Ask Less, Get More: Side-Channel Signal Hiding, Revisited

Itamar Levi, Davide Bellizia, David Bol, and François-Xavier Standaert

***Abstract*—Signal hiding countermeasures have been extensively investigated in the early side-channel attacks' literature. Due to design and physical imperfections, their stand-alone use only leads to a limited reduction of the attacks' complexity. As a result, more algorithmic countermeasures providing a more formal cost vs. security tradeoff (e.g., shuffling and masking) have gained more attention. Yet, since the cost associated with these is high, designers aim at combining countermeasures, leveraging the strength of each. In this manuscript, we demonstrate that by asking less to both signal hiding and algorithmic countermeasures (as stand-alone), we can develop combined countermeasures that indeed provide higher security at lower cost. For this purpose, we show how we can stack signal reduction and amplitude randomization techniques with ultra low cost automatic design flows and standard tools, and reach attractive security levels in combination with masking. Concretely, we examine two natural strategies for signal hiding and their combination: namely WDDL and a simple, local, scalable and easy-to-implement noise generation engine. A 65nm technology ASIC is evaluated with multiple isolated AES cores, leveraging recent information theoretic bounds which are connected to masking security proofs, significantly reducing the side-channel information leakage. We further quantify performance gains for masked designs.**

***Index Terms*—Differential Power Analysis, DPA, Dual-Rail Logic Styles, Hardware Security, Information Theoretic Metrics, Randomization, Side-Channel Signal Hiding, Worst-Case Security Evaluation, Noise Emulation, Masking.**

INTRODUCTION

SIDE-channel countermeasures already have a long history of (partial) successes and failures. There exists a rich literature detailing solutions to mitigate physical leakages at various abstraction levels. At the implementation level, hiding the side-channel signal has been proposed, for example by trying to balance the power consumed by a target device thanks to dual-rail logic styles [1]. At the design level, randomizing the data (i.e. masking [2]) and the execution of the algorithm's operations (i.e. shuffling [3]) have been proposed with the goal to amplify implementation level countermeasures. Finally, at the protocol level, leakage-resilient modes of operation have been proposed in order to make the exploitation of the side-channel signal more difficult, computationally [4].

One common pattern shared by these different protections is that their early instances usually fell short in solving the side-channel issue, because of various types of physical defaults. Hence, their stand-alone use only leads to a limited reduction of the attacks' complexity. For example, dual-rail logic styles were shown to suffer from the difficulty to perfectly balance the power consumption due to routing constraints and devices mismatch [5]; masking was shown to suffer from glitches [6] and couplings [7] and shuffling from the need for the shuffled operations to leak according to similar models [8], [9]. The main (informal) consequences of these negative results are (*i*) the heuristic rule-of-thumb that "*side-channel security can only be obtained with a combination of countermeasures*", leaving researchers in hardware security with the challenge of determining how to obtain the best security level at the lowest cost; and (*ii*) a faster development for countermeasures coming with better formalization, enabling to state clear guidelines for implementers.

In this context, masking has been among the fastest progressing countermeasures over the last decade, in good part due to an excellent combination of practical and theoretical progresses. On the practical side, concrete solutions have been formalized in order to maintain security amplification even in the presence of physical defaults such as glitches [10]. On a more theoretical side, abstract models allowing automated security proofs have been introduced [11], [12], extended to enable composability reasoning [13], and connected to more and more practically-relevant models [14]. Recent works even combined these practical and theoretical progresses into unified approaches [15], [16]. The combination of these results sets a sound basis to discuss the security vs. performance tradeoff of masked implementations with various number of shares. It also provides the adequate background to investigate different optimizations, for example in order to reduce the randomness cost of these implementations [17].

By contrast, progresses on lower-level protections for side-channel signal hiding have been scarcer. One reason for that is that countermeasures such as dual-rail logic styles are not easily captured by theoretical analysis: they do not come with a security parameter and can only bring a constant (technology-dependent) security improvement. Another reason is that noise addition solutions, which are more scalable, do not provide a satisfactory security parameter, since they essentially lead to linear security improvements at the cost of linear cost/performance overheads. In this manuscript we aim at pushing forward a sound methodology for combining countermeasures as to gain more than each independently. An example in that initial direction was demonstrated in [18] where a table lookup based masking was combined with WDDL circuits to compansate the parts where it was weaker (i.e. leaks more). In this work we aim at stacking countermeasures "*on-top*" of each other in order to amplify their effect.

I. Levi is with the Faculty of Engineering, Bar-Ilan University (BIU), Israel, e-mail: itamar.levi@biu.ac.il.

D. Bellizia, D.Bol and F-X. Standaert are with Université catholique de Louvain, ICTEAM/ELEN/Crypto Group, Belgium, e-mail: {first_name. last_name}@uclouvain.be.

Manuscript received March 25, 2020; revised May 20, 2020.

Our starting observation is that the more limited progresses of side-channel signal hiding are in part due to an improper (too demanding) goal. By nature, it is hard to argue strong security in the cryptographic sense (i.e. exponential security increase at the cost of polynimial overheads) based on physical countermeasures only. Hence, quantifying them based on the amount of security they provide as a stand-alone countermeasure is unlikely to bring convincing results (as the literature on dual-rail logic styles typically suggests).

Following, we revisit side-channel signal hiding and show that by asking less to such countermeasures, they can be an ideal complement to design leveled countermeasures with (for example) masking and lead to high security levels much more efficiently than without hiding. Informally, we show that since masking can be viewed as a noise amplification, increasing the noise, reducing the signal (or both) with side-channel signal hiding, before increasing the number of shares, is in general a good approach. In this paper we deal with how to do that.

Our main contributions in this respect are:

1) We analyze two natural strategies for side-channel signal hiding and their combination. First, the previously introduced WDDL logic style [1] which is among the most popular and understood dual-rail precharge circuits. Next, a simple, local, scalable and easy-to-implement noise generation engine is demonstrated. Both contribute to significantly reducing the information leakage.
2) We analyze these two solutions and their combination based on simulations and on a taped out ASIC in a 65nm technology. For this purpose, we show that simple metrics such as the (side-channel) Signal-to-Noise Ratio (SNR) [19] are not sufficient to capture complex leakage measurements (e.g. mixture distributions). We therefore use the recent information theoretic bounds [20] which can directly be connected to masking security proofs [14].

Based on these results, we quantify the performance gains that can be obtained by applying the proposed "*ask less, get more*" strategy to a masked implementation on a similar ASIC. We conclude that it offers a relevant way to combine countermeasures, and that side-channel signal hiding revisited in this way is an interesting building block for the design of secure and efficient implementations. Our results therefore open a possibility to revisit various other hiding countermeasures.

Paper organization. The manuscript starts with introducing the main objectives of the research and the relevant prior art. We follow with briefly surveying the implemented signal-reduction technique and noise-amplification technique in Section I as well as a brief elaboration on the tools we use for security evaluation. Section I is concluded with describing what can be theoretically achieved with masking in combination with hiding. We then present our contributions starting with an analysis on the extent of signal reduction, the proposed noise generation engine and the proposed locality driven architecture (in Section II). After we layout the main principles and building blocks used, we follow with a post fabrication security analysis in Section III. We then conclude the manuscript with a cost versus security tradeoff results and discussion, and an analysis of the theoretical benefits possible for masked designs adopting the proposed approach (Section IV). In Appendix A we elaborate on the automated design flow used and in Appendix B we elaborate on the security level subjected to environmental factors.

I. Background

In this section, we recall the structure, operation and a brief description of the used signal-reduction technique, namely Wave Dynamic Differential Logic (WDDL) [1]. We follow with a short background and discussion on randomization techniques. We then follow with a short security metrics overview and a discussion of using our "*ask less, get more*" approach as we theoretically combine it with masking.

Notations. In this manuscript, variables are denoted with (italic) capital letters, sampled values with lowercase letters, functions with sans serif fonts and vectors with bold letters. We use standard notations for the mean (μ) and standard deviation (σ), and we denote with $\mathsf{f}_{\mu,\sigma}(x)$ the probability density of a normal distribution $\mathcal{N}(\mu, \sigma^2)$ for a random variable X with realization x. We denote the (cumulative) probability with $\Pr(x \leq \alpha)$ and the conditional one with $\Pr(x \leq \alpha|\beta)$.

A. Signal Reduction

With the aim of focusing on signal reduction techniques, which can be implemented with standard cell libraries and tools, we recall the structure of a Dual-Rail Precharge (DRP) circuit and exemplify it with the WDDL logic style. Referring to Fig. 1(a) which illustrates the output voltage (V) and dissipated current (I) of a standard CMOS logic operation (f), it is clear that as the gate's output toggles, CMOS gates draw current which depends on the type of output transition. Whereas, if the output is stable, only leakage current flows through the power supply. A Dual-Rail encoding, as illustrated in Fig. 1(b), fixes the first problem: in addition to f, f' is implemented and complements the drawn current. In turn, it becomes insensitive to the type of output transition. However, some information is still *leaking* through the power supply current on the existence of an output toggle.

