Kalpa Publications in Computing

Volume 7, 2018, Pages 16–31

PROOFS 2018. 7th International Workshop
on Security Proofs for Embedded Systems

kalpa Computing

Towards Efficient and Automated Side Channel Evaluations at Design Time

Danilo Šijačić[1], Josep Balasch[1], Bohan Yang[1],
Santosh Ghosh[2], and Ingrid Verbauwhede[1]

[1] imec-COSIC, KU Leuven, Leuven, Belgium
name.surname@esat.kuleuven.be
[2] Intel Labs, Intel Corporation, Hillsboro, OR, USA
santosh.ghosh@intel.com

Abstract

Models and tools developed by the semiconductor community have matured over decades of use. As a result, hardware simulations can yield highly accurate and easily automated pre-silicon estimates for e.g. timing and area figures. In this work we design, implement, and evaluate CASCADE, a framework that combines a largely automated full-stack standard-cell design flow with the state of the art techniques for side channel analysis. We show how it can be used to efficiently evaluate side channel leakage prior to chip manufacturing. Moreover, it is independent of the underlying countermeasure and it can be applied starting from the earliest stages of the design flow. Additionally, we provide experimental validation through assessment of the side channel security of representative cryptographic circuits. We discuss aspects related to the performance, scalability, and utility to the designers. In particular, we show that CASCADE can evaluate information leakage with 1 million simulated traces in less than 4 hours using a single desktop workstation, for a design larger than 100kGE.

1 Introduction

Side Channel Analysis (SCA), introduced by Kocher et al. [15, 16], is acknowledged as a major threat to cryptographic implementations. Unlike conventional cryptanalysis techniques that stem from mathematics, SCA leverages information that leaks through inherent physical channels. These physical magnitudes carry within information about the values and operations internally processed by a circuit, including cryptographic keys. The most prominent exploitable physical side channels include timing [15], power consumption [16], and electromagnetic emissions [8]. The seminal Simple Power Analysis (SPA) and Differential Power Analysis (DPA) [16] attacks were soon followed by techniques such as Correlation Power Analysis (CPA) [3], or Mutual Information Analysis (MIA) [10]. On the other side, multiple countermeasure schemes have also emerged. Masking [4, 11], is a well-studied mitigation strategy based on randomizing the processing of sensitive variables during a cryptographic execution. Recent examples of masking schemes tailored for hardware include Threshold Implementations (TI) [21] and

Domain-Oriented Masking (DOM) [18]. Alternatively, secure logic styles can be used to mitigate SCA. These circuit-level hiding techniques, such as Wave Dynamic Differential Logic (WDDL) [29], aim to make the power consumption independent of the data being processed.

Assessing the security of SCA countermeasures is ultimately done at silicon level. The prevalent methodology in academic works is to synthesize and implement the designs on FPGA platforms mounted on dedicated evaluation boards. However, FPGA implementations can only be computational equivalents of ASICs. The fundamentally different structure of FPGA configurable logic blocks and ASIC gates can make such evaluations incomplete. Common approaches consist in exposing a design to batteries of known attacks and checking whether its security claims hold. The number of measurements for which the key can be recovered, often referred to as Measurement to Disclosure (MtD), is used as the quality metric. This approach is necessary, to empirically check whether design assumptions hold in practice and to verify that no errors are introduced from specification to implementation. It is also costly both in time and resources. Moreover, there is no guarantee that the security of a design would hold against a new type of attack. Test Vector Leakage Assessment (TVLA) [6] presents an appealing alternative. It is a generic method that contests the presence of leakage in different statistical moments.

Manufacturing secure devices remains costly and time consuming, requiring high degree of expertise. SCA vulnerabilities discovered by millions of post-silicon measurements cause major set backs that may require complete redesign. In this context, simulations rise as an attractive alternative to assess the SCA security at design time. They have the potential to capture information leakage already at pre-layout stages. Simulation techniques for typical hardware design constraints are long-studied and well integrated into Electronic Design Automation (EDA) tools. As a result, they can provide remarkably accurate *area*, *delay*, and *power* estimates even in the earliest design stages. In this work we combine existing models, EDA tools, and SCA assessment techniques to create a comprehensive, generic, and extensible framework for side channel analysis at design time. We show how it can be used efficiently to provide feedback to designers about the side channel security of a circuit. We approach the problem from a hardware designer's perspective, in a manner compliant to widely spread standard-cell design flow. We focus on power consumption waveform in time as the preferred side-channel. We call samples of this waveform the *instantaneous* power consumption (IPC).

The topic of IPC simulation in the context of SCA has been previously addressed in the literature. The generation and analysis of IPC estimates is in fact the prevalent approach to evaluate secure logic styles. Tiri and Verbauwhede [30] target an AES core implemented in WDDL [29]; Kirschbaum and Popp [14] an 8-bit controller in Masked Dual-Rail Precharge Logic; Regazzoni et al. [23] instruction set extensions in MOS Current Mode Logic; Kamel et al. [13] an AES S-box in Dynamic and Differential Swing-Limited Logic; and Bhasin et al. [2] a PRESENT engine in WDDL. All these works employ existing EDA tools to generate multiple power estimates from the circuit under test, either via SPICE simulators [30, 23, 24, 13, 2] or through logic simulators [14]. Custom design flows [31], considerations [17], and models [1, 19, 7] are tailored for specific cases. It is understood from these works that different balances of the simulation accuracy/time trade-off influence the security assessment. Logic simulations can provide quick but rough information leakage estimates at early stages. Transistor-level simulations on the other hand achieve better accuracy at the cost of more computation time. The number of measurements required for an evaluation can range from thousands to millions, which may be prohibitive in certain cases.

Although the topic of SCA evaluations based on simulations has been investigated in earlier works, to the best of our knowledge it has not yet been made an integral part of the design

process. In this paper we address this in a wholesome and methodical manner, spanning over the entire design flow – from behavioral to layout stages. We tackle both practical aspects on the implementation and evaluation of cryptographic circuits. We also provide performance and scalability figures to show the practical viability of the approach. Our goal is to enable a methodology that allows circuit designers to assess the security of their implementations at different stages, similar to what is currently done for e.g. timing constraints. Our contributions in this work are placed along three lines.

