Does Coupling Affect the Security of Masked Implementations?

Thomas De Cnudde[1], Begül Bilgin[1], Benedikt Gierlichs[1], Ventzislav Nikov[2], Svetla Nikova[1], and Vincent Rijmen[1]

[1] KU Leuven, ESAT-COSIC and iMinds, Belgium
{name.surname}@esat.kuleuven.be
[2] NXP Semiconductors, Belgium venci.nikov@gmail.com

Abstract. Masking schemes achieve provable security against side-channel analysis by using secret sharing to decorrelate key-dependent intermediate values of the cryptographic algorithm and side-channel information. Masking schemes make assumptions on how the underlying leakage mechanisms of hardware or software behave to account for various physical effects. In this paper, we investigate the effect of the physical placement on the security using leakage assessment on power measurements collected from an FPGA. In order to differentiate other masking failures, we use threshold implementations as masking scheme in conjunction with a high-entropy pseudorandom number generator. We show that we can observe differences in—possibly—exploitable leakage by placing functions corresponding to different shares of a cryptographic implementation in close proximity.
Keywords: Masking, Threshold Implementations, Crosstalk, Non-independent leakage, Leakage detection, TVLA.

1 Introduction

Side-Channel Analysis (SCA) is a powerful menace against embedded cryptosystems. Whether timing information [22], instantaneous power consumption [23] or electromagnetic radiation [17,34] is exploited, extracting sensitive information (e.g. secret keys) from cryptographic devices is feasible in contrast to cryptanalytic or brute force techniques. Counteracting SCA has consequently been an active research topic and many countermeasures have been proposed. In this paper we focus on masking methods which provide provable security given certain assumptions on the implementation, physical behavior of the device and capability of an attacker.

Masking. Masking [9, 19], which is based on secret sharing and multi-party computation, relies on randomizing the intermediate values and computations based on them. For this purpose each sensitive value $x \in \mathrm{GF}(2^m)$ is uniformly and randomly split into s shares using a certain operation $\perp$ such that the following condition holds:

$$x = x_1 \perp x_2 \perp ... \perp x_{s-1} \perp x_s$$

Typically, it is assumed that the physical leakage of calculation and storage of each share is independent of the others. With this assumption, the security proofs of masking schemes consider an attacker capable of observing leakages coming from calculation or storage depending jointly on at most d shares and hence performing d^{th}-order attacks. Therefore, $s \geq d + 1$ is a natural bound as this implies incomplete information for the attacker. Besides these uniformly distributed input and independent leakage assumptions, different flavors of masking schemes may have additional computational or behavioral requirements on the implementation. Keeping the calculation order as is defined in Trichina AND gate [40] and having ideal nodes that do not toggle in private circuits [21] are well known examples. In this paper, we focus on the failure of the independent leakage assumption and satisfy further restrictions to the utmost.

Failure of Independent Leakage. The theoretical security of masking schemes degrades when the leakage of different shares get influenced by each other. The amount of this security reduction has been investigated theoretically in [14] with respect to the strength of joint leakage in comparison to independent leakages (called the flaw constant) and to noise level. It has been shown that mutual information increases together with the flaw constant. On the other hand, second-order leakage can become easier to detect than first-order leakage as the noise increases given enough dependent leakage.

In practice, Hamming Distance (HD) leakage from one share to another and glitchy gates are natural and visible examples of non-independent leakage. It is shown in [3] that a theoretically d^{th}-order secure implementation can be attacked using $d/2^{\text{th}}$-order attack in practice due to HD leakage if the security proofs assume Hamming Weight (HW) leakage. Moreover, classical Boolean masking is shown to be futile in circuits using CMOS-like technology [25]. The temporally separated masking scheme of Prouff and Roche [33], where shares are required to interleave their computations, have also been argued to be vulnerable when static leakage is measurable [27]. In order to distinguish undesired security degradation caused by HD leakage and redundant toggling of gates from other failures of independent leakage, we ensure not to have HD leakage between different shares of the same unmasked value and use threshold implementation (TI) masking scheme which provides security in glitchy circuits [5, 29–31].

Another example of non-independent leakage is crosstalk, which originates from coupling capacitors between circuit wires, and between circuit wires and ground. Coupling capacitance between two wires is influenced by the switching activity on that wire. Only a few publications have investigated the effect of crosstalk within the field of SCA attacks so far. In [10], Chen et al. showed that the leakage intensity of glitches and the leakage caused by inter-wire capacitance are comparable using SPICE simulations. Moreover, they retrieved the key successfully using first-order attacks on a masked implementation with dual-rail pre-charge logic. This logic style was thought to avoid non-independent leakages caused by glitching implying crosstalk to be the main leakage leading to the attack. However, the latter results based on real-world devices are considered to

have measured the effect of early propagation issues in implementations using these logic styles [41] rather than crosstalk itself. Later, Dyrkolbotn considered the layout dependent phenomena of capacitive crosstalk in [15, 16] in order to derive a more precise leakage model. They showed that the detection performance of values on an 8-bit data bus increases from 2.5-bits of information per sample with a Hamming Distance detector to a theoretical 5.7-bit and simulated 5-bit of information per sample with a crosstalk based detector by simulation. Power supply noise or IR drop, another coupling effect in circuits, was also shown to have a negative impact on the security of a countermeasure [44] by relating independent logic gates through the power supply line. Finally, Schmidt et al. performed successful key-retrieval attacks by measuring the power consumption on input or output peripherals instead of using the regular power supply lines [36]. The success of their method originates from the coupling between pins of an Integrated Circuit (IC).

