Journal of Cryptographic Engineering
https://doi.org/10.1007/s13389-020-00233-8

REGULAR PAPER

Towards efficient and automated side-channel evaluations at design time

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

Received: 31 January 2019 / Accepted: 26 May 2020

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 h using a single desktop workstation, for a design larger than 100 kGE.

Keywords Side-channel analysis · ASIC · Hardware simulation · Design time methodology

1 Introduction

Side-channel analysis (SCA), introduced by Kocher et al. [19,20], 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 [19], power consumption [20], and electromagnetic emissions [12]. Seminal simple power analysis (SPA) and differential power analysis (DPA) [20] attacks were soon followed by techniques such as correlation power analysis (CPA) [6] or mutual information analysis (MIA) [14]. These attacks have been used to break security features of commercial devices [2,10,26]. In parallel with this, multiple countermeasure schemes have also emerged. Masking [7,15] is a well-studied technique based on randomizing the processing of sensitive variables during a cryptographic execution. A valuable property of masking schemes is that their security can be formally proved, as long as the leakage models and the underlying hardware assumptions hold. Recent examples of masking schemes tailored to hardware include threshold implementations (TIs) [25] and domain-oriented masking (DOM) [22]. Alternatively, secure logic styles can be used. These circuit-level hiding techniques, such as wave dynamic differential logic (WDDL) [34] and improved masked dual-rail pre-charge logic (iMDPL) [27], aim to make the power consumption independent of the data being processed. The prevalent

This work was supported in part by the European Commission through Horizon 2020 research and innovation programmes under the Marie Skłodowska-Curie Grant 643161 and Cathedral ERC Advanced Grant 695305; by the Research Council KU Leuven C1 on Security and Privacy for Cyber-Physical Systems and the Internet of Things with contract number C16/15/058; and by Intel Corporation.

✉ Danilo Šijačić
danilo.sijacic@esat.kuleuven.be

Josep Balasch
josep.balasch@esat.kuleuven.be

Bohan Yang
bohan.yang@esat.kuleuven.be

Santosh Ghosh
santosh.ghosh@intel.com

Ingrid Verbauwhede
ingrid.verbauwhede@esat.kuleuven.be

[1] imec-COSIC, KU Leuven, Belgium

[2] Intel Labs, Intel Corporation, Hillsboro, OR, USA

Published online: 22 June 2020

methodology for side-channel evaluations at design time relies on FPGA prototyping. The functionality of the desired ASIC circuit is mapped onto an FPGA platform, often equipped with dedicated circuitry to obtain power consumption measurements. The design is then subjected to batteries of attacks, in order to verify whether security assumptions hold and that no flaws were introduced during the implementation. The level of security is typically determined by the number of measurements required to recover the key, i.e., measurement to disclosure (MtD). In recent years, leakage detection testing strategies such as test vector leakage assessment (TVLA) [9] have gained popularity to assess the security of implementations against SCA. In contrast to actual attacks, TVLA decreases the evaluation costs by simply contesting the presence of information leakage in different statistical moments. A limitation of this approach is, however, that FPGA implementations can only be computational equivalents of ASICs. The fundamentally different structure of FPGA configurable logic blocks and ASIC gates can therefore make such evaluations incomplete. Formal verification methods such as [3,5] are emerging as valuable alternatives that do not require collections of measurements. Nevertheless, they operate on rather high levels of abstraction and are closely tied to certain types of countermeasures.

1.1 Motivation

Manufacturing side-channel secure devices is a costly and time-consuming process, requiring high degree of expertise. SCA vulnerabilities disclosed at post-silicon stages can cause major setbacks that may require a 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 in 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. This includes taking a closer look at the power models available in the EDA industry and making them available to the SCA community. We focus on power consumption waveform in time as the preferred side channel. We refer to this waveform as the instantaneous power consumption.

1.2 Related work

Simulating instantaneous power is the prevalent approach for evaluating the SCA security of secure logic styles. Tiri and Verbauwhede [35] target an AES core implemented in WDDL [34]; Kirschbaum and Popp [18] an 8-bit controller in masked dual-rail pre-charge logic (MDPL); Regazzoni et al. [29] instruction set extensions in MOS current mode logic; Kamel et al. [17] an AES *S*-box in dynamic and differential swing-limited logic; and Bhasin et al. [4] 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 [4,17,29,30,35] or via logic simulators [18]. Custom design flows [36,37], considerations [21], and models [1,11,23] are tailored for specific cases. It is understood from these works that different balances of the simulation accuracy versus 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.

1.3 Our contributions

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 four different lines.

Firstly, we design and implement a flexible framework to support SCA at design time. We start from decades of experience 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 take a close look at the physical gate-level models used by the EDA industry and how to use them to provide SCA assessments. Thirdly, we apply our framework to a set of representative cryptographic circuits in order to validate its functionality, performance, and utility. Lastly, we discuss the validated features and give an example of a real-world application. In particular, demonstrate a flaw in a recently proposed masked design of an AES S-box.

1.4 Paper organization

The rest of the paper is organized as follows. In Sect. 2, we present our framework and delineate the tools, models, and methodology used. In Sect. 3, we give experimental results to illustrate and validate the functioning of our framework. In Sect. 4, we benchmark and discuss our framework using several circuits. Lastly, we present our conclusions and delineate directions of our research in Sect. 5.

2 Computer-aided SCA design environment

In this section we introduce computer-aided side-channel analysis design environment (CASCADE[1]). We begin by delineating the design rationale. Next, we describe its main components and their interaction. Lastly, we present the models for timing and power simulation along with the simulation methodology.

2.1 Design rationale

The goal of CASCADE is to incorporate SCA evaluations at design time into standard cell design flow. We aim to combine knowledge of both EDA and SCA communities to develop a tool easily applicable in practice. CASCADE is built around commercial EDA tools and associated data formats. We adhere to the standard cell design flow by using EDA simulators to obtain instantaneous power consumption estimates, starting at the earliest stages of design. In order to embed SCA evaluations in all of the standard cell design stages, we design and implement additional software components that bridge the gap between EDA tools and SCA evaluations at design time. This requires addressing several challenges. Firstly, there exists a gap in current timing and power models used in EDA contexts. Timing models are primarily targeted for performance, while power is primarily a concern for heat dissipation and battery life. In contrast, SCA evaluations depend on less studied models for instantaneous power consumption estimation. Secondly, there is a gap in the handling and interpretation of simulator outputs. SCA evaluations require processing of up to millions data-dependent simulations. Therefore, enabling mechanisms to efficiently generate and cope with sheer volumes of data is of critical significance for practical applications.

SCA evaluations using simulations We argue that 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) uniformly, without any additional manual input. 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 preprocessing. Unlike FPGA evaluations, we rely on a one-to-one model of an ASIC circuit. Compared to the inherently serial nature of data acquisition from a chip, simulating multiple power traces in parallel is trivial. We stress that models, as simplifications of physical phenomena, can never fully capture the reality. Hence, simulations are only as accurate as the models they use, and they cannot 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 three aspects of simulations. Evaluations must be available as early in the design flow as possible and 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. 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.

SCA Assessment Estimates of information leakage obtained at different abstraction layers need to be analyzed in order to assess the security of a circuit. Since we want to be able to quickly assess an arbitrary piece of design, we prefer generic methods over batteries of attacks. A possible approach is to use the information-theoretic metric proposed by Standaert et al. [33]. While it is certainly useful and possible to integrate in our setting, estimating probability distributions may be too computationally and memory intensive. Instead, we focus on SCA evaluation by means of leakage detection. In particular, the TVLA methodology [9] uses the t-test distinguisher to detect statistical dependencies between sensitive data and side-channel information contained in the instantaneous power consumption measurements. The test analyzes two sets of measurements partitioned according to sensitive information. Assume μ_i, σ_i^2, and n_i to be sample mean, variance, and cardinality of set i, respectively, where $i \in \{1, 2\}$. Then, the two-tailed Welch's t-test is used to compute t as per Eq. 1.