Firstly, we design and implement a flexible framework to support SCA at design time. We build on decades of experience of the EDA industry by using commercial EDA tools. We enrich this set with optimized parsers and analysis tools written in C. Our framework strings them according to categorized sets of parameters, to allow high degree of automation of design and SCA assessment. Secondly, we apply our framework to a set of representative cryptographic circuits in order to validate its functionality, performance, and utility. Thirdly, we discuss the validated features and give an example of a real world application. In particular, we use the tool to easily detect a flaw in a recently proposed masked design of an AES S-Box.

2 Computer Aided SCA Design Environment

In this section we introduce the Computer Aided Side Channel Analysis Design Environment (CASCADE). We begin by delineating the rationale behind our approach. Next, we describe its main components and their interaction. Lastly, we present the simulation methodology and the included models for timing and power simulation.

2.1 Design Rationale

The goal of CASCADE is to incorporate SCA evaluations at design time into the standard cell (SC) design flows, as the most widespread for digital design. By doing so, we aim to combine knowledge of both EDA and SCA community to develop a tool easily applicable in practice.

We build our framework around commercial EDA tools and associated data formats. We adhere to the SC design flow by using EDA simulators to obtain IPC estimates starting at the earliest stages of design. In order to embed SCA evaluations in all of the SC design stages we design and implement additional software components that bridge the gap between EDA tools and SCA evaluations at design time. Firstly, there exists a gap in *timing* and *power* models used in EDA contexts. The former are primarily targeted for performance, while the latter are a concern for heat dissipation and battery life. Instead SCA evaluations depend on less researched models for IPC estimation. Secondly, there is a gap in the handling and interpretation of simulator outputs. SCA evaluations demand the processing of many measurements with different data. Enabling mechanisms to efficiently generate and cope with sheer volumes of data is of critical significance for practical deployment of this approach.

We argue the systematic use of simulations along the EDA flow can greatly decrease efforts of designers, while yielding more reliably secure designs prior to manufacturing. At design time it is easy to focus on critical hardware blocks, prior to evaluation of entire designs. We can treat effects of controllers, data path, and all added circuitry (e.g. clock buffers, power and ground routing) uniformly. The absence of noise and high levels of precision allows us intimate observation of the target circuit, unattainable using measuring equipment. Simulations also provide fully aligned traces, removing the need for pre-processing.

Unlike the FPGA evaluations, we rely on a one-to-one model of an ASIC circuit. We stress that models, as simplifications of physical phenomena, can never fully capture real behavior.

Hence, simulations are only as accurate as the models they use, and they can not account for artifacts of the manufacturing process. Therefore, we do not propose design time evaluations as a replacement for post-silicon measurements, but as a design technique aimed at shortening time to market and more reliably secure designs. In our view, the practical viability of SCA evaluations at design time is bound by 3 aspects of simulations. Evaluations must: be available as early in the design flow as possible, be fast and scalable in terms of circuit sizes, and guarantee a reasonable level of confidence in the security of the end device. In this work we focus on the first two aspects, both important for the adaptation of this method. In order to study the last key aspect, it is necessary to make comparisons against chip measurements for a number of different scenarios. We leave this for future work.

Information leakage estimates obtained at different abstraction layers need to be analyzed in order to assess the security of a circuit. The most straightforward approach to determine a circuit's resistance to SCA is by testing it against a range of known attacks and using MtD as a metric. One alternative approach is to use the information-theoretic metric proposed by Standaert et al. [27]. While both these approaches are certainly useful and possible to integrate in our setting, they can be computationally and memory intensive due to the large spectrum of existing attack vectors and/or need for estimating probability distributions. A more suitable approach for this setting, is the *leakage detection*. In particular, the Test Vector Leakage Assessment (TVLA) methodology presented in [6] uses the T-test distinguisher to detect statistical dependencies between sensitive data and side channel information contained in the IPC measurements. The test analyzes two sets of measurements partitioned according to sensitive information. A so-called t value is calculated by applying the Welch's t-test, to assess whether their means are different. Assume μ_i, s_i^2, and n_i to be sample mean, variance, and cardinality of set i, respectively, where $i \in \{1, 2\}$. Then, the t value is computed as:

$$t = \frac{\mu_2 - \mu_1}{\sqrt{\frac{s_2^2}{n_2} + \frac{s_1^2}{n_1}}}. \tag{1}$$

If the t value is outside the ± 4.5 range, the test rejects the null-hypothesis with confidence greater than 99.999% for large numbers of measurements, i.e. indicating that the mean of the sets at a particular sample is distinguishable and thus highlighting the existence of side-channel leakage. The IPC measurement corresponding to a single execution of the target algorithm is referred to as *power trace*. Each power trace is therefore a vector of power samples and the t-test has to be applied sample-wise. The obtained vector is often referred to as *t-trace*, or *differential trace*. The main advantages of TVLA are its fast computation time, low memory requirements and the possibility to test for leakages in higher-order statistical moments. Efficient computation strategies have been recently put forward in [26] and [25]. It is often used to locate potentially vulnerable samples within power traces, such as S-Box computations. Then attacks can be focused on these samples only, significantly decreasing computational cost of attacks. Similarly, based on the position of these samples in the simulated trace vulnerable parts can be pinpointed with a precision of the single gate.

2.2 Framework Description

CASCADE allows automated and efficient SCA evaluation during all stages of the SC design flow. While it is easily extensible, its current modules are depicted in Figure 1. CASCADE is available via a Command Line Interface (`CLI`). The Session Manager (`SM`) is the central part of the framework. Every time a new session is started, a set of `Parameters` are configured and stored within the `SM`. These are laid down in Table 1. The `SM` centrally manages all configuration

Table 1: Configuration parameters.