To fix this issue, dual rail precharge, DRP, logic was proposed, fostering a pre-computation Return-To-Zero (RTZ) spacer. That is, before each computation all the outputs are forced to '0' voltage. As a consequence, exactly one of the complementary outputs rises to a logical '1' in each computation cycle, concluding in similar current activity in each cycle, as illustrated in Fig. 1(c). One implementation of such a scheme is WDDL logic. In a WDDL circuit (Fig. 1(d)), each AND in the original circuit is appended with an additional OR. The inputs of this OR are complemented. Similarly, each OR is appended with an additional AND. As a result, for each gate output signal, it's complemented version is generated.

Concretely, samplers (Flip Flops) are duplicated and the input of one of the Flip Flops is inverted. Inverters (NOT functions) are generated by simple wire-crossing (as the complemented versions were already generated). In addition, the precharge signal can be derived from the main clock by using NOR gates at the inputs of the circuit following the

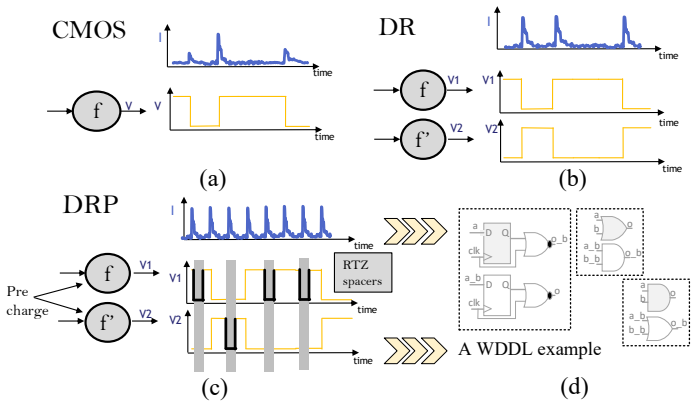


Fig. 1: Signal Reduction with standard devices: (a) CMOS single-rail gate, (b) Dual-Rail logic, (c) Dual-Rail Precharge (DRP) logic with RTZ coding, and (d) DRP example from standard-libraries cells. The ‘’’ on f represents a complement.

samplers, as illustrated on the figure. These induce a ‘0’ spacer before a computation starts at the circuit inputs, which then travels/propagates throughout the entire circuit.

In practice though, devices are not ideally symmetric nor is it possible to argue that electronic manufacturing (i.e. optical processes) can balance their physical characteristics and conclude in a perfectly constant power dissipation. Examples of known issues which break the symmetry assumption include complementary gates that do not have the same physical structure such as NANDs & NORs. Besides, even proximate transistors behave differently (due to process variations, mismatch etc.). In fact, any physical element such as routing may not be matched, either due to variations or due design tools limitations. And on top of all those, signal integrity and couplings affect different elements in different times , making perfect symmetry assumptions hardly realistic. Therefore, our goal is to deal with such imperfections and build up on this (limited) asymmetry in order to provide sound security claims.

For this purpose, the first ingredient we make use of is a standard DRP logic style for which the goal is only to reduce the side-channel signal (but not to annihilate it). In addition, usability is an important goal as well (i.e. we want this tool be simple and generic) and we propose an automated design flow to achieve this which is easy, fully standard and does not have any special requirements which may be hard to fulfill in real life design tools (e.g. Fat Wires or Backend Duplication [21], [22]) nor does it require custom design libraries. As will be discussed in the following, taking this approach of first reducing the signal will allow smaller additional noise generation and amplification requirements.

B. Noise Generation

In this subsection, we discuss some known techniques to generate noise and their associated cost. In addition, we motivate the use of True Random Number Generators (TRNGs), which have very low implementation cost. We also motivate why it might come as a useful complement for masked designs.

The efficient generation of physical noise in electronic circuits is a long lasting challenge. Generally, to increase the effective noise in the leakage, designers make use of RNGs. True Random Number Generators (TRNGs) exploit and extract randomness from physical noise sources in electronic systems. Traditionally, they are classified according to the mechanisms from which the noise is extracted. The two main groups are the metastability based and phase-jitter based generators. The first class aims to amplify very small voltage/current noise in a metastable structure (e.g. back-to-back inverter pairs, memory or flip flop elements), and to extract the final stable value. The second class, which we use next, aims to extract randomness from noise in the time domain. Many examples exist for oscillator based TRNGs (e.g. [23], [24]). Lately, [25], [26], proposed ultra small TRNGs which extracts randomness from phase-jitter in the time domain by utilizing a slow clock sampling a very high frequency jittery oscillator. Due to the expertise needed in designing TRNGs (and their increased sensitivity to external conditions), an alternative is to use Pseudo Random Number Generators (PRNGs). They generate numbers which are computationally hard to distinguish from uniform but are deterministic. For example, one of NIST PRNG recommendations is based on a cipher in counter (CTR) mode. While such PRNGs allow simpler security evaluations, they usually come at a very high electronic cost and low throughput, whereas TRNGs, under some assumptions, can be made significantly smaller ([25], [26]).

In this work, we therefore focus on area/energy-efficient solutions which can be embedded locally per module (e.g. operations on small four/eight-bit operands). For this purpose, we make use of tiny TRNGs seeding highly efficient amplitude randomization techniques. The proposed technique is based on fully EDA-supported flow for power gating, with standard cells libraries, which makes it very attractive. It is important to emphasize that alternative approach would be to leverage *global* amplitude randomization. Though system level solutions are highly attractive, they typically require some analog-circuits sub-parts, and in most cases are area-costly and requires micorelectronic expertise. For example, in [27], [28], global power regulators were proposed, assisted by randomization for some parameters of the regulator, concluding in a controlled randomized behavior. In this context, global implies that these mechanisms are not always small in area or they require special design and attention to affect/protect independently tiny parts of the system. For example, considering either a tampering adversary or a localized adversary with an EM probe, could lead to powerful attacks [29] on global building blocks. While such localized attacks are beyond the scope of this paper, they serve as a motivation for our localized noise generation engines. Despite these general remarks, we want to underline a break-through example for using regulators in a local way with a rather low-cost; this approach was investigated in [30], [31]. We believe, this is indeed a very interesting (localized) approach to examine, implement on a full system and evaluate for silicon results following the same lines of our “*Ask Less, Get More*” strategy.

C. Security Evaluation Tools

In this subsection we introduce tools for the two key steps in the security evaluations used in this work: first, identifying Points-of-Interest (POIs) in the leakage (where maximum

information is leaked), and next, bounding the security level of a device. For the latter we utilize well investigated information theoretic tools which were adjusted to the context of side-channel security evaluations and can be used for upper/lower bounding the worst-case security level of a leaking device. We consider an adversary who exploits a divide-and-conquer approach over an s bits subpart of the secret variable ($s \leq m$, where m the total number of state bits).

1) Identifying Points Of Interest: Many tools exist to identify POIs. As in this work we do not mask or split computations over multiple cycles, we focus on univariate analysis. That is, we ask which is the POI which is most informative. We tackle this question with several different tools: (1) attack based approach — we perform Moment-Correlating-Profiled DPA (MCP-DPA) attacks and standard profiling based attacks over time; (2) Signal-to-Noise ratio (SNR) and evaluation over time. We follow by finding the time sample which maximize either the attack Success Rate (SR) or the SNR. In most cases these two tools were consistent in the resulted POIs. However, the SNR was (1) faster to converge vs. the number of samples and (2) faster to compute as compared to MCP-DPA. Therefore, keeping one tool for presentation we focus on the SNR.

As introduced in [19] and investigated in numerous works, the SNR (in the side-channel sense) aims at indicating the univariate informativeness of a leakage time sample. To do so, signal and noise components are estimated. The Signal (i.e. the nominator) is estimated by first averaging out the noise per secret variable state (y), and then computing the variance over y. The Noise (i.e. the denominator) first captures the level of noise (variance) per y state, and averages over the states. Clearly the SNR can be reduced by either reducing the signal or increasing the noise. Both solutions are investigated in this manuscript. Precisely, the SNR is defined as:

$$\hat{\mathrm{SNR}}(t) = \frac{\hat{var}_y(\hat{\mathrm{E}}_i(\mathsf{L}_y^{i,t}))}{\hat{\mathrm{E}}_y(\hat{var}_i(\mathsf{L}_y^{i,t}))}.$$

It is important to highlight that as the univariate SNR and DPA attacks utilize some simplifying statistical assumptions regarding the leakage distribution, a verification step was performed. We cross-checked the chosen POIs and examined linear-regression models and the IT metrics described below vs. time (whereas the IT metrics utilize minimal statistical assumptions on the leakage). We found the results for the sake of (qualitatively) detecting POIs to be almost always consistent. As will be clear next, a different conclusion will hold for the quantitative worst-case estimation of complex leakages. Yet, since the computation of IT metrics is more resource-consuming and slower (per time sample), using the (simpler) SNR to find POIs served as a valuable speedup.