[1] A snapshot of CASCADE is available at: https://github.com/dsijacic/CASCADE.

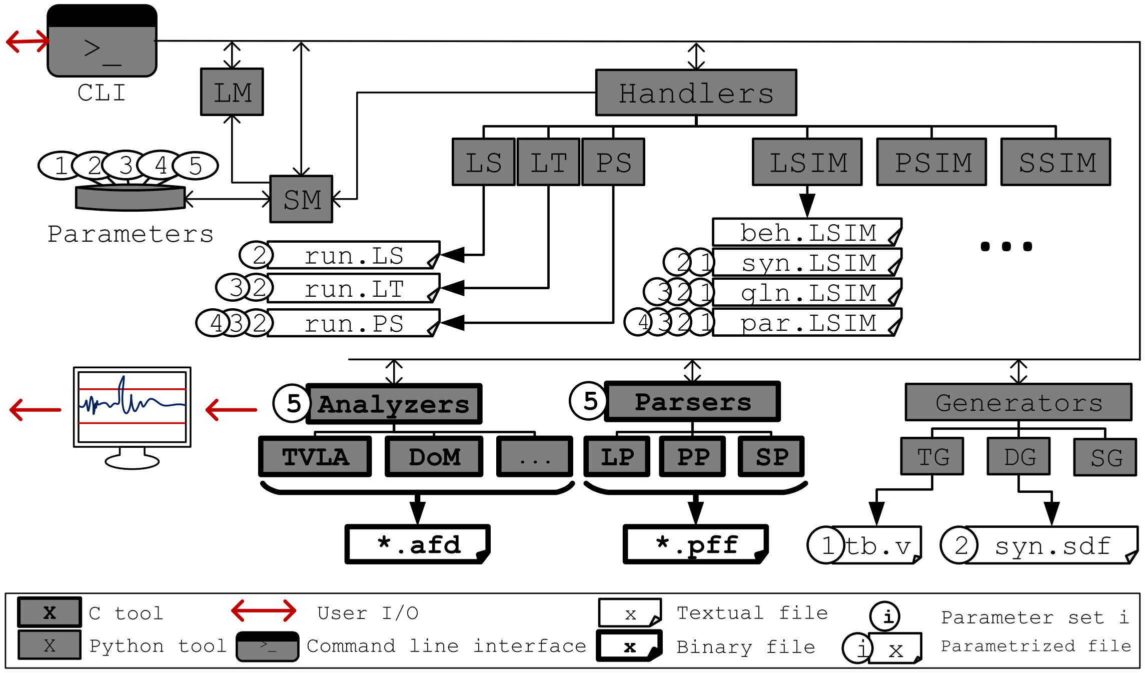

Fig. 1 High-level architecture of CASCADE

$$t = \frac{\mu_2 - \mu_1}{\sqrt{\frac{\sigma_2^2}{n_2} + \frac{\sigma_1^2}{n_1}}}. \tag{1}$$

If the t-value is outside the ±4.5 range, the test rejects the null hypothesis with confidence greater than 99.999% for a significantly large numbers of measurements [9]. Null hypothesis being that all samples are drawn from the same distribution, this indicates the distributions of the two sets are distinguishable and thus shows the existence of side-channel leakage. The instantaneous power consumption 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 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. TVLA computed using Eq. 1 is the first-order TVLA, as it checks for the existence of leakage in the first-order statistical moment. Higher-order moments can be used for more rigorous security assessment, although they are more demanding in terms of computational power. Efficient computation strategies for different orders have been recently put forward in [31,32]. In practice, TVLA 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.

Table 1 Configuration parameters

	Category	Examples
①	Simulation	Precision, duration
②	Design constraints	Critical path
③	Resources	Library resources
④	Physical constraints	Placement constraints
⑤	Power	Model parameters

2.2 Framework description

CASCADE allows automated and efficient SCA evaluation during all stages of the standard cell design flow. While it is easily extensible, its current modules are depicted in Fig. 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 shown in Table 1.

The `SM` centrally manages all configuration parameters. After evaluation, they are returned to the different tools according to their desired format. 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 standard cell library files. The

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, PX
SSIM	SPICE simulation	Synopsys HSPICE

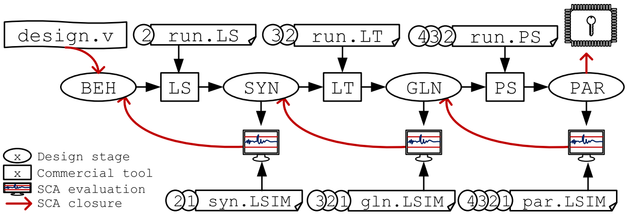

Fig. 2 Standard cell design stages using CASCADE

remaining components are: `Handlers`, `Generators`, `Parsers` and `Analyzers`.

`Handlers` wrap 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 a 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 Fig. 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 using models described in Sect. 2.3.

`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`. `TG` parses the Verilog netlist, wires the design, and facilitates control signals for data input and capturing (e.g., `trigger` signal that indicates when to record power consumption). All status and control signals used to configure and run a particular design should be handled manually. Test benches for different configurations can be easily added. Input data are read from a binary file, resulting in otherwise unchanged structure of each test bench. Depending on the desired test `TG` generates different data vectors (e.g., user-supplied functional tests, (pseudo)random inputs or EDPC sequence). In case of pipelined designs, such as TI circuits, to observe the worst case we keep the inputs stable until data have finished propagation. Feeding subsequent data would introduce noise as the pipeline would be computing on multiple statistically independent inputs at the same time. Delay generator (`DG`) is used to annotate generic netlists at `SYN` design stage (c.f. Δ-delay in Sect. 2.3). Delay annotations are stored in the standard delay format (SDF), compliant with modern EDA tools. 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 measurement setups, we design optimized `Parsers` to process and store data in a SCA-friendly manner. We design and implement them 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 storing simulated instantaneous power consumption traces. We refer to the part of simulation that corresponds to one power trace as simulation frame. Each frame starts with data associated with the frame transitions followed by time–value pairs of discrete digital events. We allow associating three vectors with each frame: input, output, and target data vector. CASCADE configuration allows binding these three vectors to arbitrary nodes. The latter allows us to leverage the native simulators to perform any post-processing, e.g., unsharing the sensitive variable. Frame-associated data also support functional validation of the design and SCA processing, e.g., frames can be partitioned on the fly.

Lastly, we use `Analyzers` to process PFF files. We design and implement them in C. Each analyzer can implement a specific SCA assessment technique, e.g., TVLA, or an attack, e.g., DPA or CPA. Focusing on TVLA, we follow the roadmap of Schneider and Moradi [32]. We abstain from applying the faster leakage assessment of Reparaz et al. [31] because of the prohibitive cost of storing 2^{64} histograms. We discuss this topic further in Sect. 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, PFF header information, and desired parameters. Secondly, analyzer's context is updated with this waveform. And thirdly, we evaluate the context and write the output trace to analyzed frame data (AFD) file, a 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. AFD files can preserve the analyzer context, so that on-the-fly evaluation can be continued at will.

2.3 Simulation models and methodology

Analog SPICE models, albeit the pinnacle of electronic modeling in terms of accuracy, feature exponential increases in runtimes with the increase in circuit sizes. We do support them in our framework, as they are useful as a reference for smaller validation circuits. For practical reasons, we focus on timing and power models that have the potential to scale efficiently.