To conclude, there is no definitive report on the observability of non-independent leakage originating from coupling on a real-world device when masking is considered. In order to distinguish between non-independent leakage originating from e.g. HD or glitches, and leakage originating from coupling, we will refer to the latter as out-of-model leakage .

Leakage Assessment. The security of a masked implementation is commonly assessed using side-channel evaluation platforms and techniques like Differential Power Analysis (DPA) [23], Correlation Power Analysis (CPA) [7] or Test Vector Leakage Assessment (TVLA) [12, 18, 37, 38]. Unlike other evaluation methods, TVLA has the advantage of being very sensitive even if the detected leakage does not necessarily lead to key recovery. Therefore, it is a preferred tool to confirm the provable security of a masking schemes with high confidence [5, 11, 39]. In order to observe the possibly small differences in observable leakage caused by having or lacking coupling-like, out of model behavior, we opt for TVLA in this paper.

Contribution. In this work we further build on the observations of out-of-model leakage . In contrast to the WDDL enhanced masked AES S-box of Chen et al. [10], our focus is specifically directed towards masking schemes alone and the Threshold Implementations scheme is selected as test case. We choose the lightweight KATAN-32 [8] as our target block cipher as we expect coupling effects to be more prominent in a low noise setting. After showing a secure TI of the lightweight KATAN-32 block cipher, we investigate the out-of-model leakage when we induce coupling between shares on an FPGA.

Organization. In Section 2, we give an overview of the internal mechanism of two out-of-model leakage sources and revisit the KATAN-32 Threshold Implementation and briefly introduce FPGA concepts used throughout the paper. In Section 3, we theoretically evaluate the effect of out-of-model leakage on the conditions of a three share masking scheme. We describe and evaluate two leakage

scenarios on the KATAN-32 Threshold Implementation in Section 4 and follow with a brief discussion and a conclusion in Section 5.

2 Preliminaries

2.1 Sources of Out-of-Model Leakage

We now revisit the conditions for masking from a power consumption point of view and give simplified models of physical phenomena that are known to lead to out-of-model leakage [24].

Power Consumption in Masking Schemes. From a power consumption perspective, a first-order masked implementation requires the following condition to hold: the mean power consumption for each unmasked sensitive value should be equal. One way to achieve this requirement is by using Boolean masking with masks drawn randomly from a uniform distribution.

If we mask a one-bit secret value x with a one-bit mask m as $\mathbf{x} = (s_1, s_2) = (x \oplus m, m)$ and denote the probability of $m = i$ by K_i, we can formalize the condition for the uniformity of the masks as:

$$K_0 = K_1 = \frac{1}{2}\,.$$

The expected power consumption P w.r.t. the unmasked value x can then be expressed as:

$$\begin{aligned} P(x=0) &= K_0 P(s_1 = 0, s_2 = 0) + K_1 P(s_1 = 1, s_2 = 1) \\ P(x=1) &= K_0 P(s_1 = 1, s_2 = 0) + K_1 P(s_1 = 0, s_2 = 1)\,. \end{aligned}$$

The condition for first-order Boolean masking is then formalized by the following equation:

$$P(s_1 = 0, s_2 = 0) + P(s_1 = 1, s_2 = 1) = P(s_1 = 0, s_2 = 1) + P(s_1 = 1, s_2 = 0)\,.$$

In this example, first-order vulnerabilities occur in the masking scheme when this condition is violated. The effect of out-of-model leakage from coupling on the security of the masking scheme can be understood by analyzing the power consumption P [35]. The instantaneous power consumption P_{inst} represents a sample of a SCA measurement trace:

$$P_{inst} = I_{inst} V_{inst}\,.$$

Where I_{inst} and V_{inst} denote the instantaneous current and instantaneous voltage respectively.

Crosstalk. Crosstalk is the result of capacitive coupling between adjacent wires. Figure 1 shows two adjacent wires, each with a parasitic capacitance to the IC substrate and an inter-wire capacitance between them. When a wire (the aggressor) switches the value it carries, another wire in its vicinity (the victim) will be influenced through the inter-wire capacitance $C_{1,2}$ between the aggressor and the victim. This influence can range from increased delay of a signal to traverse the wire, through a wrong value being temporarily induced on the victim. The reduction in SCA security introduced by crosstalk can be explained as follows. A typical first-order masked implementation represents a sensitive variable by two randomized shares, such that the mean power consumption of either share is independent of the other share. If two wires belonging to different shares are coupled, the mean power consumption of one share depends jointly on both a neighboring aggressor share and itself. The masked implementation is hence rendered insecure.

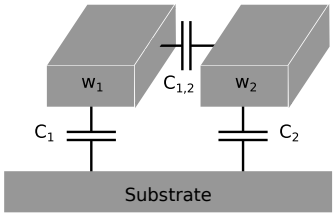

Fig. 1: Crosstalk originates from the inter-wire capacitance $C_{1,2}$.

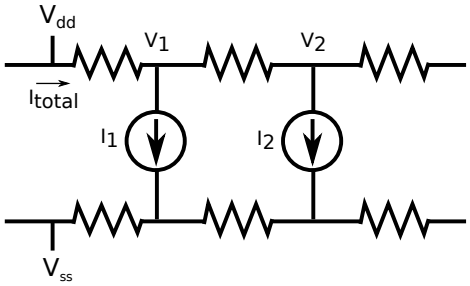

Fig. 2: Static and dynamic IR drop occurs from the non-zero resistance of conductive supply voltage and ground wires.