2) Information Theoretic Evaluation - Bounding the Security Level with IT metrics: In order to evaluate the security of our designs, we utilize a well known information theoretic (IT) metric, namely the Mutual Information (MI) [14], [32]:

$$\mathrm{MI}(Y; \boldsymbol{L}_Y) = \mathrm{H}(\mathrm{Y}) - \sum_{y \in \mathrm{Y}} \Pr[y] \cdot \sum_{l \in \boldsymbol{L}_\mathrm{Y}} \mathsf{f}_{ch}[l|y] \cdot log_2(\mathsf{f}_{ch}[y|l]), \tag{1}$$

where $\mathrm{H}[\mathrm{Y}]$ is the entropy of the sensitive variable Y and $\mathsf{f}_{ch}[l|y]$ is the conditional PDF of the leakage l from a chip (*ch*), given the secret manipulation of y. As a result, $\mathsf{f}_{ch}[y|l]$ can be computed by Bayes' theorem. Note that in case a randomizing countermeasure is used, $\mathsf{f}_{ch}[l|y]$ is then written as a mixture. For example, assuming a Gaussian noise, it would be worth $\mathsf{f}_{ch}[l|y_i] = \sum_{r \in \mathrm{R}} N(l|y_i, r, \sigma_n^2)$, where R represents the set of all possible randomizer states. As explained and demonstrated in [14], [32], the MI metric quantifies how much can be learned on Y from the leakage $\boldsymbol{L}_\mathrm{Y}$, and determines the worst case attack complexity. Yet, concretely, this quantity cannot be directly computed since the true chip distribution $\mathsf{f}_{ch}[l|y]$ is unknown and can only be estimated. As a result, we use two alternative quantities, the Perceived Information (PI) and Hypothetical Information (HI). As demonstrated in [20], if based on a non-parametric estimation of the leakage function (i.e. a non-parametric model), they serve as lower and upper bounds to the MI. Precisely:

$$\hat{\mathrm{PI}}(Y; \boldsymbol{L}_Y) = \mathrm{H}(\mathrm{Y}) - \sum_{y \in \mathrm{Y}} \Pr[y] \cdot \sum_{l \in \boldsymbol{L}_\mathrm{Y}} \hat{\mathsf{f}}_{ch}[l|y] \cdot log_2(\tilde{\mathsf{f}}_{mo}[y|l]), \tag{2}$$

$$\hat{\mathrm{HI}}(Y; \boldsymbol{L}_Y) = \mathrm{H}(\mathrm{Y}) - \sum_{y \in \mathrm{Y}} \Pr[y] \cdot \sum_{l \in \boldsymbol{L}_\mathrm{Y}} \tilde{\mathsf{f}}_{mo}[l|y] \cdot log_2(\tilde{\mathsf{f}}_{mo}[y|l]). \tag{3}$$

In these equations, $\hat{\mathsf{f}}_{ch}[l|y_i]$ represents the samples of the true distributions collected during the test phase (intuitively corresponding to the empirical distribution) while $\tilde{\mathsf{f}}_{mo}[y_i|l]$ is the PDF corresponding to the adversarial model (*mo*), estimated during a profiling phase. Roughly, the PI is the amount of information that can be extracted from a leaking device thanks to an estimated statistical model, possibly biased due to estimation and assumption errors. It is computed by testing the model against fresh samples from the chip distribution. The HI is the amount of information that would be extracted from an hypothetical device exactly following the model distribution. It is computed by testing the model against itself. Roughly, the simplest solution to estimate these metrics is sampling based estimation, using cross-validation for the PI and without cross-validation for the HI. For both metrics, we next considered histogram-based models allowing us to lower and upper bound the worst-case MI.

D. Masking after side-channel signal hiding

As a side remark in this section, and as a motivation for the proposed side-channel signal hiding solutions, we examine the case of a masked implementation. In this context, let us assume that we do not try to build perfectly balanced circuits, but rather that our goal is to reduce the side-channel leakage at limited cost. Masking is a well understood countermeasure theoretically against Side-Channel Attacks. It works by splitting any sensitive variable (s) of an implementation into d shares, where d-1 of those are drawn at random and the d^{th} share complies with $s = s_1 \oplus s_2 \oplus \cdots \oplus s_d$, where $\oplus$ is a group addition operation in a finite-field (XOR in the binary case). Computations should be performed on the shared variables only. Under the assumption that the leakages produced during the manipulation of the shares can be written as a linear

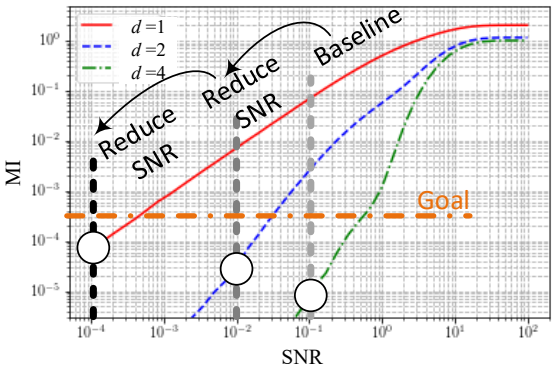


Fig. 2: MI of a masked design vs. SNR, illustrating the required number of shares in order to reach a given security goal for devices with different SNR values.

function of the underlying shares leakages, which is frequently referred to as masking's *independence assumption*, it amplifies the noise in the leakages, and therefore the implementation security. In essence, this amplification is obtained by forcing the adversary to estimate a higher-order statistical moment of the leakage distribution: a task of which the data complexity grows exponentially in the number of shares [2], [14], [33]. The lowest key-dependent moment of the leakage is denoted as the (statistical) *security order*.

Informally, masking can be viewed as a noise amplification. Hence, increasing the noise, reducing the signal (or doing both) with side-channel signal hiding before increasing the number of shares is in general a good approach which may lead to very high security levels at lower cost than without hiding. This claim is typically illustrated by Figure 2, where we show the amount of leakage after masking (on the Y axis, measured with a mutual information metric), in function of the level of physical signal and noise in the measurements, here measured with the SNR. We consider an unprotected implementation and two masked ones, with $d = 2$ and $d = 4$ with sharing over a byte. The leakages were simulated with a Hamming weight leakage model and additive Gaussian noise.

As we already know from theory, for an unmasked design, we follow quite nicely a linear trend in log-log scale with such a model. That is, with d=1 reducing the SNR by an order of magnitude leads to a similar MI reduction (which is inversely proportional to the worst-case security level [14]). Next, as the number of shares d increases, the negative slope of the MI curves also increases. Therefore, for a given baseline SNR in our design, to reach a specific security-goal (i.e. MI value) we should increase d as required. This typically induces substantial (quadratic) overheads in energy and area. With the "*ask less, get more*" approach, we demonstrate that with a substantially smaller cost, we can reduce the SNR in a way that will require smaller d values to reach a target security level, hence reducing the associated quadratic overheads. This is illustrated on the figure with vertical lines which represent SNR reduction with methods as will be demonstrated next.

II. Low-Cost Side-Channel Signal Hiding

In this section, we first detail on the proposed approach. We follow with a guided analysis over Analog SPICE simulations while stacking up low cost security measures. First, we demonstrate the extent of standard signal reduction techniques and then we embed amplitude randomization circuitry.

As indicated before, our aim is to utilize fully standard, EDA compatible, and low cost signal reduction techniques. Our "*ask less, get more*" approach accepts that the utilized techniques are not perfect and suffer from shortcomings as described above. Then, to solve some of their associated inherent problems (i.e. though reduced, information still exists), low cost noise generation techniques which are *local* are embedded, further reducing the information content.[1]

A. The Extent of Signal-Reduction

We start by examining the leakage characteristics of our WDDL designs, and evaluating their security level. For this purpose, and as a running example for this section, we have implemented a tower field AES Sbox in HDL.

We have imported this design into an analog Spice simulation environment after synthesis into an ASIC 65nm technology library, performing place-and-route (with Cadence Genus and Innovus tools, respectively). We followed by asserting all possible input vectors transitions to the Sbox and measuring the power supply current. In addition, we have transformed the same CMOS HDL representation to a WDDL one (the exact flow is discussed in Appendix A), and did the same.

The upper plots of Fig. 3 illustrate the superimposed currents of the AES Sbox over time of all possible inputs transitions. The left plot is for the standard CMOS design and the right plot for the WDDL design in the evaluation phase [1] (i.e. after the RTZ spacer). The lower plot of Fig. 3 illustrates the leakage distribution at the computed POI as

[1] The locality driven approach is motivated by the need to make it hard for advanced adversaries (e.g. taking advantage of localized EM techniques).