2.3.1 Models from the SCA community

The design of masking schemes often relies on minimal assumptions when modeling the underlying hardware. Early works, such as [38], retain their security only in the zero-delay model, i.e., they can be broken due to the effects of glitches caused by the propagation delay in CMOS circuitry. Modern masking schemes prevail in the presence of glitches, e.g., non-completeness property of TI. Splitting sensitive values in multiple shares and performing independent computations ensures no glitch in any of the shares leaks information about secret values. Consequently, modeling of circuit timing under these assumptions is not of great concern. This allows for the use of generic, library independent, Δ-delay models where each gate is assigned a fixed delay. On the contrary, the secure logic style community often relies on detailed SPICE-level simulations for the evaluations of their designs.

Leakage models often employed by the SCA community are based on the Hamming distance and the Hamming weight of the processed data. Both are based on the predominance of dynamic power consumption in CMOS logic. Hamming distance model maps every toggle with a Dirac-like pulse of unitary amplitude. Multiple toggles that happen at the exact same time are simply added together. Hamming weight model maps the number of logic ones to the amplitude of the Dirac-like pulse, without considering previous states. It is particularly useful for evaluating software implementations, where periodically pre-charged buses are the main source of leakage.

2.3.2 Models from the EDA community

Timing parameters determine performance constraints, e.g., setup and hold times. Hence, models for timing simulation (closure) are at the heart of EDA tools. Standard cell 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 standard cell libraries. In the PAR stage, delays are extracted from the actual physical layout. Therefore, at PAR stage impact of concrete parasitic elements is taken into account. These are detailed models for timing and power consumption, in the Open-Source Liberty format from Synopsys, compatible across EDA vendors.

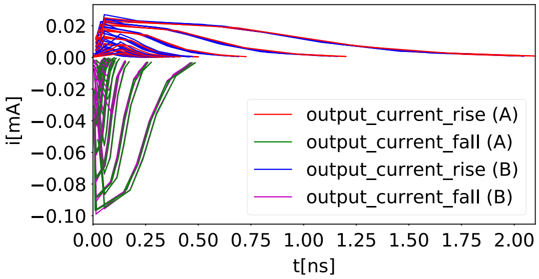

Fig. 3 CCS output current waveforms of a XOR_X1 gate

CASCADE uses composite current source (CCS) models, shown to be capable of producing power and timing estimates very close to SPICE [24]. The output current waveforms of each standard cell are captured through the process of characterization. Characterization employs detailed SPICE-level simulations, using technology parameters known only to the technology company that produces the library. It assumes different input slopes and output loads to account for different "surroundings" of the cell in the layout. Figure 3 depicts all of the output current waveforms for the XOR_X1 gate of the NanGate 45 nm library. A remarkably high level of detail is captured, such as asymmetries caused by toggles at different pins. As the underlying Boolean function is commutative, this asymmetry cannot be detected without observing physical properties of the cell. As such, CCS models are an industry standard used for "golden" sign-off estimations.

Similarly, timing, power, and noise tables are formed and stored into library files. For a given design and constraints, these curves are used to estimate timing, power and signal integrity, respectively. Coming from this abundance of information, it is interesting to see the corresponding power waveforms obtained using commercial EDA tools. In particular, we use PrimeTime with PX add on. We henceforth refer to these simulations as PTX. Figure 4 depicts instantaneous power consumption of the XOR_X1 gate, evaluated using PTX for several transitions, i.e., frames. In both cases, XOR_X1 is driven using DFF_X1. Solid blue line is obtained when the loading capacity of the output pin is set to the capacity of the DFF_X1/D pin. Dashed red line is obtained by increasing this capacity four times. When evaluating small isolated circuits using PTX, it is important to set proper design constraints. Although these representations contain more information than the unitary pulses normally used in

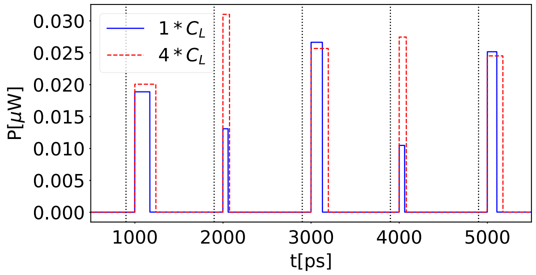

Fig. 4 PrimeTime PX simulation of the `XOR2_X1` gate using CCS models. Dotted black lines separate frames

the SCA world, the output waveforms are clearly designed with average power consumption in mind.

2.3.3 Consolidating timing and power models

We use parametrized Δ-delay model to bridge the gap between SCA and EDA worlds. Such models are useful to provide assessments before a specific library is introduced. Equation 2 states the general form of the parametrized Δ-delay model.

$$\Delta = \delta(1 + F\theta) \tag{2}$$

Here, δ is the fixed propagation delay, F is the fanout and θ is the scaling factor. By choosing $\delta = 0$, the model is reduced to $\Delta = 0$, i.e., zero-delay model. By choosing $\delta > 0$ and $\theta = 0$, the model is equivalent to fixed $\Delta > 0$ delay model. Lastly, for $\delta > 0$ and $0.05 \leq \theta \leq 0.20$ we define fanout-dependent delay Δ, F. We chose the range for θ based on empirical observations of several modern libraries and use these values in our experiments.

$$\mathrm{P}_{0\to1} = 1, \ \mathrm{P}_{1\to0} = 1 - \alpha \tag{3}$$

We expand the Hamming distance model into the marching stick model (MSM), named for its graphical interpretation. MSM is described by Eq. 3. The parameter α is used to address the asymmetry of rising and falling edges. MSM evaluations are computed by the logic parser (`LP`) and can be used on top of logic simulations orthogonally to the underlying timing model. We can relate MSM to CCS power models in the same manner as Δ-delay models relate to the timing ones. We compare the quality and performance of SCA evaluation using MSM versus PTX simulations at different design stages. MSM estimations are adjunct to the logical simulations. They are a precursor for the event-driven instantaneous power consumption estimations using CCS power models. Hence, they are the common case in terms of performance and scalability.

2.4 Simulation methodology

Our methodology is closely coupled with every stage in the traditional standard cell design flow. While these stages are alike to the functional simulations for timing closure, the rationale behind them is completely different. In traditional design, flow designers care about the values in the steady state, i.e., after all transitions have settled. We rather focus on the transitions, observing changes in the instantaneous power consumption caused by an input change. Since we make no assumptions about the functionality of target circuit (other than it being a digital circuit), this allows us to apply the approach to any standard cell design. Consequently, we can analyze implementations of masking schemes, standard cell secure logic styles or any other block of digital hardware in the same manner. In order to capture all possible transitions of a circuit with n input bits, we need to simulate $2^{2n} - 2^n$ non-trivial transitions. We call this simulation sequence exhaustive dynamic power capturing (EDPC). We use Algorithm 1 to ensure traversal of all transitions without repetition. The exponential complexity of EDPC makes it infeasible for circuits with large number of input bits. Our tests indicate that EDPC is feasible for circuits with up to 16 input bits. This is suitable for rigorous evaluations of smaller, but SCA critical, blocks. For larger designs, we generate inputs in a pseudorandom fashion. This is analogous to the acquisition in laboratory settings.

All simulations are driven by test bench output by the test bench generator (`TG`). Hierarchical designs may cause port nets of submodules to be annotated multiple times. To avoid counting the contribution of these nodes multiple times, two paths can be taken. First, hierarchical netlist can be flattened during synthesis. Second, logic simulator can be instructed to optimize away the redundancy. In QuestaSim, this can be facilitated using `-voptargs="+acc=prn+<testbenchModuleName>"` argument of the `vsim` command.