IR Drop. IR drop or power supply noise originates from the finite conductance of wires in i.a. the power distribution grid of ICs. Every wire segment has a small resistance associated with it leading to a drop in the power supply voltage when a current flows through that wire [35]. Both static and dynamic IR drop can lead to coupling between shares and hence to out-of-model leakage . A simplified model is given in Figure 2.

As with crosstalk, the problem gets worse with shrinking technology nodes [35]. In the context of SCA, the effect of IR drop has not yet been investigated.

2.2 KATAN-32 and Its Threshold Implementation

KATAN is a set of block ciphers designed specifically for lightweight applications [8]. Its efficiency in hardware translates to a small area and a low power

consumption. Three options are available for the state size: 32-, 48- or 64-bit. All options use an 80-bit key, making the security independent of the state size.

The diagram of the KATAN-32 round function is shown in Figure 3. The 32-bit plaintext is stored in a state that consists of two shift registers: a 13-bit right shifting register $L1$ and a 19-bit left shifting register $L2$. The cipher processes the state by applying a round operation 254 times. The round operation relies on a small number of AND and XOR gates and is performed on several bits in order to update the first bits of $L1$ and $L2$. The function is of the form $A = f(X, Y, Z) = X \oplus YZ$. The IR bit represents the last bit of the round counter which enables or disables the fourth bit of $L1$ in the round operation. The bits k_{2i} and k_{2i+1} are the $2i^{th}$ and $(2i+1)^{th}$ bits of the 80-bit key for rounds $i \leq 40$. In rounds $i > 40$, they are derived from the original key by an LFSR. The full description can be found in [8].

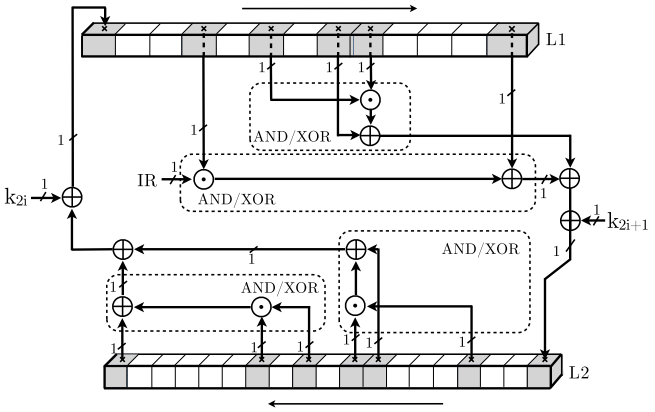

Fig. 3: KATAN-32 consists of two sets of shift registers and four nonlinear operation groups (Source: [5]).

The round operation is susceptible to glitching, making TI a natural choice for a masked implementation. This was shown by Bilgin et al. in [5] where a first-, a second- and a third-order Threshold Implementation of KATAN-32 were presented. We now revisit their first-order TI of KATAN-32.

The focus lies on the sharing of the state and its nonlinear round function since sharing nonlinear operations is more involved than sharing linear ones. An unshared key and key schedule are used, such that the key addition only needs to be performed on the first share of the state.

Since a first-order three share TI of a single AND gate with uniform outputs does not exist [29] without remasking, the AND gates of the round operations are always grouped with an XOR gate and masked using the uniform TI with $s_{in} = s_{out} = 3$ shares of the function $A = f(X, Y, Z) = X \oplus YZ$. This approach

results in the following non-complete sharing:

$$a_1 = x_2 \oplus (y_2 z_2 \oplus y_2 z_3 \oplus y_3 z_2)$$
$$a_2 = x_3 \oplus (y_3 z_3 \oplus y_3 z_1 \oplus y_1 z_3)$$
$$a_3 = x_1 \oplus (y_1 z_1 \oplus y_1 z_2 \oplus y_2 z_1)\,.$$

Since the round counter (and resultantly IR) is not key dependent and hence not shared, IR is added to the AND/XOR blocks in the following way:

$$a_i = x_i + IR \times y_i, \quad i \leq s_{in}\,.$$

The number of state shares is chosen to be three following the number of shares of the nonlinear function.

2.3 Xilinx Virtex-II Pro FPGA Overview

In order to help the reader understand the FPGA related details, we first briefly review the hardware architecture and development flow, and highlight only the concepts we will use throughout this paper. We focus our discussion specifically on the Virtex-II Pro FPGA, since it forms the target device of our implementations.

Hardware Architecture. Field Programmable Gate Arrays (FPGAs) are a type of programmable ICs containing a regular grid of Configurable Logic Blocks (CLBs) and programmable routing resources. In the Virtex-II Pro FPGA, each CLB contains four slices, which are the primitive building blocks of the FPGA. Each slice contains amongst others two 4-input Look-Up Tables (LUTs) and two registers [43].

Design Flow. The classic design flow for FPGAs starts with the Hardware Description Language (HDL). During synthesis, the HDL files are compiled and transformed into an FPGA architecture-specific design netlist. Once synthesis is completed, the next step in the design flow is the implementation which consists of three phases: translate, map and place and route (PAR). During translation, the netlist is reduced to only contain Xilinx primitives. The mapping phase then maps the Xilinx primitives in the netlist to actual FPGA resources such as slice registers or LUTs. After mapping an NGC netlist file is output that corresponds to the physical components in the Xilinx FPGA and the constraints of the design. The final stage of the implementation is the PAR. The actual allocation of resources from the NCF file and their interconnections are decided upon here. Once the implementation is finished, the bitstream file for the FPGA configuration is generated using the Generate Programming File process.