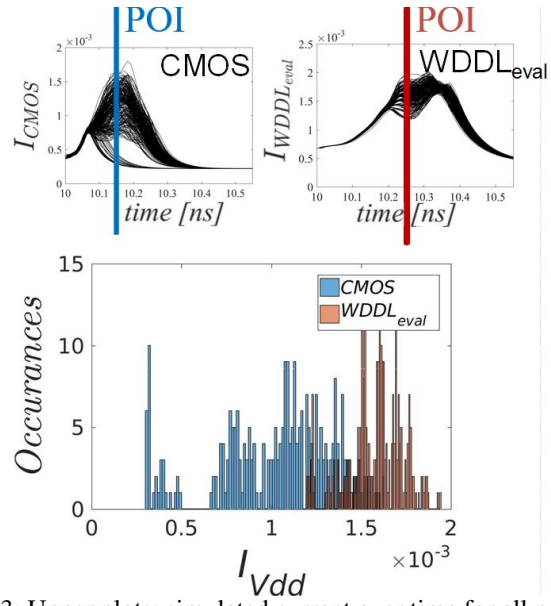


Fig. 3: Upper plots: simulated current over time for all possible inputs of a CMOS (left) and a WDDL (right) design (in the *evaluation* phase), and Lower plot: the corresponding currents distribution in a maximum SNR POI.

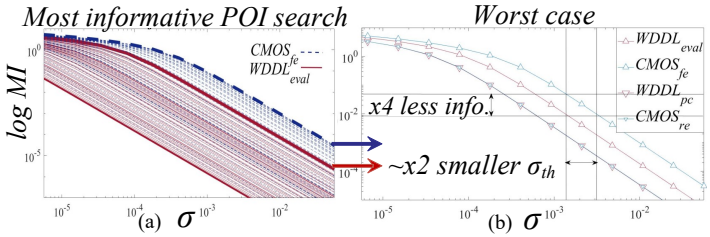


Fig. 4: MI computed on the simulated currents: (a) over all time samples and as a function of the noise level for CMOS and WDDL, and (b) on the POI which maximize the MI.

indicated on the upper plots. As expected and can be observed from the figure thanks to visual inspection, the WDDL leakage distribution is more condensed than the CMOS one, yet still leads to exploitable leakage. This is exactly the consequence of the previously mentioned imperfections: even, if we were to remove place and route steps, which are the hardest to balance [21], using standard cells primitives will still leak.

We followed by computing information theoretic metrics from the simulated data as shown in Fig. 4 merely to demonstrate these claims. In this simulated section, the noise is assumed to be additive and Gaussian with zero mean. In our simulated analysis the noise standard variation σ_n is a parameter which reflects the true physical noise of the design. Note that as we are in a simulated setting, we have an exact knowledge of the leakage distribution so we can directly compute the MI and do not need the HI/PI bounds.

The MI(t) for all the time samples vs. the simulated noise level is shown in Fig. 4(a): in red curves the WDDL design and in blue curves the CMOS design. The most informative POI matched the one computed by the SNR. The two corresponding curves (of maximum information) for CMOS and WDDL are shown separately in Fig. 4(b) for the precharge and evaluation phases, and for the high and low phases of the clock for the CMOS design. It is possible to see that in the worst case, i.e. the evaluation phase of the WDDL design, the informativeness of the leakage is ≈ 4 times smaller than that or the worst case for the CMOS design, i.e. the leakage in the falling-edge of the clock. Alternatively, to achieve the same level of security (i.e. informativeness of the leakage), it is sufficient to have a ≈ 2 times smaller standard deviation of the noise for the WDDL design.

B. Building Up with Low-Cost Noise Generation

To achieve amplitude noise generation, we first need a mechanism to inject amplitude noise into the leakage. For this purpose, we make use of standard cell Power Gating (PG) libraries which provide fully characterized pMOS devices which can be placed on the power network of the system. For each of our modules under investigation, we place a set of PG cells (or PGs) in parallel. One of them is always-on (with a logical '0' input) and the rest are asserted with fresh randomness in each clock cycle. This structure is illustrated in Fig. 5. On the figure, from left to right, the first (left) setting is the DUT in a noiseless SPICE simulation environment, the second (middle) setting is an abstraction with a randomized power network resistance and the third (right) is the instantiation of PGs to

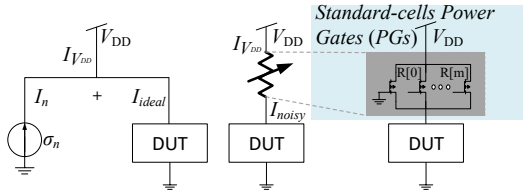


Fig. 5: Schematic representation of our design: (a) a noiseless simulation environment for our DUT (b) a DUT with power grid randomized resistance and (c) a schematic instantiation of such a randomizer with power gating pMOS devices.

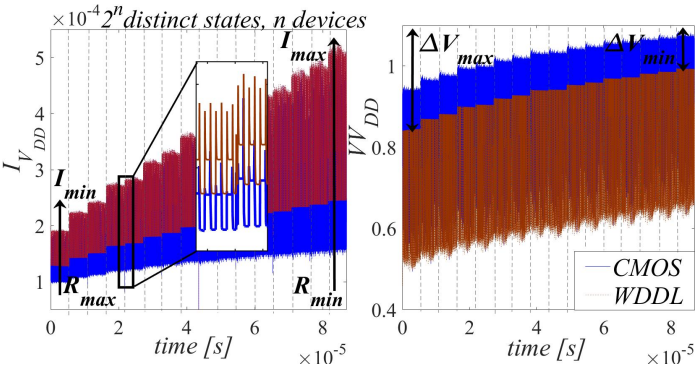


Fig. 6: Transient analysis over time while triggering all possible input vectors transitions and looping over the randomizer states (2^n). **Left:** maximum and minimum current. **Right:** minimum and maximum voltage drop.

achieve it. Both CMOS and WDDL designs were embedded with such mechanisms and simulated. A similar, but more degenerated approach was taken in [34], [35].

Fig. 6 shows the results from a transient analysis simulation of the current and voltage drop over time, while triggering all possible (2^n) randomizer states serially, with n, the number of PGs not tied to '0' set to 4. For each state, all possible input vectors transitions were serially asserted. On the left plot, the current drawn for the CMOS (blue) and WDDL (red) design is shown. The maximum and minimum currents corresponding with the equivalent minimum and maximum power grid resistance are denoted. On the right plot, the minimum and maximum power supply voltage drops are denoted (i.e. the distances from the 1.1V nominal supply voltage). It is possible to see that each PG state corresponds to a different equivalent resistance and that the distribution of these average equivalent resistances is not perfectly uniform.

The second part of our proposed noise generation consists of a fully standard and low area random sequence generation. For that purpose, we borrow ideas from [25], [26]. The mechanism we use is sampling with a slow oscillator (clk signal) an ultra fast oscillator, which accumulated phase noise (jitter). After capturing these noisy samples, they are fed into an n-bit shift register and then fed to our PGs as illustrated in fig. 7.

We denote by R^j_{norm} the average resistance of the power network, where j indexes the randomizer state $j \in \{1, .., 2^n\}$. Ideally, we would like to achieve a uniform distribution of R^j_{norm}, but as in the previous section, this is a physically hard to fulfill goal. To illustrate some tradeoffs and discuss how well it is possible to optimize the PGs dimensions to

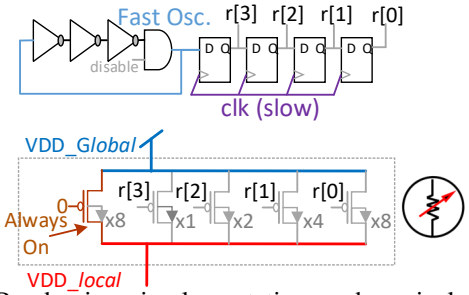


Fig. 7: Randomizer implementation and equivalent power gates sizes (in terms of W_{min}/L_{min} transistors ratio).