Algorithm 1 EDPC Sequence Generation.

```
function EDPC($nbits$)
    $init \leftarrow 1, jump \leftarrow 1, node \leftarrow 0, space \leftarrow 2^{nbits}$
    for $i \in [0, 2^{2 \cdot nbits} - 2^{nbits})$ do
        yield $node$                                   ▷ EDPC sequence value.
        $node \leftarrow (node + jump) \mod space$
        $jump \leftarrow (1 + jump) \mod space$
        if $node = 0$ then
            $init \leftarrow (init + 1) \mod space$
            $jump \leftarrow init$
        end if
        if $jump = 0$ then
            $jump \leftarrow 1$
        end if
    end for
end function
```

3 Framework validation

In this section, we validate CASCADE by applying it to representative cryptographic circuits. The security properties of these circuits are well established and therefore allow us to check the capabilities of our tool. We show how CASCADE 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. Lastly, we benchmark CASCADE using several circuits of different input spaces and area to test scalability of CASCADE. We use a 45-nm open-source SC library from NanGate.

3.1 Motivating example

We use the first-order DOM-indep multiplier (AND2 gate), a masking countermeasure from [16] depicted in Fig. 5 (top left), as a motivating example. In this simple circuit, input and output variables are split into 2 shares such that $a = a_1 \oplus a_2$, $b = b_1 \oplus b_2$ and $c = c_1 \oplus c_2 = \mathrm{AND(a, b)}$. 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 5 (top right) shows various MSM power profiles (averaged traces) based on different timing models. With a total of 5 input bits, EDPC consists of $2^{2\cdot 5} - 2^5 = 992$ input transitions. In 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. And this is indeed true if all 992 frames are used.

For the purposes of validation, we induce a flaw in the design by violating one of its security conditions. We break the independence of inputs condition by using only the frames where $a_1 = b_1$. Figure 5 (bottom left) depicts the resulting information leakage, in the first cycle. Next, we turn the masking off by fixing the value of $z = 0$. Figure 5 (bottom left) depicts the resulting information leakage, now in the second cycle. All findings hold across the other models we use. We plot results obtained using Δ-delay simulations for simplicity. Clearly, 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 due to their nonlinearity. We show how to use CASCADE regardless of the underlying countermeasure.

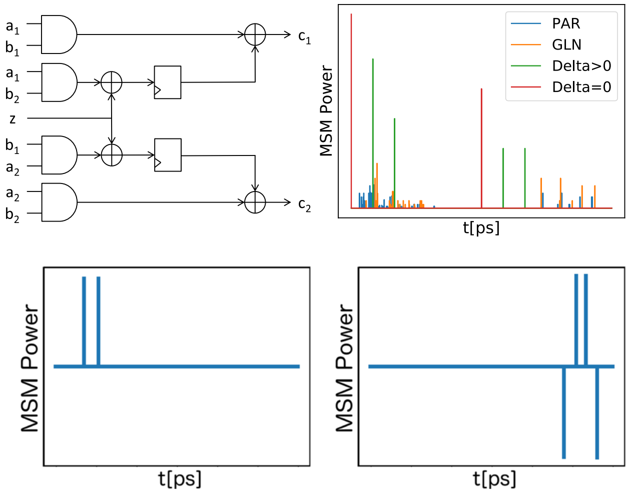

Fig. 5 DOM-*indep* AND2 gate (top left) and MSM power profiles (top right). First-order differential traces for dependent inputs $a_1 = b_1$ (bottom left) and for a fixed z (bottom right), using Δ-delay simulations

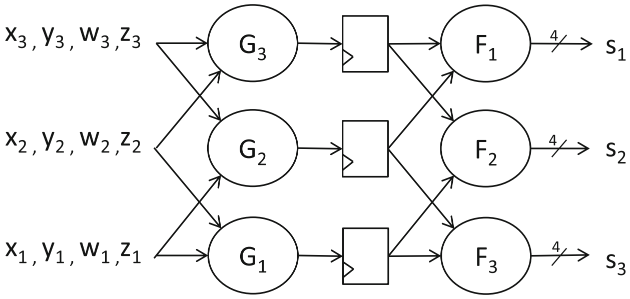

Fig. 6 Architecture of the TI PRESENT *S*-box

3.2.1 TI Present *S*-box

We target the first-order secure threshold implementation (TI) PRESENT *S*-box by Poschmann et al. [28], depicted in Fig. 6. 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.

As it is designed to provide first-order security, we first check for the existence of leakage in the second-order statistical moment. The resulting second-order t-traces (TVLA2) show significant leakage, as expected, for all the models CASCADE currently supports. Figure 7 (top) shows the results obtained using MSM simulations. The plot shows comparable levels of detected second-order leakage between using fanout-dependent Δ-delay at SYN stage and full CCS

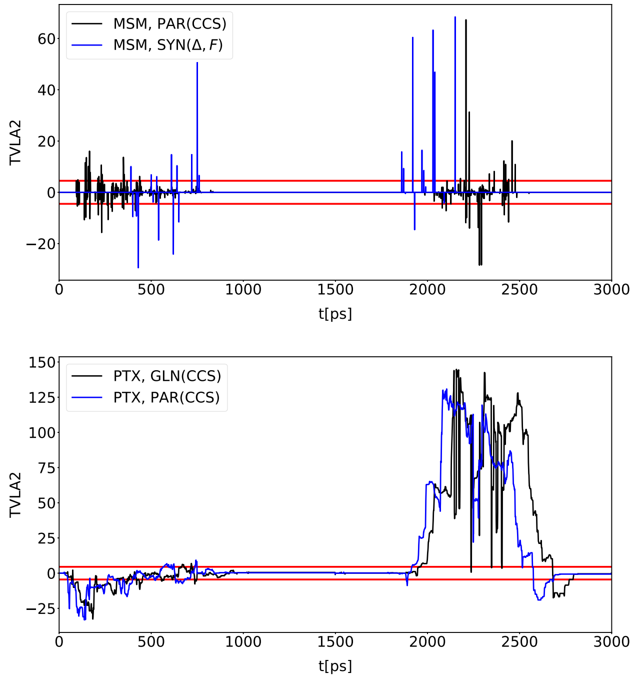

Fig. 7 Second-order t-trace using MSM (top) and PTX simulations (bottom); TI PRESENT S-box

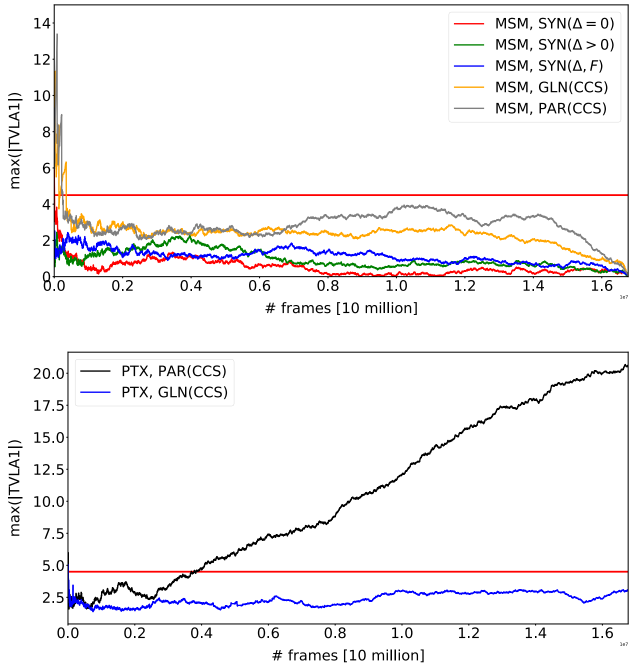

Fig. 8 First-order t-trace evolution using MSM (top) and PTX simulations (bottom); TI PRESENT S-box

timing with extracted parasitics at PAR stage. The second-order leakage is even more prominently detected using PTX evaluations based on full CCS timing and power models, as depicted in Fig. 7 (bottom). For both MSM and PTX evaluations, second order is present in both cycles, as expected.