	Category	Examples
①	Simulation	*precision, duration, test bench type*
②	Design constraints	*critical path, process corners*
③	Resources	*library resources, transistor models*
④	Physical constraints	*placement and routing parameters*
⑤	Power	*model parameters*

parameters. It evaluates them and returns to different tools in the format suited for each tool. We opt for this centralization to ensure coherency between tools, thus avoiding time loss due to error prone manual handling. The library manager (`LM`) parses and handles SC library files. The rest of the framework consists of *handlers*, *generators*, *parsers* and *analyzers*.

Handlers wrap particular EDA tools, abstracting their functionality, vendor, and software version. Each handler can be modified, or new ones can be created, independently from the rest of the framework. This makes CASCADE easily adaptable to any changes in the underlying tools or the flow itself. Handlers facilitate a design or simulation stage in a streamlined and automated manner. They produce TCL scripts (e.g. `run.LS`, `par.LSIM`) used to drive the underlying tools. Depending on the point in the flow, TCL scripts are associated with categories of parameters. Any change in session parameters is automatically propagated to all points in the flow. The set of EDA tools we currently use is given in Table 2.

The traversal of design stages is depicted in Figure 2. The initial behavioral (`BEH`) stage includes design capture and functional simulation of a circuit description in e.g. Verilog. Logic functionality is synthesized (`SYN`) using generic logic gates. This functionality is then mapped to library cells of a concrete library, to form a gate-level netlist (`GLN`). Placed and Routed (`PAR`) design stage comes before the tape-out. CASCADE enables SCA evaluation at every stage of the design flow. Similarly to timing closure, proceeding to the next stage is allowed once security requirements are fulfilled for the current stage. We perform these simulations according to models described in Section 2.3.

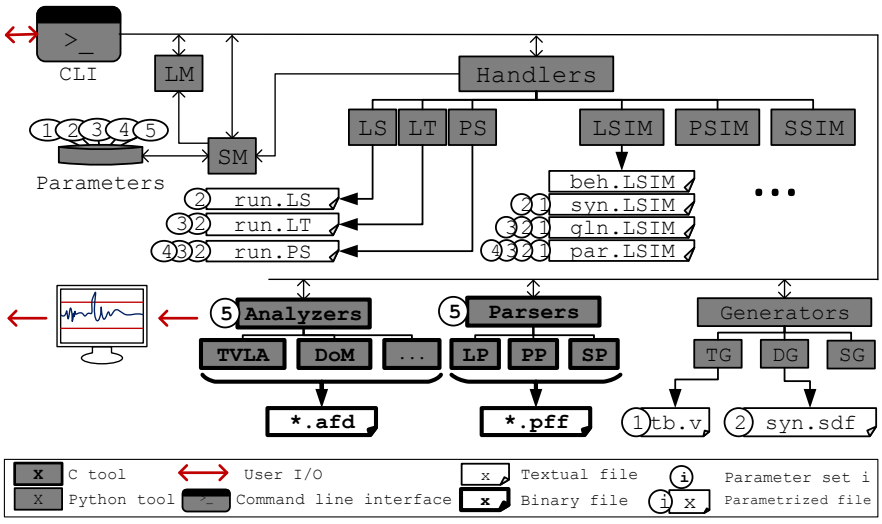

Figure 1: High level architecture of the framework.

Table 2: List of commercial EDA tools used.

Acronym	Function	Tool
`LS`	Logic synthesis	Synopsys Design Compiler
`LT`	Library translation	Synopsys Design Compiler
`PS`	Physical synthesis	Cadence Innovus
`LSIM`	Logic simulation	MentorGraphics QuestaSim
`PSIM`	Physical simulation	Synopsys PrimeTime with PX
`SSIM`	SPICE simulation	Synopsys HSPICE

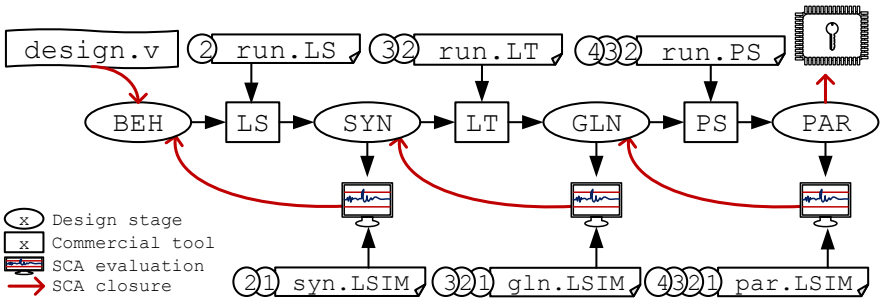

Figure 2: SC design flow stages using our toolchain.

Generators aid the automation. The test bench generator (`TG`) produces test benches based on Verilog code of the design (e.g. `tb.v`) and parameters obtained from the `SM`. It supports generation of several types of test benches. Functional test benches rely on user-defined test vectors for design and debugging, while the test benches for SCA evaluation are designed to exhaustively test all input transitions. When it is not feasible to exhaust all input transitions we may use random, or pseudo-random sequences as needed. A Delay Generator (`DG`) is additionally used to annotate generic netlists at `SYN` design stage (c.f. Δ-delay in Section 2.3). Delay annotations are stored in the Standard Delay Format (SDF), compliant with modern EDA tools. Lastly, the SPICE Generator (`SG`) includes a translator from Verilog to SPICE netlists, as well as an analog version of the test bench generator.

Similarly to data acquisition tools used in lab measurement setups, we design a set of parsers optimized to process and store IPC measurements in a SCA-friendly manner. All of our parsers are implemented in C. Regardless of the type of data we parse, Logic Parsers (`LP`), Power Parsers (`PP`), and SPICE Parsers (`SP`) output a Power Frame File (PFF); a custom binary format for simulated IPC traces. We refer to the part of simulation that corresponds to one power trace as simulation *frame*. PFF files can be generated from multiple sources and can represent simulated power that depends on different parameter values. We encode this information in the PFF file header, to allow uniform treatment of simulated traces.

