Hardware Masking, Revisited

Thomas De Cnudde

Maik Ender
Amir Moradi

September 11 - CHES 2018 - Amsterdam
Towards Sound Approaches to Counteract Power-Analysis Attacks

The instantaneous power consumption of the chip shortly after a clock edge is a combination of the consumption components from each of the events that have occurred since the clock edge. Each event’s timing and power consumption depends on physical and environmental factors such as the electrical properties of the chip substrate, layout, temperature, voltage etc., as well as coupling effects between events of close proximity. As a first approximation, we ignore coupling effects and create a linear model, i.e., we assume that the power consumption function of the chip is simply the sum of the power consumption functions of all the events that take place.

We checked the influence of these parameters on the leakage.
What do we control in the measurement setup and in the implementation?

- Supply Voltage
- Shunt Resistor
- Distance between the shares
- Temperature
- Circuit Size
- Clock Frequency
- Number of Shares
Hardware Masking, Revisited

1. Experiments on a Toy Example
to reveal the influence of the various parameters on the leakage

2. Can we make Masked Implementations leak?

3. Conclusions
   - Summary
   - Implications
Hardware Masking, Revisited

Experiments on a Toy Example

revealing the influence on the leakage by

1. the various parameters
2. coupling of FPGA wires

Can we make Masked Implementations leak?

Conclusions
One share in our toy example consists of consecutive MixColumn modules.

We can choose between
- a lower power consumption from 3 MCs
- a higher power consumption from 6 MCs
Four iterated MixColumns shares are placed next to each other and in full isolation.

We can test the influence of the number of shares and the distance between the shares on the leakage.
Supply Voltage & Shunt Resistor

The higher the supply voltage, the higher the leakage

The lower the shunt resistor, the higher the leakage

Fixed-vs-random t-test
iterated_MC1 and iterated_MC4
3 MCs active
6MHz
21°C
Distance

The distance between the shares does not influence leakage much.

Fixed-vs-random t-test
3 MCs active
6MHz
V_{dd} 1.2\text{v}, 0.0\Omega, 21^\circ C
The higher the temperature, the higher the leakage.

The diagram shows the relationship between the number of traces and the maximum t-statistic for different temperatures: 21°C, 50°C, and 70°C. The graph indicates that as the temperature increases, the leakage also increases.

Additional notes:
- Fixed-vs-random t-test
- iterated_MC1 and iterated_MC4
- 3 MCs active
- 6MHz
- $V_{dd} 1.3v, 0.0\Omega$
Circuit Size and Clock Frequency

The more MCs active, the higher the leakage
The higher the peak-to-peak power consumption, the higher the leakage

Fixed-vs-random t-test
iterated_MC1 and iterated_MC4
$V_{dd}$ 1.3v, 0.0Ω, 21°C
Number of Shares

All linear 1\textsuperscript{st}-, 2\textsuperscript{nd}- and 3\textsuperscript{rd}-order designs leak in the 1\textsuperscript{st}-order!

No 2\textsuperscript{nd}-order leakage in the 2\textsuperscript{nd}-order secure design

No 2\textsuperscript{nd}- or 3\textsuperscript{rd}-order leakage in the 3\textsuperscript{rd}-order secure design

6 MCs active
6MHz

V_{dd} 1.3v, 0.0Ω, 21°C
Hardware Masking, Revisited

1. Experiments on a Toy Example
   revealing the influence on the leakage by
   1. the various parameters
   2. coupling of FPGA wires

2. Can we make Masked Implementations leak?

3. Conclusions
Does leakage current in open switch transistors contribute to the leakage in FPGAs?
Open transistors inside a switch matrix could couple wires from two different shares.

What is the influence of the number of shared open switches?
Our experiments are designed with an increasing number of shared open switches.
We route two wires close to each other in the middle of two iterated_MC shares
Number of shared open switches

Routing does not have much effect on the observed leakage

Fixed-vs-random t-test
6 MCs active
V_{dd} 1.3v, 0.0Ω, 6MHz, 21°C
Hardware Masking, Revisited

1. Experiments on a Toy Example

2. Can we make Masked Implementations leak?
   - Threshold Implementation of PRESENT
   - Domain-Oriented Masking of AES
   - d+1 Threshold Implementations of AES

3. Conclusions
A Threshold Implementation of PRESENT

PRESENT-80 TI from [PMK+11]

1\textsuperscript{st}-order implementation with 3 shares leaks in the 1\textsuperscript{st} order

Fixed-vs-random t-test
8 rounds of encryption
12MHz
21°C
Masked AES implementations with d+1 shares

Domain-Oriented Masking from [GMK16]

Fixed-vs-random t-test
full encryption

24MHz

V_{dd} 1.2v, 1.0\,\Omega, 21^\circ C
Hardware Masking, Revisited

1. Experiments on a Toy Example
2. Can we make Masked Implementations leak?
3. Conclusions
   - Summary
   - Implications
Hardware Masking, Revisited

Can we make Masked Implementations leak?

YES!
Summary for the Spartan-6 FPGA

Higher leakage with
1. higher supply voltages
2. lower shunt resistors
3. higher temperatures
4. higher peak-to-peak power consumption
   (higher clock frequency or larger circuits)
5. lower number of shares

Leakage does not depend much on
1. the distance between the shares
2. the leakage current from open transistors between the shares
Implications

Assumptions can be violated!
Not surprising, e.g. glitches, early signal propagation

Correctly masked implementations leak?
Yes, with a high number of traces in a low noise environment

**Can this be exploited by an attacker? How?**

What about ASICs?
Likely more traces needed...
Potential Solutions

Temporal non-completeness?
  Don’t process on more than d shares per clock cycle for d\(^{th}\)-order security
  Expensive...

Embedded voltage regulators?
  Do EM signals show similar issues?

Sharing the V\(_{dd}\) lines?
  Not clear how to apply nonlinear functions in this setting...

Use the leakage detection in addition to attacks?
  Moments-Correlating DPA [MS16]