Using MSM simulations, exhausting the EDPC sequence and testing for first-order leakage (TVLA1) returns a t-trace set to zero. Instead of a flat line, we plot the evolution of the peak absolute value of the TVLA1 in Fig. 8 (top). As TVLA is a statistical method, it is possible for the t-trace to briefly leave the confidence interval due to an insufficient amount of processed measurements. With the increasing number of frames, the law of large numbers takes over and the t-trace settles within the confidence interval. Hence, it is important to observe the trend of the t-trace, not a single evaluation with a relatively small number of frames. When all the possible transitions are exhausted, all t-traces evolve to zero. This is to be expected as all observations are noiseless and the uniformity property of TI guaranties that every input and output value appears with the same probability. Once all transitions are exhausted, we effectively fully populate distributions for each fix value, including the value fixed zero based on which we partition.

The results obtained for PTX simulations are shown in Fig. 8 (bottom). While simulation at GLN stage yields a similar outcome as obtained for MSM, i.e., no leakage, the outcome t-trace at PAR stage steadily evolves above zero. We speculate this effect is caused by physical asymmetries (recall Fig. 3) coming into play, making the contributions of toggles not egalitarian. In contrast, at GLN stage capacitances are extracted based on statistical wire load models; hence, they can cause not coupling between the shares. Being a pre-layout stage, no potential sources of coupling between shares exist. Hence, non-completeness of TI is preserved and design is evaluated as secure. From the work of De Cnudde et al. [8], we know that leakage can "unexpectedly" appear in a TI design. The authors attribute this to an unknown source of coupling, the power distribution network (PDN) or ground couplings being prominent candidates. Yet in our experiments, we do not include PDN, nor use advanced extraction techniques to capture ground coupling. Moreover, the indication of first-order leakage using PTX evaluations at PAR stage prevails whether we extract capacitances as coupled, lumped to ground or whether we completely omit reading capacitances.

Figure 9 shows the first-order t-traces obtained at the end of 16 million traces evolution depicted in Fig. 8 (bottom). All the significant leakage takes place in the second clock cycle, indicating a potential composability issue. When analyzed separately, as fully combinatorial designs, both G and F components do not show first-order leakage. This behavior is expected since both G and F are correct, uniform, and non-complete. Hence, they can be freely composed together. For discussion on this apparent contradiction, see Sect. 4.

Lastly, we check whether intentionally introduced vulnerabilities in the design can be captured with our models. Figure 10 (top) shows the first-order t-trace evolution when

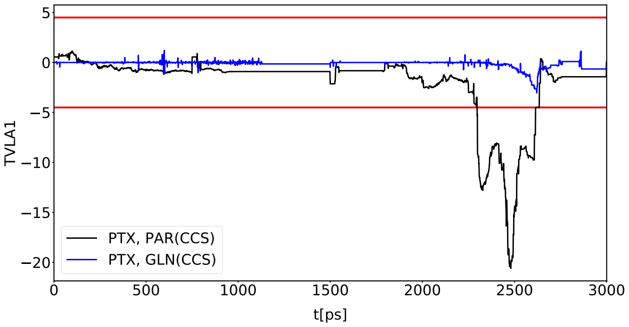

Fig. 9 First-order t-trace using PTX with 16M traces

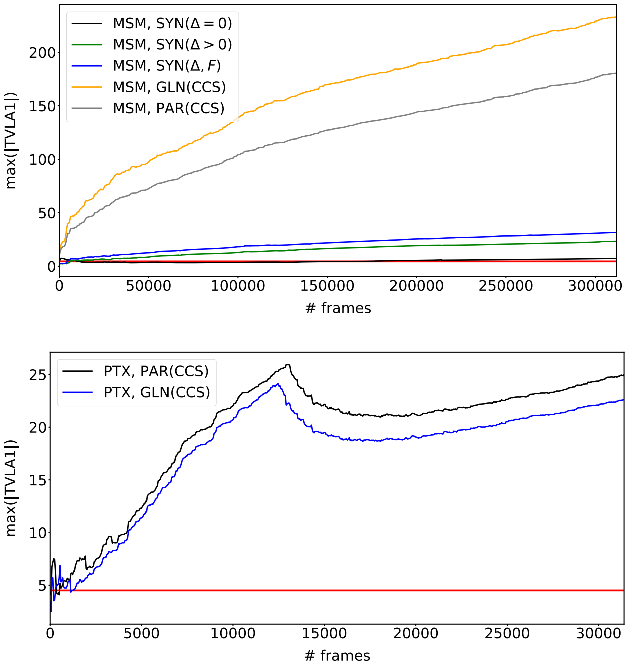

Fig. 10 First-order t-trace evolution with one TI share turned off using MSM (top) and PTX (bottom)

turning off one share of the TI PRESENT S-box, i.e., equivalent to setting $x3 = y3 = w3 = z3 = 0$. Leakage is correctly detected even with the simplest $\Delta = 0$ model, roughly after processing 250k traces. Using any other timing model results in even faster detection, down to 10k traces. Results obtained when using PTX simulations with CCS models are shown in Fig. 10 (bottom). In this case, first-order leakage is visible after processing the first 2k traces.

3.2.2 Boyar–Peralta AES S-box

Ghoshal and De Cnudde [13] proposed a first-order secure implementation of Boyar–Peralta AES S-box, designed to consume no randomness. In a later work, Wegener and Moradi [39] showed that this design exhibits leakage due to a uniformity problem. Their experiments were carried using an FPGA setup and processing 10 million measurements. We have validated the same vulnerability can be captured using CASCADE. In particular, our experiments indicate the presence of significant leakage starting from 400k MSM frames at `GLN` stage, using CCS timing models. Such evaluation can be performed in 30 min, including manual work, and using a single desktop workstation.

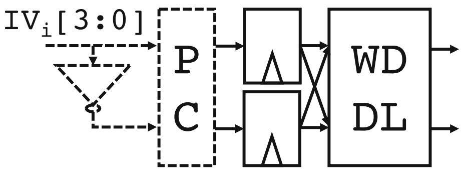

Fig. 11 Principal architecture of a WDDL design

3.2.3 WDDL Present S-layer

WDDL is a dual-rail secure logic style compatible with standard cell design flow. 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 represents a more realistic setting. The larger circuit is more difficult to balance and captures the routing effects more prominently. The S-layer contains all the logic gates of a PRESENT round, excluding the key schedule, as the remaining P-layer is mapped to simple wiring in hardware. Given the impossibility to exhaust all $2^{2\cdot 64} - 2^{64}$ EDPC transitions, in this experiment we perform a classical fixed versus random first-order leakage detection test using 10 million frames.

The principal architecture of WDDL logic is depicted in Fig. 11 The differential pair of WDDL modules needs to be periodically pre-charged (`PC`) and evaluated. For example, the `AND2` gate computes $a \cdot b = c$. The WDDL version of this gate, `WDDL_AND2`, computes $a_p \cdot b_p = c_p$ (positive end) and $a_n + b_n = c_n$ (negative end). Hence, the `WDDL_AND2` gate consists of one `AND2` gate and its complement `OR2` gate. In other words, the differential networks are mutually dependent according to De Morgan's law. In the pre-charge 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. In case of the `WDDL_AND2` gate, it will always be exactly 1. 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 cannot be fully attained. Still, if the asymmetries remain small enough WDDL circuits can be secure for a very large number of traces. We implement the pre-charge control (dashed lines in Fig. 11) as a part of the test bench and focus on the worst-case evaluation of the registered S-layer.