3 Coupling in Threshold Implementations

3.1 Crosstalk

Since we are interested in the effect of out-of-model leakage on the first-order security of the KATAN-32 TI with three shares, we first provide a discussion of crosstalk in masking schemes with three shares.

Masking the secret value x yields $\mathbf{x} = (s_1, s_2, s_3) = (x \oplus m_1 \oplus m_2, m_1, m_2)$, where the masks m_i are drawn from a uniform random source to satisfy the condition of masking, i.e.

$$K_{0,0} = K_{0,1} = K_{1,0} = K_{1,1} = \frac{1}{4}$$

where $K_{i,j}$ denotes the probability of $m_1 = i$ and $m_2 = j$.

The masking condition $P(x = 0) = P(x = 1)$ on the expected power consumption P w.r.t. the unmasked value x is then expressed as:

$$\begin{aligned} &P(s_1 = 0, s_2 = 0, s_3 = 0) + P(s_1 = 0, s_2 = 1, s_3 = 1) \\ &+P(s_1 = 1, s_2 = 0, s_3 = 1) + P(s_1 = 1, s_2 = 1, s_3 = 0) \\ &= P(s_1 = 0, s_2 = 0, s_3 = 1) + P(s_1 = 0, s_2 = 1, s_3 = 0) \\ &+P(s_1 = 1, s_2 = 0, s_3 = 0) + P(s_1 = 1, s_2 = 1, s_3 = 1)\,. \end{aligned}$$

In order to examine the influence of out-of-model leakage on the security of the masking scheme, we need to find whether or not a dependence exists between the instantaneous power consumption $P_{inst} = I_{inst}V_{inst}$ and the unmasked value x. In order to perform this exemplary analysis, we rely on a data bus model [13]. The relation of the instantaneous power with the consumed energy can be seen from the expression for the energy required to charge a wire i from a bus from $V_i(t^-) = 0$ to $V_i(t^+) = V_{dd}$:

$$E_{rise,i} = \int_{t^-}^{t^+} V_{dd} \cdot I_j(t)dt\,.$$

The total energy consumption to change a three wire bus can be written as:

$$E_{total} = \sum_{i=0}^{3}(1 + 2\lambda - \lambda\delta_{i,i-1} - \lambda\delta_{i,i+1}) \cdot C_L \cdot V_{dd} \cdot V_i$$

where $\lambda = C_I/C_L$ and $C_I = C_{1,2} = C_{1,3}$ and $C_L = C_1 = C_2 = C_3$ is assumed. Furthermore, $\delta_{i,j} \in \{-1, 0, 1\}$ is the normalized relative voltage change of the j^{th} line w.r.t. the i^{th} line, V_{dd} is the supply voltage and V_j is the final voltage on the j^{th} line.

We can now group and calculate the total energy transitions per unmasked value using values from [26] for $C_L = 400fF$, $C_I = 250fF$ and $V_{dd} = 3V$:

$$\begin{aligned} E_{total,0\to0} &= 0.430nJ & E_{total,1\to0} &= 0.529nJ \\ E_{total,0\to1} &= 0.475nJ & E_{total,1\to1} &= 0.498nJ\,. \end{aligned}$$

The difference in total energy per unmasked value is analytically distinguishable and hence the masking scheme is not secure in the presence of crosstalk.

3.2 IR Drop

Power supply noise or IR drop is a result of the finite conductance of wires. Figure 4 shows a simplified version of IR drop that focuses on shared subcircuits [4]. The influence of IR drop on the security of the masking scheme is best understood by looking at the instantaneous power consumption $P_{inst} = I_{inst}V_{inst}$ on the voltage nodes V_1, V_2 and V_3:

$$\begin{aligned} V_1 &= V_{dd} - (I_1 + I_2 + I_3)R_1 \\ V_2 &= V_{dd} - (I_1 + I_2 + I_3)R_1 - (I_2 + I_3)R_2 \\ V_3 &= V_{dd} - (I_1 + I_2 + I_3)R_1 - (I_2 + I_3)R_2 - I_3R_3\,. \end{aligned}$$

We can now write the instantaneous power consumption of all the shares $P_{inst,Share1}$, $P_{inst,Share2}$ and $P_{inst,Share3}$ as:

$$\begin{aligned} P_{inst,Share1} &= I_1V_1 = V_{dd}I_1 - I_1^2R_1 - I_1I_2R_1 - I_1I_3R_1 \\ P_{inst,Share2} &= I_2V_2 = V_{dd}I_2 - I_1I_2R_1 - I_2^2R_1 - I_2I_3R_1 - I_2^2R_2 - I_2I_3R_2 \\ P_{inst,Share3} &= I_3V_3 = V_{dd}I_3 - I_1I_3R_1 - I_2I_3R_1 - I_3^2R_1 - I_2I_3R_2 - I_3^2R_2 - I_3^2R_3\,. \end{aligned}$$

The power consumption of any one share thus theoretically depends on all shares and hence the masking scheme is not secure in the presence of IR drop.

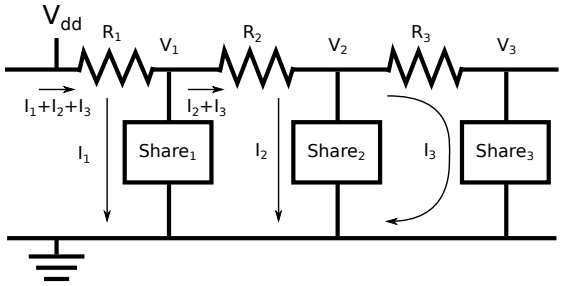

Fig. 4: Power supply noise or IR drop couples shares.