We use analyzers to process PFF files. Each analyzer implements a specific SCA technique, e.g. TVLA or CPA. In particular for TVLA, we follow the roadmap of Schneider and Moradi [26]. The reason why we abstain from applying the faster leakage assessment of Reparaz et al. [25] is the prohibitive cost of storing histograms in our setting. We discuss this topic further in Section 4. The analysis consists of three steps that are performed on the fly for each frame. Firstly, a continuous power waveform is reconstructed from the frame data and the PFF header information. Secondly, an appropriate data set is updated with this waveform. PFF embeds the associated data values embedded in each frame, to allow on the fly partitioning. And thirdly,

we evaluate the context and write the output trace to Analyzed Frame Data (AFD) file; custom binary for convenient visual inspection. The latter step is mandatory after the final frame, but can be done periodically to observe the evolution of the SCA assessment.

2.3 Simulation Models and Methodology

Analog SPICE models, albeit the pinnacle of electronic modeling in terms of accuracy, feature exponential increases in run times with the increase in circuit sizes. We do support them in our framework, for they are useful as a reference for smaller validation circuits. Nevertheless, we rely on timing and power models, and simulators described bellow.

Timing parameters determine performance constraints, e.g. setup and hold times. Hence models for timing simulation (closure) are at the heart of EDA tools. SC libraries contain detailed information on how to extract timing parameters for `GLN` and `PAR` stages. In the `GLN` stage, interconnect delays are extracted from statistical wire load models embedded in the SC libraries. In the PAR stage, delays are extracted from the physical layout to include information about parasitic elements. These are detailed models for timing and power consumption, in the Open Liberty format compatible across EDA vendors. In particular, we rely on Composite Current Source (CCS) models, shown to be capable of producing power and timing estimates close to SPICE [20]. For each SC they capture a high level of detail, such as asymmetries of transitions caused by different input pins of a SC. As such, CCS models are an industry standard used for "golden" sign-off estimations. Prior to these stages, timings are roughly estimated using Δ-delay models. Since they leverage generic properties of CMOS gates, not particularities of a SC library, we use these models for the `SYN` stage. For each SC, regardless of its functionality, size, or input transition, the output is delayed by Δ such that: i) $\Delta = 0$, also known as zero-delay model; ii) constant $\Delta > 0$; iii) $\Delta = \delta(1 + (F-1)\theta)$ where F is the SC fanout and δ is the delay when $F = 1$, with a scaling factor θ. From empirical observations of several modern libraries, we assume that $0.05 \leq \theta \leq 0.20$. On the other hand, simulators such as PrimeTime PX from Synopsys can use CCS models to perform event driven IPC estimation. The static power of a CMOS gate depends on its state, and is significantly smaller than its dynamic power. CMOS predominantly consumes dynamic power while transitioning from one state to another. Roughly speaking, the value of the power contribution of a toggle depends on the structure (functionality) of the gate itself, the input transition time, and the output load. CCS models also capture subtle differences such as physical asymmetries of input pins and influence of short-circuit currents. For example, given a simple unloaded `XOR2` gate 8, in some cases drastically, different power rectangles can be observed for different transitions, as shown in Figure 3. A frame for each transition is separated using dashed vertical lines.

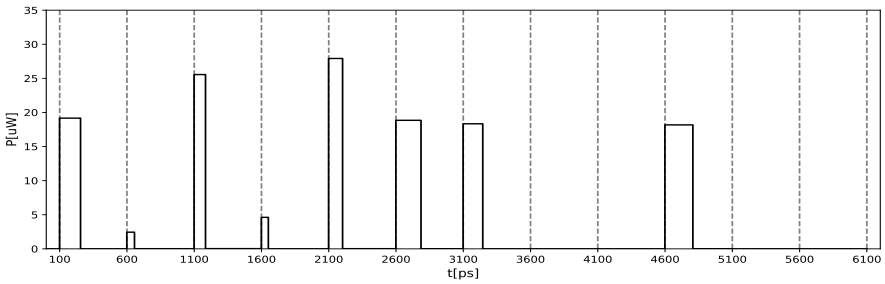

Figure 3: Dynamic power of different transitions in an XOR2 gate using CCS models.

On the other hand, the Hamming Distance model has repeatedly proven its worth in the SCA community. It is a simple model based on the predominance of dynamic power consumption. It is completely symmetrical in a sense that each transition always results in a Dirac-like pulse of unitary height. We refer to it as Marching-Sticks Model (MSM) for its graphical interpretation and to emphasize the difference from its theoretical use. In particular, the design of SCA countermeasures often relies on making the number of toggles independent of the processed data, by making arithmetic assumptions about the behavior of the circuit. The Hamming distance is used to describe this behavior. The MSM is a generic model hence it can be applied from `SYN` stage onwards. Unlike the plain toggle counting, in the MSM we try to address asymmetry between raising and falling edges such that $P_{0\to1} = 1$ and $P_{1\to0} = 1 - \alpha$, with parameter $-1.0 \leq \alpha \leq 1.0$. In a sense, we can relate MSM to CCS power models in the same manner as Δ-delay models relate to the timing ones. In this work we will focus on presenting MSM based estimation. They are built directly on top of logical simulations, which are the precursor for event-driven IPC estimations using CCS power models. We observe them as the common case in terms of performance and scalability, for one. For two, in the experiments we have conducted they have shown to be as successful as CCS models in detecting side channel leakage. In contrast to the traditional digital design we focus on the transitions rather than the steady states. Since we make no assumptions about the functionality of the target circuit, we can equally analyze implementations of masking schemes, SC-based secure logic styles, or any other block of digital hardware. In order to capture all possible transitions of a circuit with n input bits we need to simulate $2^{2n} - 2^n$ transitions. We call this type of simulation Exhaustive Dynamic Power Capturing (EDPC). We use a custom algorithm, to ensure that we traverse all transitions exactly once, without repetitions. The exponential complexity of EDPC makes it unusable for circuits with large number of input bits. We find EDPC feasible for circuits with up to 16 input bits. We use EPDC for rigorous evaluation of smaller, but SCA critical, blocks. For larger designs, we resort to generate inputs in a pseudo-random fashion. This is completely analogous to the acquisition approach followed in a lab environments, i.e. at post-silicon level. All simulations are driven by test benches output by the testbench generator (`TG`).