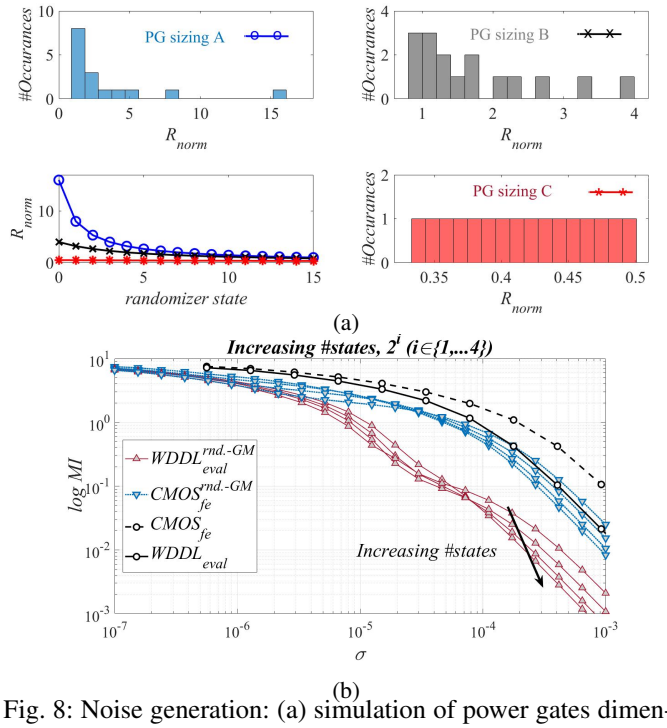


Fig. 8: Noise generation: (a) simulation of power gates dimensions from *PG Sizing A* to *C* where the equivalent-resistance distribution is uniform, (b) security evaluation (MI) while increasing the number of states of the randomizer.

uniformly distribute R^j_{norm}, we demonstrate several different PG transistors sizing examples, referring next to Fig 8a. With a clockwise orientation, from top left to bottom right, three PGs device sizing approaches are demonstrated. In the first, we set equal device sizing for all PGs. As the PGs are connected in parallel, there exist many states which will conclude in the same R_{norm}, concluding in a biased distribution. In order to get a unique R_{norm} per-state, each PG should clearly be sized differently. However, the limiting factors are in fact the limited set of dimensions we typically have in a standard PGs library, and the fact that the resistance of the *always on* PG biases R_{norm}. This becomes a real challenge to find a nice set of dimensions which both limit the span, $min_j(R^j_{norm})$ to $max_j(R^j_{norm})$, and the spread of R^j_{norm} in the range (i.e. uniformity). The lower right plot illustrates the distribution we were able to achieve (with this 65nm technology, from a specific vendor). The lower left plot shows the big span of optimizations designers have. An important optimization which should be considered during design stages is that a too-small spread of the resistances is clearly an issue, and on the opposite side if the spread is too large and each of the data-dependent "lobes" is stand-along, an adversary can classify and exclude all lobes but one, putting us again back to the starting point. Therefore, we would like to maximize overlap between data-dependent spreads with the randomization circuitry, while keeping it's distribution as close to uniform as possible. Specifically, for this work the final sizing factors of our PG devices, in terms of W_{min}/L_{min} dimensions were x{8,1,2,4,8} as denoted on Fig. 7.

Fig 8b shows the computed MI (in log scale) vs. the noise level for the CMOS and WDDL designs embedded with the proposed randomizer. The black dashed and smooth curves are the CMOS and WDDL curves without the regulator from the previous subsection for comparison. As expected, our distributions are statistical mixtures (with 2^n elements), hence, we fit a Gaussian mixture model for the simulated leakages. In the figure, we show an analysis of each of the designs while increasing the #states in the randomizer (by enforcing a '0' input to one PG at a time). With a careful investigation, it is possible to see that: (1) just adding the randomizer already increases the security level — this is expected as it decouples the power grid and low pass filters; (2) as the number of states in the randomizer increases, the security in high noise regimes increases (while for extremely low noise levels, it does not have much impact). Interestingly, in any case the MI achieved by the WDDL design is more than one order of magnitude smaller than its randomized CMOS counterpart, and 1.5 order of magnitude smaller than the *clean* CMOS design (for most realistic noise levels). It further illustrate how even a slightly biased distribution of the randomizer provide significant improvement.

C. Architecture and Locality

The chosen design manufactured and tested in this work is an AES encryption core in a parallel loop architecture (iterations over 16 parallel Sboxes). Each AES 8-bit Sbox is efficiently implemented in HDL, using tower Galois-field (GF) arithmetic (precisely, we used GF($(2^4)^2$)). After synthesis towards a standard cell library, the process of transforming our CMOS gate level HDL to a WDDL gate level representation takes place. Following this step, we embed a synthesized digital version of our noise generators. In this respect, and besides our ultra small and low cost design goals, we also aimed at keeping this part local. For that purpose, each generator was assigned to a very small partition in the design, as illustrated in Fig. 9. Overall, the ingredients of our signal hiding solution are (1) signal reduction, (2) noise generation, (3) locality, all of this achieved using only (4) fully standard tools, process flows and libraries.

To be able to embed our noise generators locally in the physical design, we first divide the entire core (as illustrated in 10(a)) into small partitions which are affected by at most 8-bit operands/secret variables. This makes a natural choice for Sboxes, muxes, registers and the key-scheduler. The only exception was the Mixcolumns hardware which was naturally divided into to 32-bit partitions, concluding in 75 PDs. Considering Fig. 10(b), after importing the entire design into

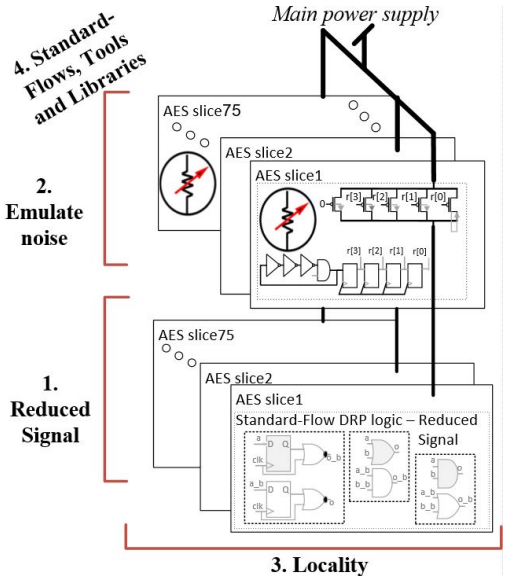


Fig. 9: The ingredients we make use of in our proposed approach: signal-reduction, noise generation, locality driven partitioning with standard tools, flows and libraries.

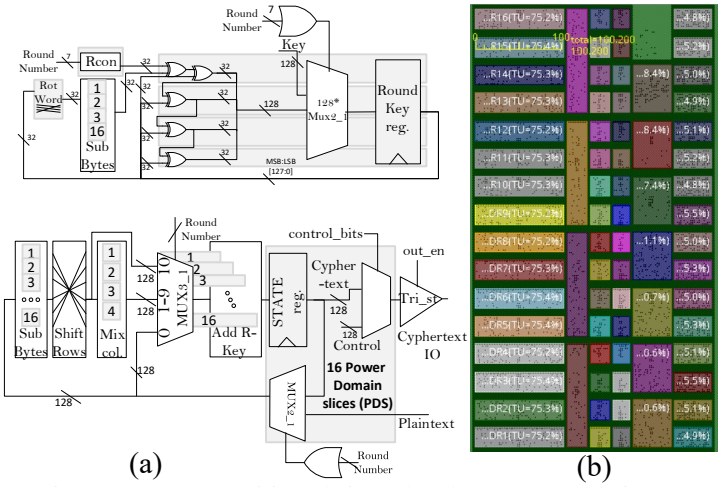


Fig. 10: AES partitioned into local Power Domains: (a) schematic representation, (b) physical PDs placement.

a physical implementation tool (P&R), each partition was placed in a separate Power Domain (PD) with its own noise generation circuitry. As shown in the figure, it is possible to nicely pack PDs with minimal spacing constraints. The power grid supplying the array of randomizers is mashed across the entire area of the core and each of our cores power line is supplied by a separate power Analog IO. The average area of a PD in this technology is $30um^2$. In state-of-the-art technologies such approach can conclude in $5-10um^2$. Note that in Section IV we relate more specifically to area overhead.

Architectures and naming conventions: In our ASIC chip, we have embedded three full architectures with some knobs to disable the noise generation circuitry:

- **CMOS:** a reference rolled CMOS architecture.
- **R-CM$_{dis}$:** A randomized CMOS architecture. The lowercase dis denotes that the randomizer circuit is disabled.
- **R-CM:** A randomized CMOS architecture where the randomizer is enabled.
- **R-DR$_{dis}$:** A randomized WDDL architecture. The lowercase dis denotes that the randomizer circuit is disabled.
- **R-DR:** A randomized WDDL architecture where the randomizer is enabled.

III. Post Fabrication Security Analysis

In this experimental part of the manuscript, we first layout the baseline conditions of our evaluation environment, then discuss the evaluation tools we used, and we conclude with the description of our experimental results.

The architectures from above were designed and taped out on a 65nm process. The die photograph is shown in Fig. 11(a). The equivalent physical layout view from Innovus is shown in Fig. 11(b). The yellow encircled areas are the designs presented in this work whereas the grayed designs are unrelated projects on the same die. The design inhabits an SPI interface and control/reset/debug mechanisms. Fig. 11(c) zooms in to the lower section of the R-DR architecture, and we take a closer view to one of the PDs on the right. The area in red is the entire area needed to place the generator circuitry (oscillator, four registers and PGs). It is possible to see the area ratio and to understand that the added cost is small. On top of this region, a Metal-4 vertical line passes which connects to the power grid mesh of the core.

We have constructed tailored measurement boards for the chip with measurement access for each core independently. The measurement is possible to be carried out directly (through e.g. a passive current probe without an amplifier), or after an on-board preamplifier. Architectures under investigation were clocked with a 6MHz clock frequency for a conservative security evaluation which nicely captures all digital activities without leakage overlap of different computations (i.e. reducing the noise). The measurement board is depicted in Fig. 18 in Appendix A. Importantly, special care was taken in the design of our board to reduce noise and parasitics. For example, grounding planes of the different regulators were separated.