4 Coupling in Threshold Implementations of KATAN-32

We now investigate what effect coupling has on the first-order side-channel leakage of the KATAN-32 Threshold Implementation with three shares. In a first experiment, we measure the side-channel resistance of a regular Threshold Implementation of KATAN-32, for which we followed the design rules mentioned in the literature. In a second experiment, we show that placement has an influence on the leakage of the same (netlist-wise) KATAN-32 TI. Before we describe the actual experiments, we explain three constraints we use to guide the synthesis, map and place and route tools. Their full description and application are documented in [42].

Xilinx Constraints.

Keep Hierarchy. "Keep Hierarchy" is a synthesis and implementation constraint and is commonly used in papers about Threshold Implementations [5, 6, 28, 32]. HDL designs are generally a collection of hierarchical modules and submodules. In masked implementations the constraint is used to avoid optimizations over share boundaries, as its effect preserves the hierarchy throughout the implementation process and avoids the flattening of the design. In a masking context, the option is set globally as a synthesis option. Three values can be set for this option: true, soft and false. True preserves the design hierarchy throughout both synthesis and implementation, soft keeps the hierarchy during synthesis but not during the implementation phase while false allows all the submodules of the design to be merged within the top level module.

Keep. "Keep" is a constraint that influences the mapping phase of the implementation. It avoids nets from being merged into a single logic block. Taking the AND/XOR function $X \oplus YZ$ of KATAN-32 as example, the HDL code would explicitly declare an AND and an XOR operation while the mapper would merge both gates into a single LUT. This constraint is applied to signals in the HDL code.

Prohibit. "Prohibit" is a placement constraint that forbids the use of selected CLBs or Slices during PAR.

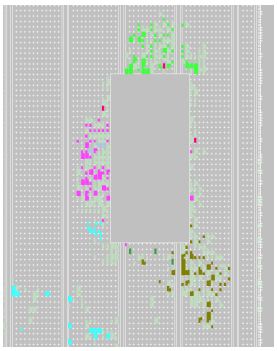

Fig. 5: The individual shares are placed far apart.

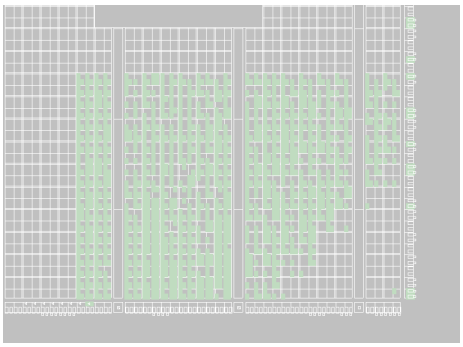

Fig. 6: All shares are placed in close proximity.

4.1 Secure Threshold Implementation of KATAN-32

To achieve a secure Threshold Implementation, we set the "Keep Hierarchy" synthesis option to true globally, as is done in related practical TIs [6, 28, 32].

Resulting from the hierarchy in both the synthesis and place and route phases, the individual shares are separately placed on the Virtex-II Pro floorplan and can be clearly distinguished. Figure 5 shows the separation of the three individual shares on the floorplan of the FPGA.

Evaluation We proceed with leakage assessment to evaluate whether or not the out-of-model leakage from coupling can be detected. To detect leakage in higher-order moments, we run the t-test on preprocessed traces. In all our evaluations we provide favorable measurement conditions for leakage detection: we use a very low noise platform (Virtex-II Pro FPGA on the SASEBO-G board [1]) clocked at a fixed frequency of 3.072 MHz while the instantaneous power consumption is measured with a Tektronix DPO 7254C oscilloscope at 1 GS/s.

Methodology. The evaluation methodology to check the masking scheme for leakage is as follows.

We first disable the masking scheme by turning off the masks. In that case, the first share equals the plaintext while the second and third shares are chosen to be zero. Leaks, i.e. t-values exceeding ± 4.5, are expected in the leakage detection test as the masking scheme is effectively not applied. In their presence, this experiment gives us confidence that the measurement setup is sound. We proceed by assigning the masks from a uniform random distribution, i.e. we activate the masking scheme, and repeat the leakage detection test. Any decrease of leakages is accredited exclusively due to a proper masking scheme. If leaks are detected, the implementation of the masking scheme is concluded to be erroneous.

Masks Off. The result of the leakage detection test with the masks turned off is shown in Figure 7. As expected, the t-value threshold of ± 4.5 is exceeded meaning the design with disabled masks leaks with 20k traces.

Masks On. Turning the masks on results in the first- and second-order leakage detection tests in respectively the middle and bottom graphs shown in Figure 7. The expected second-order leaks are present and suggest that we have enough measurements to be able to detect leakage in lower-order moments, if any would be present.

The dashed line in Figure 9 shows the evolution of the point of maximum first-order leakage in function of the number of traces in increments of 1M. The maximum of the absolute t-value fluctuates around the threshold but no steady increase in the maximum value is recognizable. We therefore conclude that no out-of-model leakage is observable with 100M traces.