3 Framework Validation

In this section we present results of applying CASCADE to representative cryptographic circuits. We show how it can be applied to both masked designs instantiated to provide *first-order* security, i.e. devised to resist power analysis attacks that exploit information leakages in the first-order moment, as well as SC based secure logic styles. We use these circuits to validate the operation of our toolchain. And second, we can test the boundaries of our toolchain by evaluating multiple circuits with the same security guarantees. We use a 45nm open source SC library from NanGate.

3.1 Motivating Example

We use the first-order DOM-*indep* multiplier (`AND2` gate), a masking countermeasure from [12] depicted in Figure 4 (left most) as a motivating example. Input and output variables are split into 2 shares such that $a = a_1 \oplus a_2$, $b = b_1 \oplus b_2$ and $c = c_1 \oplus c_2 = AND(a, b)$. The design consumes one bit of randomness z per evaluation. A register stage is inserted in order to prevent leakage of sensitive information due to glitches.

Figure 4 (left) shows various MSM power profiles (averaged traces) based on different timing models. With a total of 5 input bits, EDPC consists of $2^{2\cdot5} - 2^5 = 992$ input transitions. In

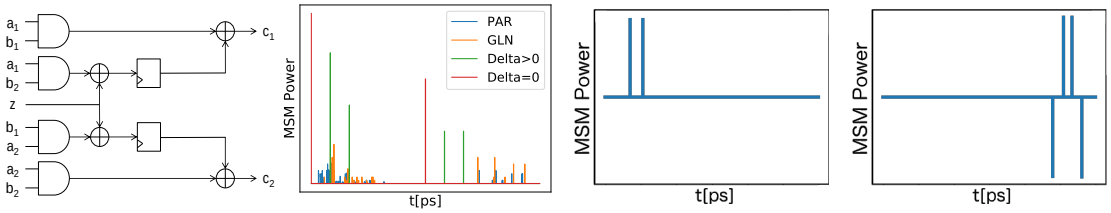

Figure 4: DOM-*indep* `AND2` gate design (left most) and overview of MSM power profiles (left). First-order differential traces for dependent inputs $a_1 = b_1$ (right) and for a fixed z (right most), using Δ-delay simulations.

this situation, a first order SCA estimation can be simply done by computing the difference of means of measurement sets, partitioned according to the value of sensitive variables. In what follows, the unshared output value c determines the splitting into sets. If the implementation is secure the differential has a constant zero value. This is indeed true if all 992 frames are used. For the purposes of validation, we induce a flaw in the design by violating the constituting conditions of the design. We break the independence of inputs condition by using only the frames where $a_1 = b_1$. Figure 4 (right) depicts the resulting information leakage, in the first cycle. We turn the masking off by fixing the value of z. Figure 4 (right most) depicts the resulting information leakage, now in the second cycle. We plot results obtained using Δ-delay simulations for simplicity. All findings hold across the other models we use. We see that the information leakage can be detected fairly early in the design flow. Also, the precision and discrete nature of models may allow us to pinpoint the source of leakage in the design.

3.2 Protected S-Boxes

S-Boxes are often the most SCA vulnerable parts of cryptographic algorithms. We show how our toolchain can be used regardless of the underlying countermeasure.

TI PRESENT S-Box. We target the first-order secure Threshold Implementation (TI) PRESENT S-Box by Poschmann et al. [22], depicted in Figure 5 (top left). The design is decomposed into two quadratic S-boxes F and G, which are split into three shares, per variable in accordance with the TI principles. The total number of inputs (resp. outputs) is thus 12 (4 sensitive bits masked with 3 shares), resulting in $2^{2 \cdot 12} - 2^{12} \approx 16$ *million* transitions long EDPC. Designed to provide only first-order security, leakage can easily be detected in the second-order moment, as depicted in Figure 5 (top right). Testing for first order leakage results in a constant zero value of the differential trace across all models used. We introduce a vulnerability, by using TI with 2 shares instead of 3 by fixing $x_3 = y_3 = w_3 = z_3 = 0$. Figure 5 (bottom left) clearly shows leakage obtained using the GLN simulation. Lastly, Figure 5 (bottom right) shows evolution of the absolute value of the t-trace peak over the increasing number of frames. More detailed CCS timing models detect leakage in the first 20000 frames, whereas generic Δ-delay models require several hundreds of thousands frames. We see that even the simplest $\Delta = 0$ simulations are capable of detecting leakage.

Boyar-Peralta AES S-Box. Ghoshal and De Cnudde [9] proposed a supposedly first-order secure implementation of Boyar-Peralta AES S-Box, designed to consume no randomness. Wegener and Moradi [32] used an elaborate FPGA setup and process 10 million to show that this

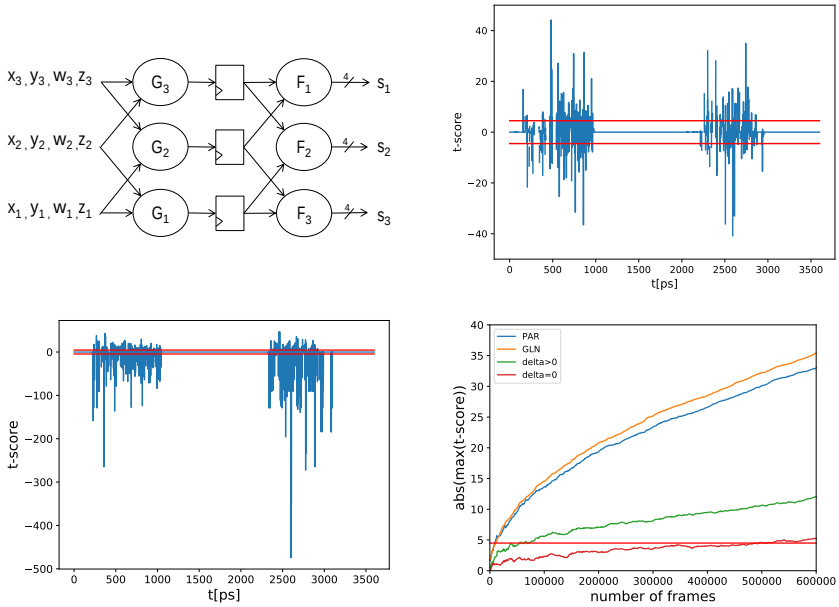

Figure 5: TI PRESENT S-Box (top left); second-order TVLA (top right) and first-order TVLA with 2 shares (bottom left). Evolutaion of the first-order t-value with 2 shares (bottom right).

design exhibits leakage due to a uniformity problem. Instead, we apply our toolchain to detect significant leakage with less than 400 thousand GLN frames, in less than 30 minutes.