The device was powered from an on-board linearly regulated low noise power supply, while its current absorption has been measured through a 0.1Ω precision resistor. To improve the SNR ratio, we have used an on-board x10 gain preamplifier, placed on a properly designed ground plane to ensure better signal integrity. Current traces have been collected by a Picoscope 5244B running at 500MS/s, providing 12 bits of amplitude resolution. For each round (clock cycle) we had $\sim$84 leakage samples and $\sim$1200 for a full trace.

Starting with a visual inspection of the leakage traces, Fig. 12(a-c) illustrate the mean leakage, quantized over the 12 oscilloscope bits, over time samples. It is possible to see the 10 rounds of an AES-128 for the CMOS design (Fig. 12(a)): each large and small current peak matches a clock rising and falling edge in a round. For the WDDL design, shown in Fig. 12(c), it is possible to see large current peaks both in the rising and the falling clock edges (precharge and evaluation phases).

We follow with evaluating the resulting SNRs of the different architectures and finding the best POIs. The SNR(t) of the CMOS, R-CM and the R-DR designs are shown in Fig. 12(d-f). While targeting one Sbox output and computing the

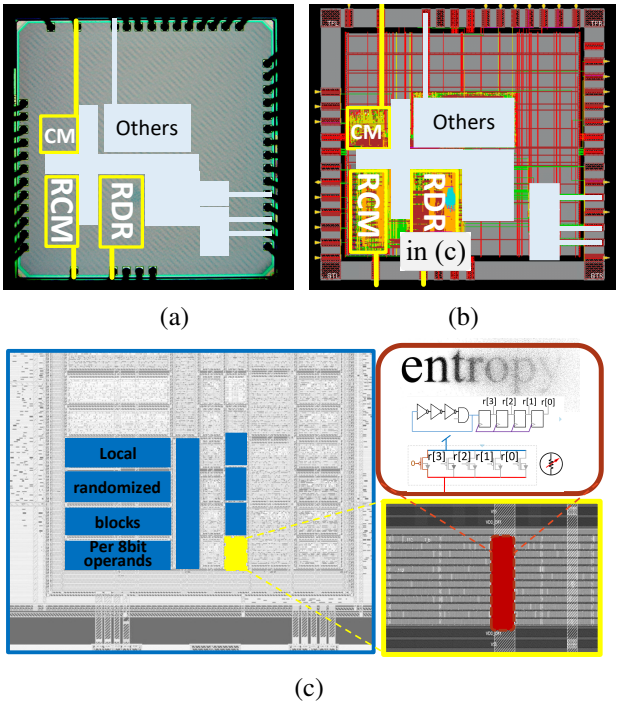


Fig. 11: Manufactured device: (a) chip micro photograph (b) corresponding (pre-fabrication) view (c) zoomed-in view of partitioned region and randomizer.

SNR, it is visually evident that both the rising and falling edges of the clock carry information. We have computed the SNR and correlation coefficient (resulting from a CPA attack with a Hamming weight model) and found the best POIs per architecture. The blue (red) marked points provide the larger (in absolute value) SNR (correlation) for $10*10^6$ traces. As expected, the POIs of these two collide when the actual leakage does fit well enough the model, and sometimes do not (where the actual leakage does not fit well enough the model). We next use theSNR derived POIs. It is further shown that:

- The SNR of the {CMOS,R-CM,R-DR} designs are in the scale of $\{10^{-3}, 10^{-4}, 10^{-5}\}$, respectively. A closer look vs. the number of measurements follows next.
- With the R-DR design analyzed in Fig. 12(f), the signal becomes close to the noise level (i.e. SNR of further away rounds) for the amount of measured traces.

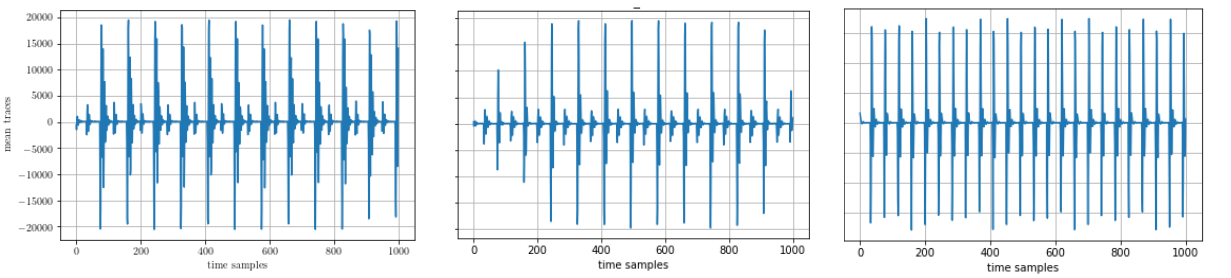
To conclude the investigation of the SNR and to draw some more general conclusions, we collected $100 * 10^6$ traces and computed the SNR vs. the number of traces (in the optimal POI) for all designs. The resulting plots are shown in Fig. 13. In this figure, both axes are in log scale and we can observe:

- The CMOS SNR stabilizes at $5 * 10^{-3}$.
- The R-CM$_{dis}$ SNR stabilizes at $2 * 10^{-3}$ — we assume this reduction is the resulting effect of the power supply decoupling through the disabled PGs mechanism.
- The R-CM SNR stabilizes at $5.5*10^{-4}$ — this reduction corresponds to the intended effect of the noise generators.
- The R-DR$_{dis}$ SNR stabilizes at $3 * 10^{-4}$ — this corresponds to the effect of the signal reduction (WDDL) while the noise generation is disabled.
- The R-DR SNR stabilizes at $5*10^{-5}$ — we assume this reduction is due to the joint effect of the noise generators and the signal reduction.
- As the level of security increases, the number of traces required to set on a stable SNR value rapidly increases, from $2 * 10^6$ traces for the CMOS design to $70 * 10^6$ traces for the R-DR design, which leads to a significant measurement effort.

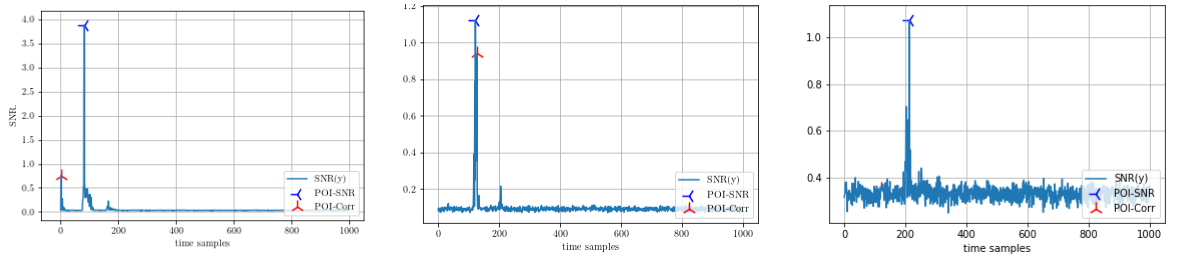
To better understand the effect of the evaluated mechanisms, we also investigated the leakage distribution of the different designs. Fig. 14(a) shows the leakage probability distribution of the R-CM and R-CM$_{dis}$ designs (quantized with 12 and 16 bits of the oscilloscope when needed). As expected, the R-CM distribution is much wider due to the randomizer. Fig. 14(b) shows the leakage distribution of the R-DR and R-DR$_{dis}$ designs. As expected, the R-DR$_{dis}$ distribution is more narrow due to the signal reduction, and the R-DR distribution is much wider due to the randomizer. However, clearly the randomized distributions are biased. This is due to physical defaults leading to a suboptimal distribution of R^j_{norm} (the correction of which is not simple post-P&R and therefore not an "ask less" goal).

To conclude the post-fabrication evaluation phase, we follow with computing the IT metrics on the collected traces, namely, the PI and the HI. Fig. 15 shows in solid and dashed lines the HI and PI (respectively) of all evaluated designs vs. the number of profiling traces @POI. A nice observation from the figure is that after $7*10^6$ traces, all the PI and HI values converge (note that the Y-axis is in log-scale). The interval between the PI to HI value of a given architecture reflects the uncertainty we have regarding the worst-case leakage (i.e. the tightness of the bound). In practice, in our reduced noise experimental setup it is quite small for all designs. Two more important observations are that: (1) comparatively, the asymptotic values of the curves of the different designs follow the trends already observed by the SNR vs. #samples (2) the #profiling-traces required to stabilize the PI value gradually increases with the security level as expected. Regarding the quantitative values from the figure, the CMOS design leaks 10^{-1} bits of information on the 8-bit subkey. The R-CM$_{dis}$ design leaks $7*10^{-2}$ bits of information which is not a very significant gain in security, presumably due to power lines decoupling. The R-CM design leaks $3.5*10^{-2}$ bits of information, which is only a factor of two less due to the noise generation and is not sufficient to hide the large information leakage of pure CMOS-based architectures. The R-DR$_{dis}$ design leaks $2 * 10^{-2}$ bits of information, which is the sole contribution of the embedded signal reduction. In turn, it highlights the 'imperfect' nature of our WDDL primitives. Finally, the R-DR design leaks $3 * 10^{-3}$ bits of information, which demonstrate a considerable reduction in information-leakage (similarly to the SNR trends).