4.2 Leaking Threshold Implementation of KATAN-32

To investigate the effect of the placement and its inducing coupling, we first convert the NGC netlist back to an HDL file. Since the netlist is only produced after the synthesis step, and therefore is influenced by the "Keep Hierarchy" constraint, the resulting HDL file consists of Xilinx specific primitives grouped into

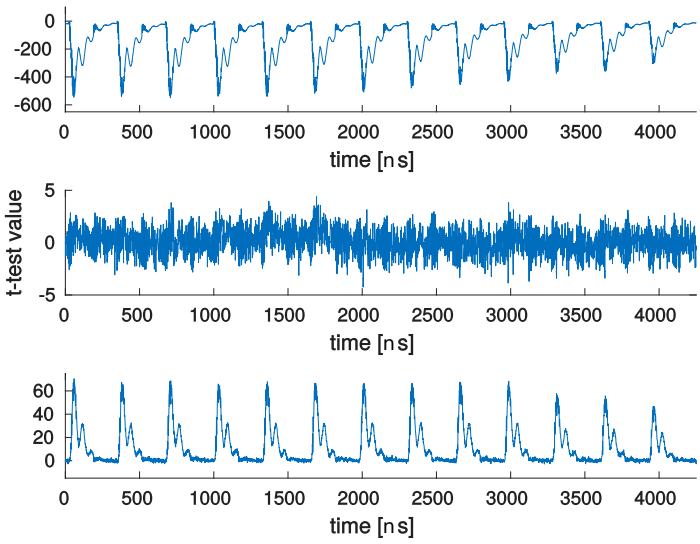

Fig. 7: Leakage detection test of a secure KATAN-32 TI, 20k traces masks off (top), 100M traces masks on 1^{st}-order (middle), 100M traces masks on 2^{nd}-order (bottom).

separate modules that reflect the hierarchical structure of the secure KATAN-32 TI. By merging the resulting HDL modules and assigning the "Keep" constraint to all signals, we preserve the integrity of the secure implementation while dropping the placement constraints originating from the "Keep Hierarchy" constraint. We proceed by synthesizing the HDL file with "Keep Hierarchy" set to false and force the placement of the components to the lower right corner of the FPGA floorplan using the "Prohibit" constraints. Figure 6 shows the floorplan of the FPGA. The three individual shares are now placed in close proximity.

Evaluation We follow the same pattern for leakage detection tests.

Masks Off. The result of the leakage detection test with the masks turned off is shown in Figure 8. Since the masks of the Threshold Implementation are set to zero, the t-value threshold of ± 4.5 is exceeded with 20k traces.

Masks On. The middle and bottom graphs in Figure 8 show the result of the first- and second-order leakage detection tests with 100M traces respectively. Small, periodic first-order leaks are visible and indicate the presence of out-of-model leakage .

The solid line in Figure 9 shows the evolution of the point of maximum first-order leakage for the leaking KATAN-32 TI. Unlike the uncertain fluctuation around the ± 4.5 threshold for the secure KATAN-32 TI, we now see a steady increase in the maximum of the absolute t-value. We conclude that out-of-model leakage , albeit small, is observable.

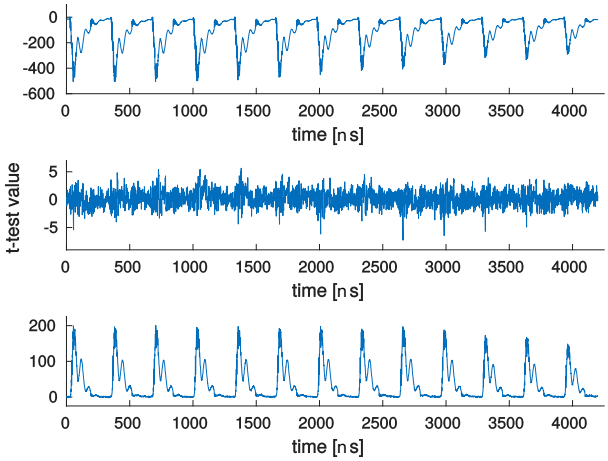

Fig. 8: Leakage detection test of an insecure KATAN-32 TI, 20k traces masks off (top), 100M traces masks on 1^{st}-order (middle), 100M traces masks on 2^{nd}-order (bottom).

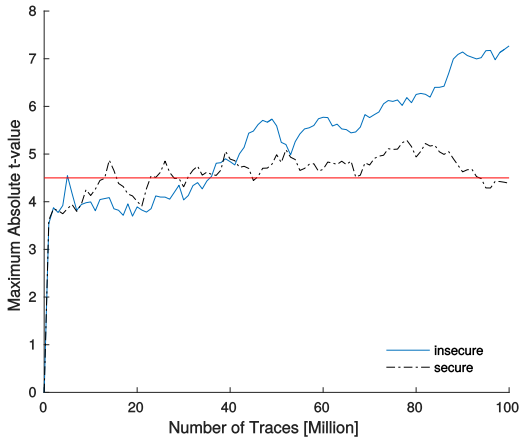

Fig. 9: Evolution of the points of maximum leakage with increasing number of traces for the secure and insecure KATAN-32 TI.

5 Discussion

5.1 A Note on "Keep Hierarchy".

In the majority of the Threshold Implementations literature, the "Keep Hierarchy" constraint is attributed with the function to keep the synthesis phase from optimizing over share boundaries. While this explanation is correct and different shares are indeed prevented from being merged in the same LUT, "Keep Hierarchy" also serves the purpose of not packing different shares in the same FPGA slice, which is shown to cause observable leakage in this work. When translating the "Keep Hierarchy" option to ASIC toolchains in order to avoid optimizations between shares, a common approach is to use "Compile Ultra" on different submodules and compiling the resulting netlists using the regular "Compile" process. While this effectively avoids optimizations across the boundaries of the shares, care might still be required to avoid standard cells belonging to different shares to be placed in the vicinity of each other, and routed wires of a share to be routed next to wires of other shares.