WDDL PRESENT S-Layer. Instead of evaluating a single S-Box we implement the S-Layer of one round of PRESENT in WDDL. Since WDDL relies on symmetries in hardware, observing 16 4-bit S-Boxes in parallel is more realistic and captures the routing effects more promenently. Instead trying to run $2^{2\cdot 64} - 2^{64}$ EDPC transitions, we run fixed vs. random TVLA. High level architecture of one S-Box is depicted in Figure 6 (left). Differential pair of WDDL modules needs to be periodically precharged (`PC`) and evaluated. For example, `AND2` gate computes $a \cdot b = c$. WDDL version of this gate, `WDDL_AND2` gate computes $a_p \cdot b_p = c_p$ (positive end) and $a_n + b_n = c_n$ (negative end). Hence `WDDL_AND2` gate consists of one `AND2` gate and its complement `OR2` gate. In the precharge phase WDDL complementary inputs are set to zero, i.e. $a_p = 0, b_p = 0, a_n = 0, b_n = 0$. Next, in the evaluation phase WDDL complementary inputs are set to: $a_p = a$, $b_p = b$, $a_n = \bar{a}$, $b_n = \bar{b}$. This guaranties that the sum of toggles in the differential pair is constant. If the underlying `AND2` and `OR2` gates are completely symmetrical in terms of propagation delay and power consumption, WDDL yields a power consumption independent of the data it processes. In practice this can not be fully attained. Still, if the assymetries remain small enough WDDL circuits can be secure for a very large number of traces. We implement the precharge control (dashed lines in the figure) as a part of the test bench and focus on the worst case evaluation of the registered S-Layer.

Figure 6 (right) shows the evolution of the absolute peaks of the t-trace up to 10 million traces. Detected amount of leakage in the `GLN` is caused by imbalances in `AND2` and `OR2` gates of the target SC library. We confirm this by evaluating Δ-delay MSM simulations, which yield a constant zero value of the t-trace. Nevertheless, GLN leakage is bellow the threshold and

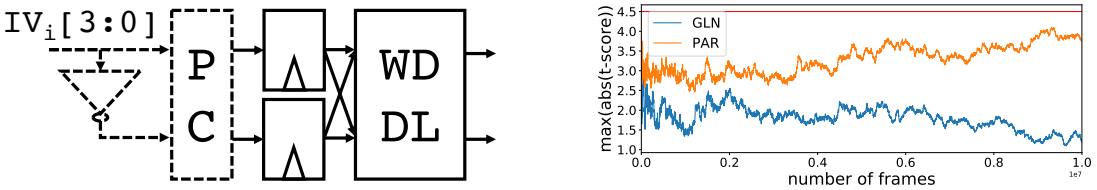

Figure 6: WDDL PRESENT S-layer architecture (left), evolution of absolute peaks of the t-value over 10 million frames (right).

falling. In contrast, t-trace evolution of MSM simulations at the `PAR` exhibit a growing trend. As expected, assymetries introduced by physical placement and routing are increasing the differences in the IPC depending on the processed data. These asymmetries can be fixed using a dedicated routing technique by Tiri and Verbauwhede [28]. In practice, breaking this design would require several tens of millions of traces, which greatly surpasses typical device lifetime. Also, this level of precision is physically unatainable with today's measuring equipment. Still, we can not claim that this will result in a secure chip, for manufacturing innevitably introduces variations which are not accounted for using these models. Instead we show how CASCADE can be used to preliminary assess the SCA leakage. x

4 Discussion

4.1 Utility to the Designer

Designed in compliance with commercial EDA tools and standards, CASCADE can be easily included in a designer's toolbox. It allows early SCA evaluations of building blocks prior to integration. Designers may pinpoint bugs and flaws, and proceed to fix them before moving on to the next stage. Regardless of the target countermeasure, as long as the design is implemented using SC libraries, this approach can be applied. A popular alternative to simulation is FPGA prototyping. Its advantage is that measurements are directly obtained from a chip. Unfortunately, the internal structure of FPGA is radically different from the actual layout of the target ASIC. Specifics of the FPGA internals are proprietary to its vendor. This hinders the identification and/or fixing of issues identified in a security analysis. An example is given by De Cnudde et al. [5], who investigate the impact of coupling effects on protected designs running on FPGA platforms.

In contrast, simulation-based evaluations allow designers to work with the direct model of the target ASIC. Thus they have the potential to overcome these issues. Here, two things need to be considered. On the one hand, we use 1ps time resolution combined with single precision power values, in a noiseless environment. They can therefore give a too clear view of the hardware by capturing effects that can not be observed in practice. This may lead to *false positive* assessments (i.e. the tools indicates leakage, but in practice the circuit can not be broken). Methods to overcome this may involve derating the simulated data or introducing noise, but this demands further investigation. On the other hand, simulations are only as accurate as the model they employ. Hence they can neglect some physical effects of the circuit, leading to *false negative* assessments (i.e. the tool does not indicate leakage, but in practice physical phenomena outside of the model lead to the broken device). Studying the accuracy of power models is beyond the scope of this work, but it is clear that there exists a gap in

comparison to the measurements on an ASIC target. We note the same gap exists between FPGA and ASIC implementations as well. To the best of our knowledge this gap is not yet quantified.