Since the security of WDDL is based on physical symmetries, MSM evaluations during the `SYN` stage always yield a constant zero value of the first-order t-trace, as parametrized Δ-delay model is always symmetrical. Figure 12 (top) depicts the evolution of the first-order t-trace during `GLN` and `PAR` stages. In this case, imbalances in the `AND2` and `OR2` gates of the target standard cell library result in the t-trace not being zero. Instead, it evolves within the confidence level with a slightly downward trend. A similar result is obtained in the `PAR` stage, with the t-trace evolution trend slightly rising, but well within the confidence level (for 10 million traces). Figure 12 (bottom) shows the equivalent plots obtained when using PTX simulations at `GLN` and `PAR` stages. Interestingly, the additional about the amplitude and duration of each event-driven toggle has no significant impact compared to the

The increasing trend of the `PAR` t-trace evolution indicates that the design might start leaking information with more processed frames. Being a noiseless evaluation, one might argue that the number of measurements to exploit this apparent leakage might be prohibitive in a real setting. This is simply because any type of measuring equipment will introduce noise to the traces, increasing the complexity of the attack in terms of MtD. On the other hand, the models we use do not account for the effects of manufacturing artifacts such as process variations or the routing of the power distribution network. Therefore, considering the implementation as secure might be too strong a claim. As discussed in Sect. 2, insight into this matter demands to compare our simulations with real chip measurements.

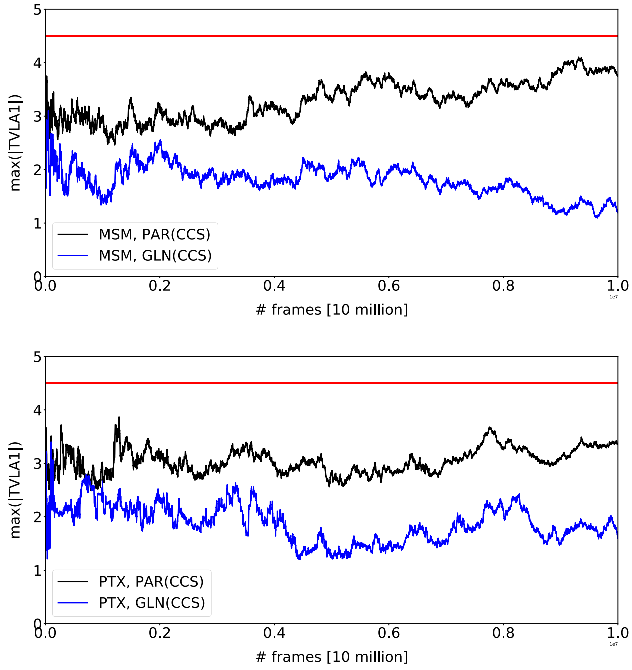

Fig. 12 First-order t-trace evolution using MSM (top) and PTX (bottom); WDDL PRESENT S-layer

4 Discussion

In this section, we discuss SCA evaluations at design time using CASCADE. We first argue its utility for digital designers. Then we show its feasibility on a set of representative cryptographic circuits, providing practical insights into the section.

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 efficient early SCA evaluations of critical building blocks prior to integration. Designers may pinpoint bugs and flaws, and proceed to fix them before moving on to the next stages. CASCADE can be applied regardless of the target countermeasure, as long as the design is implemented using standard cell libraries. All data and control paths along with any other auxiliary gates are treated uniformly and automatically, without the need for additional modeling. The previously mentioned analysis of the Boyar–Peralta S-box is a good example of CASCADE's practical utility. As a largely automated framework, it can be used effectively, avoiding problems with measurement setups and saving time. MSM evaluations allow designers to detect and correct vulnerabilities in the pre-layout stages. For masking schemes, such as TI, flaws can be corrected without considering the specifics of a particular library. Furthermore, in all our experiments on masked circuits CASCADE detects leakage with a relatively low number of frames. Hence, even for larger designs flaws can be detected long before input transitions were exhausted.

A popular alternative to simulation is FPGA prototyping. Its advantage is that measurements are directly obtained from a chip. Hence, evaluations include physical effects due to noise, thermal drift, real measuring equipment, etc. Unfortunately, the internal structure of FPGA is radically different from the 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. [8], who investigate the impact of coupling effects on protected designs implemented 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. To make the security evaluations based on simulations reliable, two important considerations must be made. 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 cannot be observed in practice. This may lead to false positive assessments (i.e., the tools indicates leakage, but in practice the circuit cannot be broken). We believe that the first-order leakage found using PTX evaluations of TI PRESENT *S*-box at PAR stage falls under this category. We have verified that the first-order leakage is reduced when degrading the precision of our tools or when artificially adding low-level noise to the simulations. Such artifacts, which are inherently present in practical settings, represent a potential direction to reduce the impact of false positives. Determining meaningful levels for these magnitudes is, however, beyond the scope of this work. 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). The WDDL PRESENT *S*-layer evaluation falls definitely under this category, as the tool does not account for, e.g., process variations or early propagations. Secondly, we see in all the examples that the precision of CCS models, even when degraded to form PTX rectangles, is more than enough to detect leakage. Therefore, more focus should be put into qualitatively determining which elements, parasitic or otherwise, contribute to the existence of leakage. Investigating this demands further analysis of power and RC extractions models. Overall, it is clear that there exists a gap in comparison with 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. Its sufficiently long EDPC sequence makes the simulation efforts non-trivial and convenient for comparing performance given a fixed circuit. The runtimes in minutes of logic and power simulations and their processing are summarized in Table 3. MSM evaluations involve a sequence of LSIM → LP → TVLA*, whereas PTX evaluations involve a sequence of PSIM → PP → TVLA*. Here, TVLA* can either include TVLA1 or TVLA2. For all cases we simulate, parse and analyze all $2^{12 \cdot 2} - 2^{12}$ transitions using a single thread of a Intel i7-7700 desktop workstation.

Our simulations are done with 1ps precision, as this is the precision given in CCS libraries. Simulating at lower precisions can save the parsing and analysis time as it produces less output samples. However, given that the tools internally perform computations at 1ps precision, the simulation times remain unchanged. Consequently, downsampling does not yield substantial performance benefits.

Table 3 EDPC performance benchmark for TI PRESENT *S*-box for different tools, stages, and models

	LSIM/PSIM	LP/PP	TVLA1	TVLA2
MSM, SYN($\Delta = 0$)[1]	4.85	1.65	0.03	0.05
MSM, SYN($\Delta > 0$)	7.15	1.98	0.92	4.13
MSM, SYN(Δ, F)	7.30	2.13	0.97	4.16
MSM, GLN(CCS)	11.38	2.14	1.05	4.23
MSM, PAR(CCS)	9.25	2.17	1.07	4.28
PTX, GLN(CCS)	57.75	5.15	1.53	4.85
PTX, PAR(CCS)	60.07	5.66	1.64	4.87

Runtimes are given in minutes

[1] Clock period is 10ps, as opposed to 1500ps in other cases

In our experiments, the simulation clock is set to 1500 ps resulting in 3000 samples per frame. The exception is made for $\Delta = 0$ simulations, where we use 10ps clock resulting in 20 samples per frame. The simulation times are primarily determined by the total number of events. More complex models cause more different propagation delays, resulting in more glitches and different toggling times. This trend holds for both logic simulation, LSIM, and power simulation, PSIM, with one exception. MSM simulation at PAR stage using CCS timing models produces more events ($\approx$ 1.3 billion) compared to its GLN counterpart ($\approx$ 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 cannot give a certain reason for this discrepancy. One possible reason might be the way the extracted SDF data are 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.