One important observation related to this final plot is that the MI values observed are significantly higher than the SNR values observed in the Figure 13. This difference can be explained by the shape of the distributions in Figure 14 which are not Gaussian. So this result shows that in the context of signal hiding countermeasures involving a randomization mechanism, security evaluations based on the SNR (which relies on a Gaussian assumption) can overstate the security of



(a) CMOS rolled (*mean*) (b) R-CM, enabled (*mean*) (c) R-DR, enabled (*mean*)



(d) CMOS rolled (*SNR*), **y scale: $\mathbf{x10^{-3}}$** (e) R-CM enabled (*SNR*), **$\mathbf{x10^{-4}}$** (f) R-DR enabled (*SNR*), **$\mathbf{x10^{-5}}$**

Fig. 12: Measured leakages and corresponding SNR/correlation: (a-c) leakages measured from our ASIC chip of {CMOS,R-CM,R-DR} designs, respectively; (d-f) computed SNR/correlation of the {CMOS,R-CM,R-DR} designs, respectively.

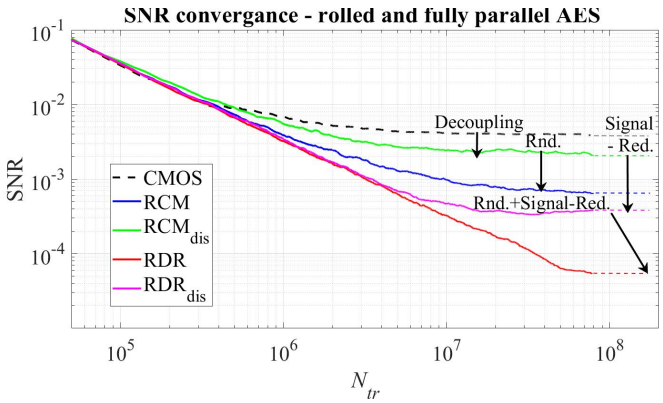
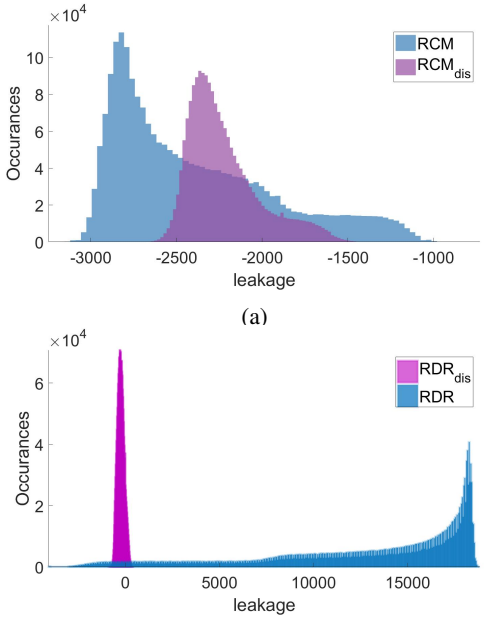


Fig. 13: SNR(t=POI) vs. the #traces for all architectures.



(a) (b)

Fig. 14: Leakage distribution of (a) R-CM and R-CM$_{dis}$ (quantization with 12-bits ADC) and, (b) R-DR and R-DR$_{dis}$ (quantization with 16-bits ADC).

an implementation, and therefore worst-case bounds for the MI are actually handy to reflect the actual security level. Note that despite the MI values are larger than the SNR ones, the next section will show that the low cost MI reductions that we reach (compared to a CMOS design) can have high practical relevance in the context of masking.

As a final note, we mention that we also wanted to understand how sensitive our security evaluation is to variations of environmental factors (e.g. voltage and temperature). A detailed examination of our device with a monitored power source inside an oven was performed and the results are given in Appendix B. The main observation is that the CMOS and R-CM$_{dis}$ designs are more sensitive to such variations than the R-CM, R-DR and R-DR$_{dis}$ designs. That is, they leak significantly more in extreme cases. By contrast, the noise-generation circuitry 'protects' the R-CM and R-DR designs from leaking more information in such extreme cases, and the R-DR design has especially limited sensitivity thanks to the anyway nicely reduced signal (details in appendix B).

IV. Cost vs. Security Tradeoff

We conclude this paper by discussing the overhead factors associated with the proposed architectures. The main goal is to highlight that the revisited hiding mechanisms support considerable security gains at limited implementation cost.

Starting from a baseline comparison in nominal conditions (room temperature, nominal supply voltage and 6MHz of clock frequency), Table I lists MI metrics from the previous section (approximated as the mid value between the estimated HI and

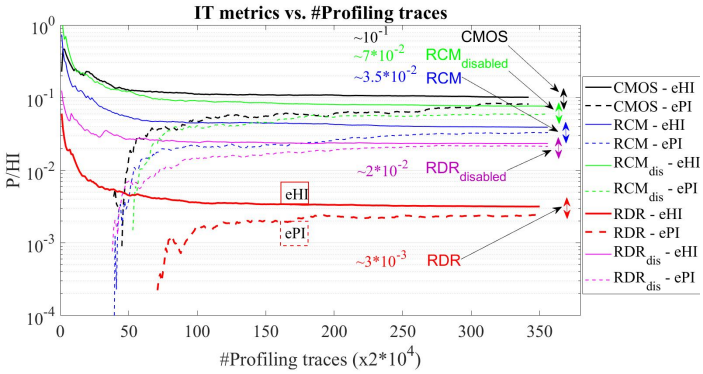


Fig. 15: P/HI (@POI) after a 10-fold cross-validation (with 90% train -10% test sets), for the: CMOS, R-CM-disabled, R-CM, R-DR-disabled, and the R-DR designs.

65nm	Area	Energy @6MHz*	Max. Freq.	MI
CMOS	150um X 170um (X1)	400 nJ (X1)	400MHz**	10^{-1}
R-CMOS	120um X 300um (X1.41)	520 nJ (X1.3)	340MHz	$3.5 * 10^{-2}$
R-DR	**160um X 300um (X1.88)**	**910 nJ (X2.27)**	**170MHz**	$3 * 10^{-3}$

TABLE I: Cost and security metrics for the different AES architectures in nominal conditions. * of 10 rounds, ** note that our aim was not to push for maximum performance, but rather a straight forward synthesis

#shares	Area (kGE)	Area overhead	#shares	Area (kGE)	Area overhead
2	6	1	10	67	11.1
3	10	1.6	11	80	13.3
4	19	3.2	12	95	15.8
5	24	4	13	105	17.5
6	30	5	14	122	20.3
7	38	6.3	15	138	23.3
8	47	7.8	16	157	26.1
9	60	10			

TABLE II: Approximate area overhead for an AES DOM [36].

PI bounds), and several cost metrics (area utilization, energy per encryption, and maximum frequency).

As clear from the table, the area utilization of the R-CM (resp. R-DR) design is only x1.4 (resp. x1.9) the one of unprotected CMOS implementation. This factor implies that the extra cost of power domains separation and noise-generation embedding is 41% from the total area, while the cost of the signal reduction (i.e. the reminder) is 50%. Note that metrics such Gate-Equivalents or gate count are not representative as we are interested in the entire area cost (spacing, routing, isolations etc.). Regarding the (average) energy per encryption, we see that the noise generation cost is quite small: 20% for the R-CM and 11% for the R-DR design. The reason for the difference is that the power drop over the randomizer depends on the actual current load of the logic and parasitic network capacitances. These clearly vary from one block to another. On the other hand, the cost associated with the WDDL signal reduction is a bit less than x2 in terms of energy. This is expected due to the precharge RTZ mechanism and the logic duplication of some of the internal gates. Overall, a rough area and energy footprint of x2 for such mechanisms is quite minimal considering the described security gains.

We follow with a cost vs. security tradeoff discussion. For this purpose, let us start with an example and consider an exemplary target security level corresponding to MI= $2^{-32} \approx 10^{-10}$. As per [14], it means that at least $2^{32} \approx 10^{10}$ traces are needed to break the implementation (that is, the security level is inversely proportional to the MI).

Our goal is to qualitatively evaluate what is the added value of the proposed designs prior to masking. For this purpose, we first recall the area utilization overheads available for the Domain Oriented Masking scheme in [36]. The authors have implemented an AES with a quite standard hardware masking scheme (reflective of the state-of-the-art). In Table II, we provide the area utilization (in terms of gate equivalents, kGE) and the area overheads while increasing the number of shares (#shares). From Table I, we know that MI(CMOS)=10^{-1}, MI(R-CM)=$3.5*10^{-2}$ and MI(R-DR)=$3*10^{-3}$. Knowing that the MI of a masked implementations decreases as MI^d [14], it theoretically implies that for CMOS, we would already need 10 shares to reach this low security target, meaning an area overheads factor of 11. For R-CM, this would be reduced to 7 shares, implying an area overhead factor of 6.3*1.4=8.82 (corresponding to the masking overheads times the hiding overheads). For R-DR, this would be further reduced to only 4 shares, and an overheads factor of 3.2*1.88= 6.