4.2 Performance

With 350 GE in size, the TI PRESENT S-Box is a small, but critical, design block. Still, its EDPC sequence requires over 16 million frames, which makes this effort non-trivial, and suitable for comparing timing models given a fixed circuit. The run times of MSM simulations, shown in Table 3, differ slightly depending on the timing model used. We have observed identical trends in all the target circuits.

Table 3: Performance of EDPC of TI PRESENT S-Box.

Model	$\Delta = 0$[1]	$\Delta > 0$	$\Delta = f(F, \delta)$	GLN	PAR
`LSIM`[min]	4.85	7.15	7.30	11.38	9.25
`LP`[min]	1.65	1.98	2.13	3.10	3.15
`TVLA1`[min]	0.07	7.95	8.05	8.82	9.47
`TVLA2`[min]	0.13	16.58	20.51	21.45	22.57

[1] Clock period is 10ps, as opposed to 1800ps in other cases.

For all different design stages we perform 3 steps: `LSIM` → `LP` → `TVLA`. `TVLA` is the analyzer that can perform TVLA on first-order (`TVLA1`) and second-order (`TVLA2`) statistical moments. In each case a total of $2^{24} - 2^{12}$ simulations are run and analyzed using a single CPU thread. The simulation resolution is set to 1ps, since this is the precision of library models. Running simulations at lower resolutions impacts only the size of output files, not the runtime of `LSIM`. Simulations times are mostly determined by the total number events. The more complex the model, the more different propagation delays result in more glitches and different toggling times. Since our parsers and analyzers are event driven, this monotonous dependency on the number of events prevails. In fact, this trend is broken only in case of GLN vs PAR simulation. PAR simulation produces more events ($\approx$ 1.3 billion) compared to GLN ($\approx$ 1.2 billion). Also, PAR and GLN netlists differ in only a single (clock buffer) gate inserted during physical synthesis. Without looking at the implementation of the simulator, we can not give a certain reason for this discrepancy. We do notice that the only significant difference is the way the extracted SDF data is presented. At the GLN stage, statistical wire load models are written to SDF as interconnect delays. At the PAR stage, wire delays are extracted from the layout and back annotated to the cell delays. Hence this subtle difference may lead to fewer instructions during simulation, causing faster runtime in the PAR stage. `TVLA` evaluations are performed on the fly using the approach of Schneider and Moradi [26]. Unlike the 8–12-bit resolution of modern oscilloscopes, simulations produce single precision floating point traces. Hence, following the approach of Reparaz et al. [25] would require storing 2^{64} histograms. Alternatively, simulated results can be quantized down to 8–12 bit range. We abstain from this for two reasons. High precision of simulation is one of its main advantages. To properly quantize values we must know global extrema of all traces. This goes against on-the-fly execution. This could arguably be solved by conservatively estimating global extrema, at a further loss of precision.

To test the scalability of our the approach we apply it to a fully unrolled implementation of AES-128 (AES-U). We use a placed and routed design of 127.18 kGE with extracted layout parasitics. Table 4 shows the average run times for processing 1 million PAR traces of AES-U along with the other circuits studied in this work. At this scale, the computational cost of the

AES-U simulations themselves becomes predominant. Simulations are done using sophisticated CCS models with a precision of 1ps, at the post-layout stage that includes extracted parasitic elements. With this level of detail they akin to the "golden sign-off" simulations for the timing closure. The size of the AES-U exceeds security-dedicated area budgets of many embedded devices. Still, a million traces can be simulated and processed in less than 32 hours, using a single thread of the i7-7700 workstation. Running simulations of the target circuit for different input stimuli can be computed in parallel, as we simply need to divide the sequence of frames into multiple batches. Experimenting with batch sizes between 10 thousands and a 100 thousands frames we did not notice any significant performance difference. Nevertheless, smaller batches equal smaller storage requirements and earlier evaluations. Each batch is processed in the same manner as a whole simulation would be, updating the analyzer's context (in this case `TVLA`) with new values. Furthermore, as all computations are performed on the fly there is no need for storing terabytes of simulated data dumped by `LSIM`. Hence, using the same 8-thread workstation 1 million traces can be simulated and analyzed in less than 4 hours.

Table 4: Simulation and analysis run times on a single thread of i7-7700 per 1 million frames.

Design	Area [kGE]	Frame [ps]	`LSIM` [h]	`LP` [h]	`TVLA1` [h]	`TVLA2` [h]
TI PRESENT S-Box	0.35	1800	0.02	< 0.01	0.01	0.02
WDDL PRESENT S-Layer	2.98	1800	0.14	0.06	0.01	0.02
BP AES S-Box	5.45	5000	0.41	0.07	0.06	0.18
AES-U	127.18	30000	25.81	5.75	0.16	0.26

5 Conclusions and Future Work

In this work we have design and implement CASCADE, a comprehensive framework for SCA evaluation at design time. CASCADE is built on the state of the art EDA tools and SCA evaluation methodologies, combining them in a methodical and automated manner. We show it can be applied in the early design stages regardless of the type of SCA countermeasure, as long as it uses standard cell design flow. We benchmark the performance of selected modules in our framework to show its suitability for testing realistic cryptographic designs, and argue its feasibility for real-world use even when relying on a single desktop workstation. As the future work we plan to compare simulated results with measurements from the corresponding chip. We aim to use these insights to calibrate our framework and to refine the models for more efficient and reliable SCA evaluation at design time. Lastly, we plan to release CASCADE in the form of open-source software, available to the research community.

6 Acknowledgments

This project has received partial funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 643161, and HECTOR grant agreement No. 644052; as well as from Intel Corporation.

References

[1] M. Aigner, S. Mangard, F. Menichelli, R. Menicocci, M. Olivieri, T. Popp, G. Scotti, and A. Trifiletti. Side channel analysis resistant design flow. In *2006 IEEE International Symposium on Circuits and Systems*, pages 4 pp.–2912, May 2006.