Runtimes of parsers and analyzers are depend on the number of samples and the number of events they have to process. The former dependency is easily observable in the $\Delta = 0$ example. The latter is observable in the increasing runtimes with the increased complexity of models.

The TVLA evaluations are performed on the fly using the approach of Schneider and Moradi [32]. The approach of Reparaz et al. [31] is effective when working with modern oscilloscopes as they provide 8–12 bits of resolution. Consequently, they require storing between $2^{2 \cdot 8}$ and $2^{2 \cdot 12}$ histograms to fully represent signal distributions. Simulations produce single-precision floating point traces. Hence, applying this approach would require storing $2^{2 \cdot 32}$ histograms. Simulated results can be quantized down to 8–12 bit range to allow the latter approach. Nevertheless, looking at Fig. 13, processing time of TVLA is just a fraction of the time needed

Table 4 CASCADE runtimes on a single thread of i7-7700 per 1 million `PAR` frames

#	Circuit	Area [kGE]	Frame [ps]	`LSIM` [h]	`LP` [h]	`PSIM` [h]	`PP` [h]	`TVLA1` [h]	`TVLA2` [h]
1	TI Present *S*-box	0.35	3000	0.01	<0.01	0.06	<0.01	<0.01	<0.01
2	WDDL Present *S*-layer	2.98	3600	0.15	0.17	0.31	0.05	0.02	0.04
3	BP AES *S*-box	5.45	25000	0.39	0.33	1.13	0.07	0.05	0.16
4	Unrolled AES-128	127.18	30000	25.81	2.58	52.61	1.17	0.14	0.25

for the simulation. Therefore, we find this optimization to yield negligible returns in our setting.

To test the scalability of our approach, we apply it to a fully unrolled implementation of AES-128. We use a placed and routed design of 127.18 kGE with extracted layout parasitics. Table 4 shows the average runtimes for simulating, parsing, and analyzing 1 million PAR traces of unrolled AES-128 along with the other circuits studied in this work. Figure 13 places these data into perspective and emphasizes that PTX evaluations must be preceded by the MSM evaluations. Since simulated frames have equal number of samples in both PTX and MSM evaluations, differences in runtimes of analyzers are negligible at these scales.

With the increase in the circuit area, the cost of simulations 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 are akin to the "golden sign-off" simulations for the timing closure. The size of the unrolled AES-128 exceeds security-dedicated area budgets of many embedded devices. Still, a million traces can be simulated and processed in less than 32 hours using MSM evaluations. This can be done on a single thread of the i7-7700 desktop workstation. Running simulations for different stimuli can be computed in parallel by simply dividing the input sequence into multiple batches. Experimenting with batch sizes between 10 and 100 thousand frames, we did not notice any significant performance difference. Batching can facilitate earlier estimates and alleviate storage issues. As shown in Sect. 3, leakage can be detected much before all frames are analyzed. Therefore, in practice only a fraction of batches may need to be processed if a flaw is present. Each batch is processed in the same manner as a whole simulation would be, updating the analyzer's context (in this case `TVLA`). As all computations are performed on the fly, there is no need for storing terabytes of simulated data dumped by logic simulators. In the unlikely case of limited disk space, storage requirements can be tweaked by adjusting batch sizes. Hence, using the same 8-thread workstation 1 million traces using MSM evaluations can be simulated and analyzed in less than 4 hours. Section 3 also shows that MSM evaluations are very capable of detecting leakage. PTX evaluations can achieve the same with smaller number of traces due to additional physical effects they take into account. Nevertheless, we do not think that this speed up if worth the added simulation time for early design time estimates. PTX evaluations do remain valuable for more detailed evaluations, especially in the post-layout stages. With the improvements of models and RC extraction techniques geared toward SCA, PTX evaluations will only gain more importance.

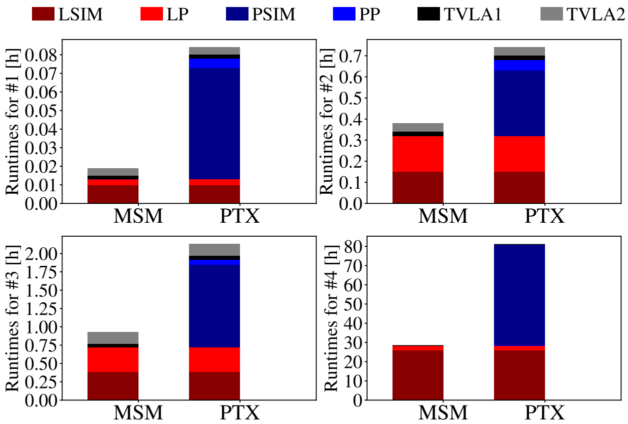

Fig. 13 Runtimes split between different evaluations for the four test circuits

5 Conclusions and future work

In this work, we have presented the design and implementation of CASCADE, a comprehensive framework for SCA evaluation at design time. CASCADE is built on the state-of-the-art EDA tools and SCA evaluation and methodologies, combining them in a methodical and automated manner. We show how 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 have benchmarked the performance of selected modules in our framework to show its aptitude in testing realistic cryptographic designs, and argued its feasibility for real-world use even when relying on a single desktop workstation. Additionally, we have discussed the use of composite current source models at length and presented how they can be used for side-channel evaluations. As 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, a snapshot of CASCADE has been released in the form of open-source software, available to the research community.

