r_{i,j} \oplus a_i b_j) \oplus a_j b_i$, where $i \neq j$ and $1 \leq i \leq j \leq 2t+1$.
  3. Compute $c_i = a_i b_i \oplus \bigoplus_{j \neq i} r_{i,j}$, where $1 \leq i \leq 2t$ and $1 \leq j \leq 2t$.

AND Gate Example

- Inputs of the *AND* gate are two vectors $\grave{a} = (a_1, a_2, a_3)$ and $\grave{b} = (b_1, b_2, b_3)$, Output $\grave{c} = (c_1, c_2, c_3)$ is calculated as follows:

$$c_1 = a_1 b_1 \oplus r_{1,2} \oplus r_{1,3} \tag{1}$$
$$c_2 = a_2 b_2 \oplus (r_{1,2} \oplus a_1 b_2) \oplus a_2 b_1 \oplus r_{2,3} \tag{2}$$
$$c_3 = a_3 b_3 \oplus (r_{1,3} \oplus a_1 b_3) \oplus a_3 b_1 \oplus (r_{2,3} \oplus a_2 b_3) \oplus a_3 b_2 \tag{3}$$

Figure: $t = 1$ private circuit for AND third coordinate on 4-input LUTs

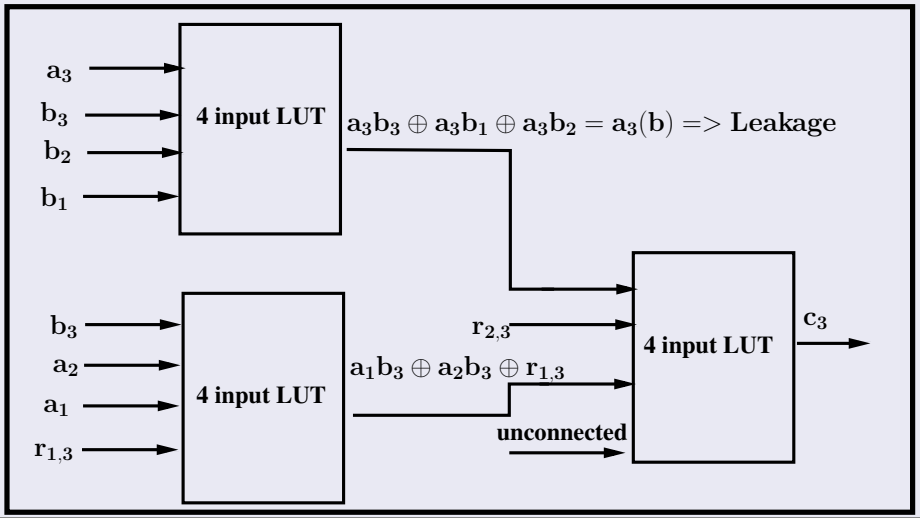
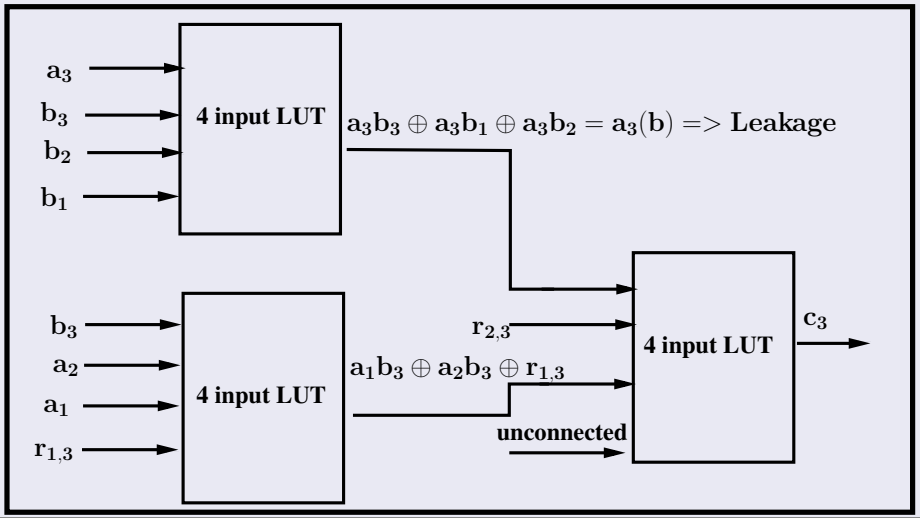


Figure: $t = 1$ private circuit for AND third coordinate on 4-input LUTs



$$\begin{cases} p(b=0|x=0) &= 2/3, \\ p(b=1|x=0) &= 1/3, \end{cases} \quad \text{and} \quad \begin{cases} p(b=0|x=1) &= 0, \\ p(b=1|x=1) &= 1. \end{cases} \tag{4}$$

Delay in Random Variables

- There are two ways in which random variables can be provided to the private circuit: as external input or from a *Random Number generator (RNG)*. Generally, random numbers are provided to the circuit from an *RNG*.