5.2 A Note on the Measurement Platform.

Our measurement setup is a low-noise platform based on a Virtex-II Pro FPGA. The 90nm technology node it uses delivers clean power traces with rather large amplitudes. As the out-of-model leakage effects we observed might not be as prominent with a 90nm technology as with smaller nodes, other side-channel evaluation boards will need evaluation. As crosstalk and IR drop are known to become more prominent with smaller technology nodes, the 65nm technology of the Virtex-5 on the Sasebo-GII platform [2], the 45nm technology of the Spartan-6 on the Sakura-G board [20] and the 28nm technology of the Kintex-7 on the Sakura-X platform form interesting targets for further investigation.

5.3 Conclusion

In this paper, we checked if coupling may be an issue in masking schemes. By using Threshold Implementations, we made sure the leakage we induced in our experiments originates from coupling, as the effects of glitches are ruled out. We achieve a secure KATAN-32 TI using the state-of-the-art "Keep Hierarchy" implementation technique and show its security using state-of-the-art leakage detection methods. Afterwards, we induced out-of-model leakage by placing the gates and registers of the secure design in close proximity, as would be done in a real-world design. The leakage detection shows this new design to leak and leads us to the following conclusion. Leakage from coupling can be induced deliberately or "by accident" in masking schemes by placing shares in the vicinity of each other. As is shown from the related TI FPGA implementations using the "Keep Hierarchy" option that pass the leakage detection test [5, 11, 39], this does not necessarily happen as it has not been observed before. Since this problem can be caused on an FPGA however, we believe that careful examination of other

environments is required when shares of a masking scheme might be densely packed, e.g. in cryptographic ASIC implementations. While the number of traces required for the leakage to be noticeable is high for our 90nm platform, smaller process technologies are known to be more susceptible to crosstalk and IR drop coupling [35] and can lead to more leakage and hence possibly insecure designs.

Acknowledgments

This work was supported in part by NIST with the research grant 60NANB15D346. In addition, this work was partially supported by the Research Council KU Leuven, OT/13/071 and by the Flemish Government through FWO project Cryptography secured against side-channel attacks by tailored implementations enabled by future technologies (G0842.13). Begül Bilgin and Benedikt Gierlichs are Postdoctoral Fellows of the Fund for Scientific Research - Flanders (FWO). Thomas De Cnudde is funded by a research grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

1. Research Center for Information Security, National Institute of Advanced Industrial Science and Technology, Side-channel Attack Standard Evaluation Board SASEBO-G Specification. `http://satoh.cs.uec.ac.jp/SASEBO/en/board/sasebo-g.html`
2. Research Center for Information Security, National Institute of Advanced Industrial Science and Technology, Side-channel Attack Standard Evaluation Board SASEBO-GII Specification. `http://www.rcis.aist.go.jp/special/SASEBO/SASEBO-GII-en.html`
3. Balasch, J., Gierlichs, B., Grosso, V., Reparaz, O., Standaert, F.: On the cost of lazy engineering for masked software implementations. In: CARDIS. Lecture Notes in Computer Science, vol. 8968, pp. 64–81. Springer (2014)
4. Bhooshan, R., Rao, B.P.: Optimum IR drop models for estimation of metal resource requirements for power distribution network. In: VLSI-SoC. pp. 292–295. IEEE (2007)
5. Bilgin, B., Gierlichs, B., Nikova, S., Nikov, V., Rijmen, V.: Higher-order threshold implementations. In: ASIACRYPT (2). Lecture Notes in Computer Science, vol. 8874, pp. 326–343. Springer (2014)
6. Bilgin, B., Gierlichs, B., Nikova, S., Nikov, V., Rijmen, V.: A more efficient AES threshold implementation. In: AFRICACRYPT. Lecture Notes in Computer Science, vol. 8469, pp. 267–284. Springer (2014)
7. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: CHES. Lecture Notes in Computer Science, vol. 3156, pp. 16–29. Springer (2004)
8. Cannière, C.D., Dunkelman, O., Knezevic, M.: KATAN and KTANTAN - A family of small and efficient hardware-oriented block ciphers. In: CHES. Lecture Notes in Computer Science, vol. 5747, pp. 272–288. Springer (2009)

9. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: CRYPTO. Lecture Notes in Computer Science, vol. 1666, pp. 398–412. Springer (1999)
10. Chen, Z., Haider, S., Schaumont, P.: Side-channel leakage in masked circuits caused by higher-order circuit effects. In: ISA. Lecture Notes in Computer Science, vol. 5576, pp. 327–336. Springer (2009)
11. Cnudde, T.D., Bilgin, B., Reparaz, O., Nikov, V., Nikova, S.: Higher-order threshold implementation of the AES s-box. In: CARDIS. Lecture Notes in Computer Science, vol. 9514, pp. 259–272. Springer (2015)
12. Cooper, J., DeMulder, E., Goodwill, G., Jaffe, J., Kenworthy, G., Rohatgi, P.: Test Vector Leakage Assessment (TVLA) Methodology in Practice. International Cryptographic Module Conference (2013), http://icmc-2013.org/wp/wp-content/uploads/2013/09/goodwillkenworthtestvector.pdf
13. Duan, C., LaMeres, B.J., Khatri, S.P.: On and off-chip crosstalk avoidance in VLSI design. Springer
14. Duc, A., Faust, S., Standaert, F.: Making masking security proofs concrete - or how to evaluate the security of any leaking device. In: EUROCRYPT (1). Lecture Notes in Computer Science, vol. 9056, pp. 401–429. Springer (2015)
15. Dyrkolbotn, G.O., Wold, K., Snekkenes, E.: Security implications of crosstalk in switching CMOS gates. In: ISC. Lecture Notes in Computer Science, vol. 6531, pp. 269–275. Springer (2010)
16. Dyrkolbotn, G.O., Wold, K., Snekkenes, E.: Layout dependent phenomena A new side-channel power model. JCP 7(4), 827–837 (2012)
17. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: CHES. Lecture Notes in Computer Science, vol. 2162, pp. 251–261. Springer (2001)
18. Goodwill, G., Jun, B., Jaffe, J., Rohatgi, P.: A Testing Methodology for Side-Channel Resistance Validation. NIST non-invasive attack testing workshop (2011), http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf
19. Goubin, L., Patarin, J.: DES and differential power analysis (the "duplication" method). In: CHES. Lecture Notes in Computer Science, vol. 1717, pp. 158–172. Springer (1999)
20. Guntur, H., Ishii, J., Satoh, A.: Side-channel attack user reference architecture board sakura-g. In: 2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE). pp. 271–274 (Oct 2014)
21. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: CRYPTO. Lecture Notes in Computer Science, vol. 2729, pp. 463–481. Springer (2003)
22. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: CRYPTO. Lecture Notes in Computer Science, vol. 1109, pp. 104–113. Springer (1996)
23. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: CRYPTO. Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer (1999)
24. Mangard, S., Oswald, E., Popp, T.: Power analysis attacks - revealing the secrets of smart cards. Springer (2007)
25. Mangard, S., Schramm, K.: Pinpointing the side-channel leakage of masked AES hardware implementations. In: CHES. Lecture Notes in Computer Science, vol. 4249, pp. 76–90. Springer (2006)
26. Moll, F., Roca, M., Isern, E.: Analysis of dissipation energy of switching digital CMOS gates with coupled outputs. Microelectronics Journal 34(9), 833–842 (2003)

27. Moradi, A.: Side-channel leakage through static power - should we care about in practice? In: CHES. Lecture Notes in Computer Science, vol. 8731, pp. 562–579. Springer (2014)
28. Moradi, A., Poschmann, A., Ling, S., Paar, C., Wang, H.: Pushing the limits: A very compact and a threshold implementation of AES. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 6632, pp. 69–88. Springer (2011)
29. Nikova, S., Rechberger, C., Rijmen, V.: Threshold implementations against side-channel attacks and glitches. In: ICICS. Lecture Notes in Computer Science, vol. 4307, pp. 529–545. Springer (2006)
30. Nikova, S., Rijmen, V., Schläffer, M.: Secure hardware implementation of non-linear functions in the presence of glitches. In: ICISC. Lecture Notes in Computer Science, vol. 5461, pp. 218–234. Springer (2008)
31. Nikova, S., Rijmen, V., Schläffer, M.: Secure hardware implementation of nonlinear functions in the presence of glitches. J. Cryptology 24(2), 292–321 (2011)
32. Poschmann, A., Moradi, A., Khoo, K., Lim, C., Wang, H., Ling, S.: Side-channel resistant crypto for less than 2, 300 GE. J. Cryptology 24(2), 322–345 (2011)
33. Prouff, E., Roche, T.: Higher-Order Glitches Free Implementation of the AES Using Secure Multi-party Computation Protocols. In: CHES. Lecture Notes in Computer Science, vol. 6917, pp. 63–78. Springer (2011)
34. Quisquater, J., Samyde, D.: Electromagnetic analysis (EMA): measures and counter-measures for smart cards. In: E-smart. Lecture Notes in Computer Science, vol. 2140, pp. 200–210. Springer (2001)
35. Rabaey, J.M.: Digital Integrated Circuits: A Design Perspective. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1996)
36. Schmidt, J., Plos, T., Kirschbaum, M., Hutter, M., Medwed, M., Herbst, C.: Side-channel leakage across borders. In: CARDIS. Lecture Notes in Computer Science, vol. 6035, pp. 36–48. Springer (2010)
37. Schneider, T., Moradi, A.: Leakage assessment methodology - A clear roadmap for side-channel evaluations. In: CHES. Lecture Notes in Computer Science, vol. 9293, pp. 495–513. Springer (2015)
38. Schneider, T., Moradi, A.: Leakage assessment methodology - extended version. J. Cryptographic Engineering 6(2), 85–99 (2016)
39. Schneider, T., Moradi, A., Güneysu, T.: Arithmetic addition over boolean masking - towards first- and second-order resistance in hardware. In: ACNS. Lecture Notes in Computer Science, vol. 9092, pp. 559–578. Springer (2015)
40. Trichina, E., Korkishko, T., Lee, K.: Small size, low power, side channel-immune AES coprocessor: Design and synthesis results. In: AES Conference. Lecture Notes in Computer Science, vol. 3373, pp. 113–127. Springer (2004)
41. Wild, A., Moradi, A., Güneysu, T.: Evaluating the duplication of dual-rail precharge logics on fpgas. In: COSADE. Lecture Notes in Computer Science, vol. 9064, pp. 81–94. Springer (2015)
42. Xilinx: Constraints guide 10.1, `http://www.xilinx.com/itp/xilinx10/books/docs/cgd/cgd.pdf`
43. Xilinx: Virtex-ii pro and virtex-ii pro x platform fpgas: Complete data sheet, `http://www.xilinx.com/support/documentation/data_sheets/ds083.pdf`
44. Zussa, L., Exurville, I., Dutertre, J., Rigaud, J., Robisson, B., Tria, A., Clédière, J.: Evidence of an information leakage between logically independent blocks. In: CS2@HiPEAC. pp. 25–30. ACM (2015)