References

1. Aigner, M., Mangard, S., Menichelli, F., Menicocci, R., Olivieri, M., Popp, T., Scotti, G., Trifiletti, A.: Side channel analysis resistant design flow. In: 2006 IEEE International Symposium on Circuits and Systems, pp. 4 pp. 2912 (2006)
2. Balasch, J., Gierlichs, B., Verdult, R., Batina, L., Verbauwhede, I.: Power analysis of atmel cryptomemory - recovering keys from secure eeproms. In: O. Dunkelman (ed.) Topics in Cryptology-CT-RSA 2012—The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27 - March 2, 2012. Proceedings, *LNCS*, vol. 7178, pp. 19–34. Springer, Berlin (2012)
3. Bertoni, G., Martinoli, M.: A methodology for the characterisation of leakages in combinatorial logic. In: Carlet, C., Hasan, M.A., Saraswat, V. (eds.) Security, Privacy, and Applied Cryptography Engineering-SPACE 2016, pp. 363–382. Springer, Berlin (2016)
4. Bhasin, S., Danger, J., Graba, T., Mathieu, Y., Fujimoto, D., Nagata, M.: Physical security evaluation at an early design-phase: A side-channel aware simulation methodology. In: C. Berger, I. Schaefer (eds.) Engineering Simulations for Cyber-Physical Systems-ES4CPS 2014, p. 13. ACM (2014)
5. Bloem, R., Gross, H., Iusupov, R., Könighofer, B., Mangard, S., Winter, J.: Formal verification of masked hardware implementations in the presence of glitches. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology-EUROCRYPT 2018, pp. 321–353. Springer International Publishing, Cham (2018)
6. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J. (eds.) Cryptographic Hardware and Embedded Systems-CHES 2004, LNCS, vol. 3156, pp. 16–29. Springer, Berlin (2004)
7. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: Wiener, M.J. (ed.) Advances in Cryptology-CRYPTO '99, LNCS, vol. 1666, pp. 398–412. Springer, Berlin (1999)
8. Cnudde, T.D., Bilgin, B., Gierlichs, B., Nikov, V., Nikova, S., Rijmen, V.: Does coupling affect the security of masked implementations? In: Guilley, S. (ed.) Constructive Side-Channel Analysis and Secure Design -COSADE 2017, LNCS, vol. 10348, pp. 1–18. Springer, Berlin (2017)
9. 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)
10. Eisenbarth, T., Kasper, T., Moradi, A., Paar, C., Salmasizadeh, M., Shalmani, M.T.M.: On the power of power analysis in the real world: A complete break of the keeloqcode hopping scheme. In: Wagner, D. (ed.) Advances in Cryptology-CRYPTO 2008, LNCS, vol. 5157, pp. 203–220. Springer, Berlin (2008)
11. Fujimoto, D., Nagata, M., Katashita, T., Sasaki, A.T., Hori, Y., Satoh, A.: A fast power current analysis methodology using capacitor charging model for side channel attack evaluation. In: Hardware-Oriented Security and Trust-HOST 2011, pp. 87–92. IEEE (2011)
12. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2001, LNCS, vol. 2162, pp. 251–261. Springer, Berlin (2001)
13. Ghoshal, A., Cnudde, T.D.: Several masked implementations of the boyar-peralta AES s-box. In: Progress in Cryptology-INDOCRYPT 2017 Chennai, India, December 10–13, 2017, Proceedings, pp. 384–402 (2017)
14. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual information analysis. In: Oswald, E., Rohatgi, P. (eds.) Cryptographic Hardware and Embedded Systems-CHES 2008, LNCS, vol. 5154, pp. 426–442. Springer, Berlin (2008)
15. Goubin, L., Patarin, J.: DES and differential power analysis (the "duplication" method). In: Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems-CHES'99, LNCS, vol. 1717, pp. 158–172. Springer, Berlin (1999)
16. Gross, H., Mangard, S., Korak, T.: Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order. Cryptology ePrint Archive, Report 2016/486 (2016). http://eprint.iacr.org/2016/486
17. Kamel, D., Renauld, M., Flandre, D., Standaert, F.: Understanding the limitations and improving the relevance of SPICE simulations in side-channel security evaluations. J. Cryptogr. Eng. **4**(3), 187–195 (2014)
18. Kirschbaum, M., Popp, T.: Evaluation of power estimation methods based on logic simulations. In: Posch, K.C., Wolkerstorfer, J. (eds.) Austrochip 2007, pp. 45–51. Verlag der Technischen Universität, Graz (2007)
19. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: Koblitz, N. (ed.) Advances in Cryptology-CRYPTO '96, LNCS, vol. 1109, pp. 104–113. Springer, Berlin (1996)
20. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) Advances in Cryptology-CRYPTO '99, LNCS, vol. 1666, pp. 388–397. Springer, Berlin (1999)
21. Macé, F., Standaert, F., Quisquater, J.: Information theoretic evaluation of side-channel resistant logic styles. In: Paillier, P., Verbauwhede, I. (eds.) Cryptographic Hardware and Embedded Systems-CHES 2007, LNCS, vol. 4727, pp. 427–442. Springer, Berlin (2007)
22. Mangard, S., Schramm, K.: Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations. In: Goubin, L., Matsui, M. (eds.) Cryptographic Hardware and Embedded Systems-CHES 2006, LNCS, vol. 4249, pp. 76–90. Springer, Berlin (2006)
23. Moradi, A., Salmasizadeh, M., Shalmani, M.T.M., Eisenbarth, T.: Vulnerability modeling of cryptographic hardware to power analysis attacks. Integr. VLSI J. **42**(4), 468–478 (2009). https://doi.org/10.1016/j.vlsi.2009.01.001
24. Motassadeq, T.E.: 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), pp. 1–5 (2011)
25. Nikova, S., Rijmen, V., Schläffer, M.: Secure hardware implementation of non-linear functions in the presence of glitches. In: Lee, P.J., Cheon, J.H. (eds.) Information Security and Cryptology-ICISC 2008, LNCS, vol. 5461, pp. 218–234. Springer, Berlin (2008)
26. Oswald, D., Paar, C.: Breaking mifare desfire MF3ICD40: power analysis and templates in the real world. In: Preneel, B., Takagi, T. (eds.) Cryptographic Hardware and Embedded Systems-CHES 2011, LNCS, vol. 6917, pp. 207–222. Springer, Berlin (2011)
27. Popp, T., Kirschbaum, M., Zefferer, T., Mangard, S.: Evaluation of the masked logic style mdpl on a prototype chip. In: Paillier, P., Verbauwhede, I. (eds.) Cryptographic Hardware and Embedded Systems-CHES 2007, pp. 81–94. Springer, Berlin (2007)
28. Poschmann, A., Moradi, A., Khoo, K., Lim, C., Wang, H., Ling, S.: Side-channel resistant crypto for less than 2, 300 GE. J. Cryptol. **24**(2), 322–345 (2011)
29. Regazzoni, F., Cevrero, A., Standaert, F., Badel, S., Kluter, T., Brisk, P., Leblebici, Y., Ienne, P.: A design flow and evalua-

tion framework for dpa-resistant instruction set extensions. In: Clavier, C., Gaj, K. (eds.) Cryptographic Hardware and Embedded Systems-CHES 2009, LNCS, vol. 5747, pp. 205–219. Springer, Berlin (2009)

30. Regazzoni, F., Eisenbarth, T., Poschmann, A., Großschädl, J., Gürkaynak, F.K., Macchetti, M., Deniz, Z.T., Pozzi, L., Paar, C., Leblebici, Y., Ienne, P.: Evaluating resistance of MCML technology to power analysis attacks using a simulation-based methodology. Trans. Comput. Sci. IV Spec. Issue Secur. Comput. **4**, 230–243 (2009)
31. Reparaz, O., Gierlichs, B., Verbauwhede, I.: Fast leakage assessment. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2017, LNCS, vol. 10529, pp. 387–399. Springer, Berlin (2017)
32. Schneider, T., Moradi, A.: Leakage assessment methodology - A clear roadmap for side-channel evaluations. In: Güneysu, T., Handschuh, H. (eds.) Cryptographic Hardware and Embedded Systems-CHES 2015, LNCS, vol. 9293, pp. 495–513. Springer, Berlin (2015)
33. Standaert, F., Malkin, T., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Joux, A. (ed.) Advances in Cryptology-EUROCRYPT 2009, LNCS, vol. 5479, pp. 443–461. Springer, Berlin (2009)
34. Tiri, K., Verbauwhede, I.: A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In: Design, Automation and Test in Europe - DATE 2004, pp. 246–251. IEEE Computer Society (2004)
35. Tiri, K., Verbauwhede, I.: Simulation models for side-channel information leaks. In: W.H.J. Jr., G. Martin, A.B. Kahng (eds.) Design Automation Conference-DAC 2005, pp. 228–233. ACM (2005)
36. Tiri, K., Verbauwhede, I.: A vlsi design flow for secure side-channel attack resistant ICs. Design, Autom. Test Eur. **3**, 58–63 (2005). https://doi.org/10.1109/DATE.2005.44
37. Tiri, K., Verbauwhede, I.: A digital design flow for secure integrated circuits. IEEE Trans. CAD Integr. Circuits Syst. **25**(7), 1197–1208 (2006)
38. Trichina, E.: Combinational Logic Design for AES SubByte Transformation on Masked Data. Cryptology ePrint Archive, Report 2003/236 (2003)
39. Wegener, F., Moradi, A.: A first-order sca resistant aes without fresh randomness. Cryptology ePrint Archive, Report 2018/172 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.