Delay in Random Variables

- There are two ways in which random variables can be provided to the private circuit: as external input or from a *Random Number generator (RNG)*. Generally, random numbers are provided to the circuit from an *RNG*.
- $$c_3 = a_3 b_3 \oplus (r_{1,3} \oplus a_1 b_3) \oplus a_3 b_1 \oplus (r_{2,3} \oplus a_2 b_3) \oplus a_3 b_2 \quad (5)$$

  Delay in the arrival of random bits $r_{1,3}$, $r_{2,3}$, a_1 and a_2 lead to information leakage.

Experimental Setup

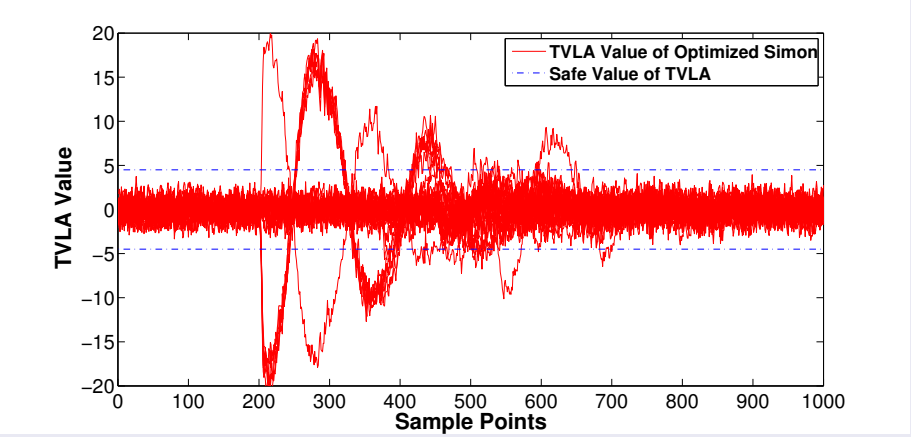
- A parallel implementation SIMON32/64 crypto-core, running at clock frequency of 24-MHz, along with a simple UART interface is used to test our design on the Xilinx Virtex `XC5-VLX30` FPGA of the SASEBO-GII platform.

Experimental Setup

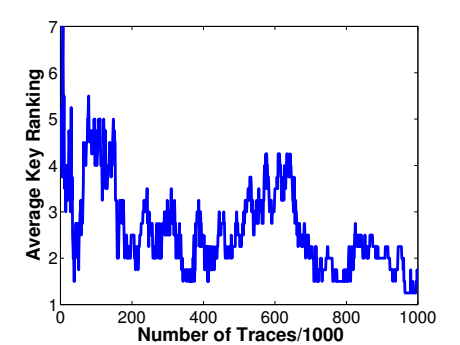
- A parallel implementation SIMON32/64 crypto-core, running at clock frequency of 24-MHz, along with a simple UART interface is used to test our design on the Xilinx Virtex `XC5-VLX30` FPGA of the SASEBO-GII platform.
- For $t = 1$, total number of random bits required by SIMON is 272, whereas for $t = 2$ and $t = 3$, number of required random bits become 608 and 1008. Random numbers are generated by a maximal length LFSR.