[2] Shivam Bhasin, Jean-Luc Danger, Tarik Graba, Yves Mathieu, Daisuke Fujimoto, and Makoto Nagata. Physical security evaluation at an early design-phase: A side-channel aware simulation methodology. In Christian Berger and Ina Schaefer, editors, *Engineering Simulations for Cyber-Physical Systems - ES4CPS 2014*, page 13. ACM, 2014.

[3] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 16–29. Springer, 2004.

[4] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, pages 398–412. Springer, 1999.

[5] Thomas De Cnudde, Begül Bilgin, Benedikt Gierlichs, Ventzislav Nikov, Svetla Nikova, and Vincent Rijmen. Does coupling affect the security of masked implementations? In Sylvain Guilley, editor, *Constructive Side-Channel Analysis and Secure Design -COSADE 2017*, volume 10348 of *LNCS*, pages 1–18. Springer, 2017.

[6] Jeremy Cooper, Elke DeMulder, Gilbert Goodwill, Joshua Jaffe, Gary Kenworthy, and Pankaj Rohatgi. Test Vector Leakage Assessment (TVLA) methodology in practice. International Cryptographic Module Conference, 2013.

[7] Daisuke Fujimoto, Makoto Nagata, Toshihiro Katashita, Akihiro T. Sasaki, Yohei Hori, and Akashi Satoh. A fast power current analysis methodology using capacitor charging model for side channel attack evaluation. In *Hardware-Oriented Security and Trust - HOST 2011*, pages 87–92. IEEE, 2011.

[8] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *LNCS*, pages 251–261. Springer, 2001.

[9] Ashrujit Ghoshal and Thomas De Cnudde. Several masked implementations of the boyar-peralta AES s-box. In *Progress in Cryptology - INDOCRYPT 2017 Chennai, India, December 10-13, 2017, Proceedings*, pages 384–402, 2017.

[10] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2008*, volume 5154 of *LNCS*, pages 426–442. Springer, 2008.

[11] Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES'99*, volume 1717 of *LNCS*, pages 158–172. Springer, 1999.

[12] Hannes Gross, Stefan Mangard, and Thomas Korak. Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order. Cryptology ePrint Archive, Report 2016/486, 2016. http://eprint.iacr.org/2016/486.

[13] Dina Kamel, Mathieu Renauld, Denis Flandre, and François-Xavier Standaert. Understanding the limitations and improving the relevance of SPICE simulations in side-channel security evaluations. *J. Cryptographic Engineering*, 4(3):187–195, 2014.

[14] Mario Kirschbaum and Thomas Popp. Evaluation of power estimation methods based on logic simulations. In Karl-Christian Posch and Johannes Wolkerstorfer, editors, *Austrochip 2007*, page 45–51. Verlag der Technischen Universität Graz, 2007.

[15] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96*, volume 1109 of *LNCS*, pages 104–113. Springer, 1996.

[16] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, pages 388–397. Springer, 1999.

[17] François Macé, François-Xavier Standaert, and Jean-Jacques Quisquater. Information theoretic evaluation of side-channel resistant logic styles. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *LNCS*, pages 427–442. Springer, 2007.

[18] Stefan Mangard and Kai Schramm. Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *LNCS*, pages 76–90. Springer, 2006.

[19] Amir Moradi, Mahmoud Salmasizadeh, Mohammad Taghi Manzuri Shalmani, and Thomas Eisenbarth. Vulnerability modeling of cryptographic hardware to power analysis attacks. *Integration, the VLSI Journal*, 42(4):468 – 478, 2009.

[20] T. E. Motassadeq. Ccs vs nldm comparison based on a complete automated correlation flow between primetime and hspice. In *2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*, pages 1–5, April 2011.

[21] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. In Pil Joong Lee and Jung Hee Cheon, editors, *Information Security and Cryptology - ICISC 2008*, volume 5461 of *LNCS*, pages 218–234. Springer, 2008.

[22] Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-channel resistant crypto for less than 2, 300 GE. *J. Cryptology*, 24(2):322–345, 2011.

[23] Francesco Regazzoni, Alessandro Cevrero, François-Xavier Standaert, Stéphane Badel, Theo Kluter, Philip Brisk, Yusuf Leblebici, and Paolo Ienne. A design flow and evaluation framework for dpa-resistant instruction set extensions. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *LNCS*, pages 205–219. Springer, 2009.

[24] Francesco Regazzoni, Thomas Eisenbarth, Axel Poschmann, Johann Großschädl, Frank K. Gürkaynak, Marco Macchetti, Zeynep Toprak Deniz, Laura Pozzi, Christof Paar, Yusuf Leblebici, and Paolo Ienne. Evaluating resistance of MCML technology to power analysis attacks using a simulation-based methodology. *Transactions on Computational Science IV, Special Issue on Security in Computing*, 4:230–243, 2009.

[25] Oscar Reparaz, Benedikt Gierlichs, and Ingrid Verbauwhede. Fast leakage assessment. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017*, volume 10529 of *LNCS*, pages 387–399. Springer, 2017.

[26] Tobias Schneider and Amir Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015*, volume 9293 of *LNCS*, pages 495–513. Springer, 2015.

[27] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 443–461. Springer, 2009.

[28] K. Tiri and I. Verbauwhede. A vlsi design flow for secure side-channel attack resistant ics. In *Design, Automation and Test in Europe*, pages 58–63 Vol. 3, March 2005.

[29] Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In *Design, Automation and Test in Europe - DATE 2004)*, pages 246–251. IEEE Computer Society, 2004.

[30] Kris Tiri and Ingrid Verbauwhede. Simulation models for side-channel information leaks. In William H. Joyner Jr., Grant Martin, and Andrew B. Kahng, editors, *Design Automation Conference - DAC 2005*, pages 228–233. ACM, 2005.

[31] Kris Tiri and Ingrid Verbauwhede. A digital design flow for secure integrated circuits. *IEEE*

Trans. on CAD of Integrated Circuits and Systems, 25(7):1197–1208, 2006.

[32] Felix Wegener and Amir Moradi. A first-order sca resistant aes without fresh randomness. Cryptology ePrint Archive, Report 2018/172, 2018.