Result: Optimized Simon

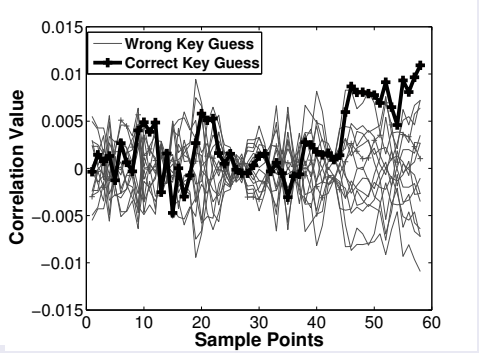
(a) TVLA Plot



(b) Average Key Ranking

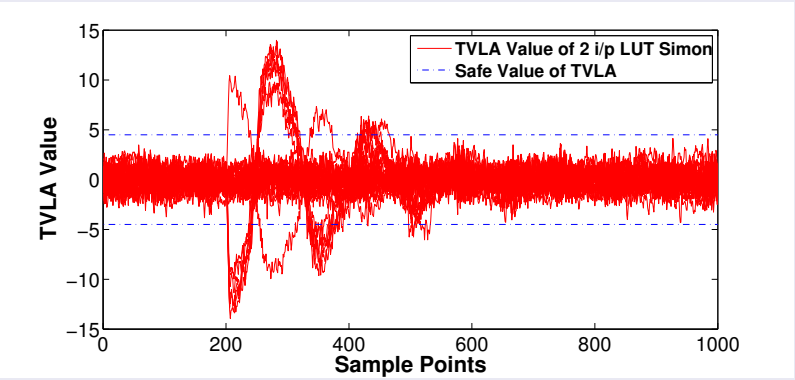


(c) Correlation Value

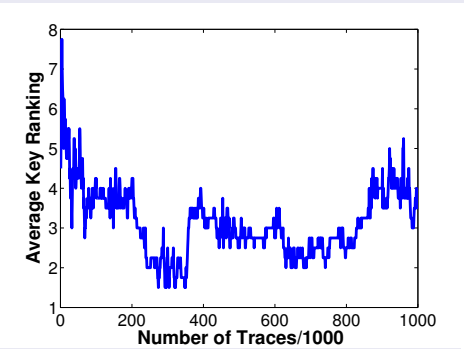


Result: 2 input LUT based Simon

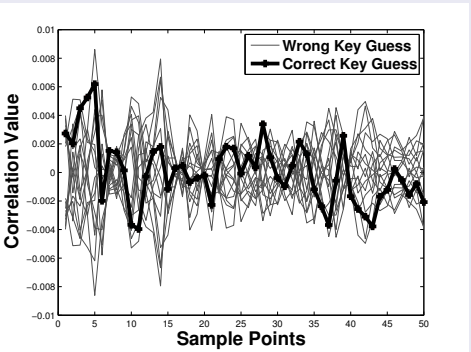
(a) TVLA Plot



(b) Average Key Ranking

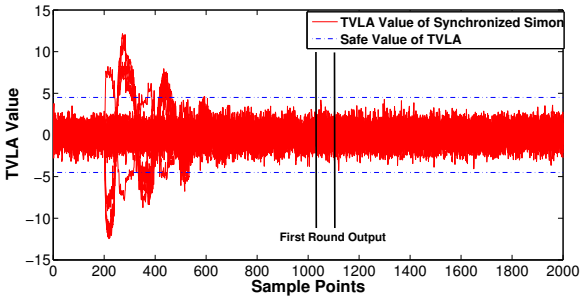


(c) Correlation Value

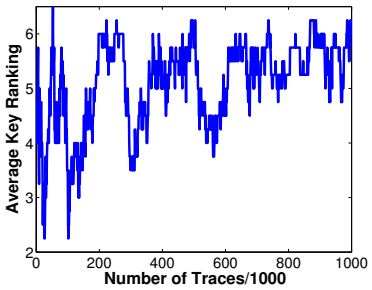


Synchronized 2-input LUT based SIMON

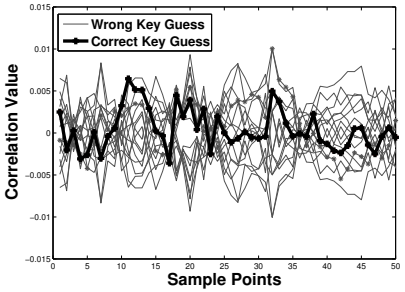
(a) TVLA Plot



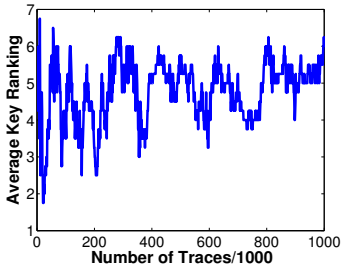
(b) Avg. Key Rank_P)



(c) Correlation Value_P



(d) Avg. Key Rank_1



(e) Correlation Value_1

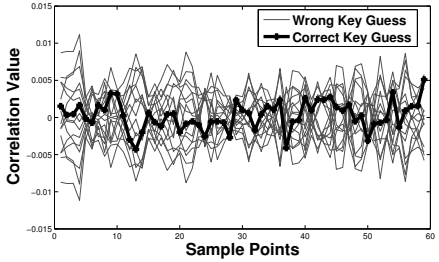


Figure: Side Channel Analysis of Synchronized 2 input LUT SIMON

Attack Summary

Table: Summary of Side Channel Analysis

Design Name	**TVLA Test**	**Avg. Key Ranking**	**Remarks**
Optimized SIMON	Fails, significant information leakage	Key ranking is low, successful attack	Not secure
2 input LUT based SIMON	Fails, but less information leakage compared to optimized SIMON	Key ranking is high, attack is not successful	Secure against CPA, could be broken by better model
Synchronized 2 input LUT based Simon	Passes: no leakage at first round. Initial peaks are caused by plain-text loading	Key ranking is high, attack is not successful	Secure

Resource Comparison

Name	LUTs	Registers	Slices	Freq. (MHz)	Clock Cycles
Optimized SIMON	761 (1×)	805 (1×)	595 (1×)	147 (1×)	32 (1×)
2 i/p LUT based SIMON	1305 (1.71×)	805 (1×)	1241 (2.08×)	88 (0.59×)	32 (1×)
Synchronized 2 i/p LUT based SIMON	1309 (1.71×)	2920 (3.62×)	4090 (6.87×)	104 (0.70×)	288 (9×)

Conclusion

- We analyzed private circuits at an implementation level on a SIMON crypto-processor. Our results show that it is very easy for a CAD tool to override the basic requirements of private circuits.

Conclusion

- We analyzed private circuits at an implementation level on a SIMON crypto-processor. Our results show that it is very easy for a CAD tool to override the basic requirements of private circuits.
- Practical evaluations indicate that with proper constraints the leakage can be reduced. Moreover, by synchronizing each gate, we remove glitches and delay and approach much closer to theoretical evaluation of private circuits, but at a huge overhead.