We next illustrate this tradeoff in a larger scale thanks to Fig. 16, where the same analysis is repeated for various security targets. The figure shows the target MI (in log scale) on the left Y axis and the total area overhead factors on the right Y axis, versus the #shares (on the X axis). The curves with a triangle marker correspond to the CMOS design masked with DOM (red curve for security level and blue for area overhead); the square marked curves correspond to the R-CM design masked with DOM; the circle marked curves correspond to the R-DR design with the DOM masking. The horizontal gray-scale lines indicate exemplary security targets. From them, it is possible to search for the crossing of the CMOS, R-CM or R-DR security curves (in red), and from these intersections, to find the appropriate area overheads and compare. Following this analysis, we observe that R-DR is by far the best in terms of cost vs. security, leading to area savings of up-to 20 depending on the security level. In general, we also conclude that side-channel signal hiding becomes increasingly useful as the target security level increases.

We acknowledge that the Noise level and SNR depend on the type of masking, the implementation technology and the architecture (e.g. serializing). However, this final section is aimed at showing trends and clarify the potential impact of our design. In general, regarding our round-based design and the serialized architecture of DOM, the area trends we indicate represent a worst-case as Sboxes-parallel architectures get more expensive with d. As a final remark, we mention that in addition to these attractive cost factors, an equally important aspect of our proposal is that the design effort is negligible: we utilize fully standard flows and tools (see Appendix A).

V. Conclusions

We have implemented in a tailored combination two natural side-channel signal hiding techniques with the aim of using

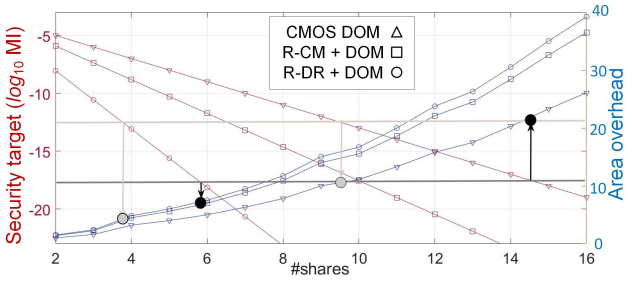


Fig. 16: Area overheads & security target for various DOM masked designs.

their joint instance as a building block. Namely, WDDL logic and a novel simple, local, security-scalable noise generation engine. Both demonstrate concrete SNR and MI reduction. It is demonstrate how simple evaluation metrics are not sufficient to capture complex leakage measurements of systems designed for side-channel immunity; and provide theoretical area/energy gains that can be obtained by utilizing such instances in combination with masking. The proposed "*ask less, get more*" strategy enable considerable reduction of the masking order. The results are supported with theoretical analysis and exhaustive simulation and 65nm ASIC fabrication results with multiple and full AES encryption cores. We believe that this work provide good understanding and formulation of what we need to ask from different mechanisms to reduce cost and complexity to gain a certain security level; and that side-channel signal hiding revisited in this way is highly efficient. We believe that many research directions opens-up from this work including better improving the uniformity of the local randomizer, implementing combined systems with masking and improving on cost factors with other techniques.

APPENDIX A
THE PROPOSED AUTOMATED DESIGN FLOW

In this Appendix we elaborate on the proposed ASIC design flow. In Fig. 17a we illustrate a standard design flow steps from synthesis, through post synthesis timing, back-end design and sign off verification stages. Fig. 17b illustrated the design flow we modified to reach the required design. Starting from the synthesis stage, we restrict the standard cell library to a reduced set of cells, which we know easily how to transform to WDDL cells. I.e. Flip flops, ANDs and ORs of all input sizes and inverters. The inclusion of inverters is an artifact of using standard CMOS synthesizers which must accept such cells for synthesis. At later stages we eliminate those. Constraining the library set can be performed with standard library_groups directives of synthesis tools or by providing a dedicated and reduced .lib files. Following synthesis and once a netlist is generated we use a script to do necessary manipulations as illustrated in the gray cloud in Fig. 17b: (1) parse netlist; (2) duplicate all nets and appending an `_B` to their name to indicate an inverted net; (3) duplicate all logical cells while transforming ANDs to ORs and vice-versa and appending an `_B` to their instance names and input and output signals; (4) replacing all inverters cells with wire crossing; (5) duplicating all flip-flops and appending precharge

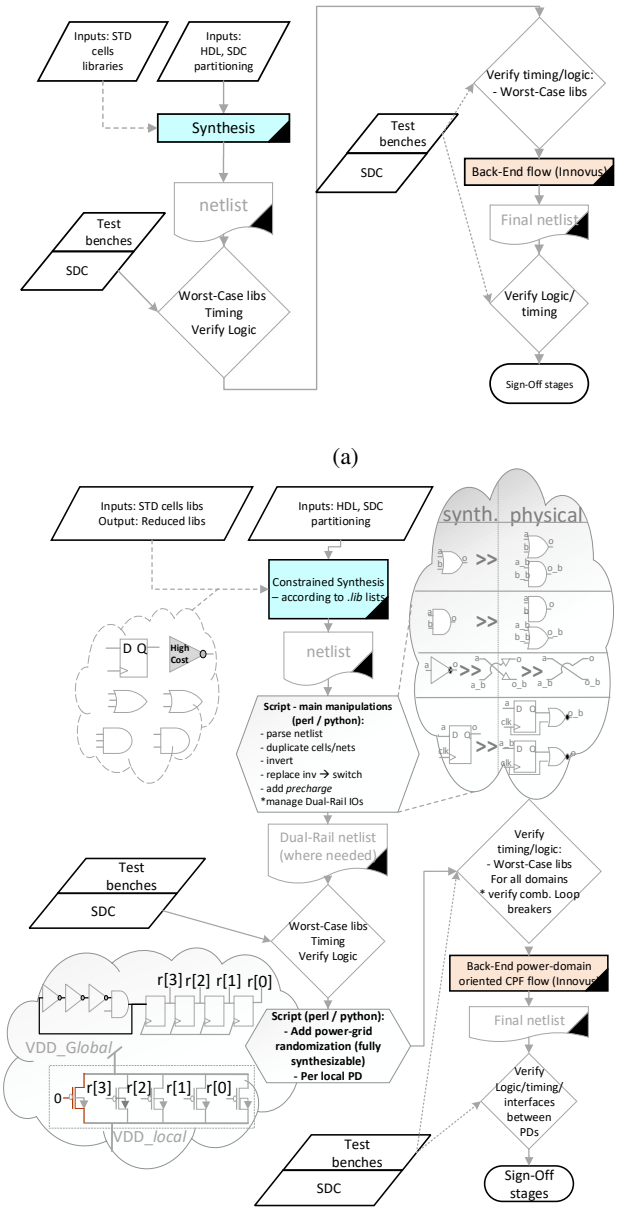


Fig. 17: EDA flows: (a) standard (b) proposed

mechanisms (NORs) at their outputs; (6) at design boundaries we manage IOs by duplicating input and output signals, inserting inverters and buffers as required and adding artificial load capacitances (as matched as possible). Specifically we used *perl* scripts to perform these stages quite elegantly and easily (it can be alternatively done by any other scripting language, e.g. *python*). We followed by performing timing and logical verification. To embed the randomization circuitry we synthesized it once using `dont_care`, `dont_touch` directives and then another *perl* script appended it to the power network signal of each of the modules (equivalently power domains) that were described in the design partitioning diagram from above. We follow by verifying timing again with the power gating libraries used. This later stage was performed while enforcing the timing modes to evaluate the

	V_{DD}=1.2V T=18°		V_{DD}=1.0V T=5°		V_{DD}=1.0V T=35°	
	SNR	MI	SNR	MI	SNR	MI
CMOS	$5 \cdot 10^{-3}$	10^{-1}	$3 \cdot 10^{-3}$	$8 \cdot 10^{-2}$	$4.5 \cdot 10^{-3}$	$9 \cdot 10^{-2}$
R-CM$_{dis}$	$2 \cdot 10^{-3}$	$7 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$4 \cdot 10^{-2}$	$1.7 \cdot 10^{-3}$	$6 \cdot 10^{-2}$
R-CM	$5.5 \cdot 10^{-4}$	$3.5 \cdot 10^{-2}$	$3 \cdot 10^{-4}$	$1 \cdot 10^{-2}$	$3.5 \cdot 10^{-4}$	$1.5 \cdot 10^{-2}$
R-DR$_{dis}$	$3 \cdot 10^{-4}$	$2 \cdot 10^{-2}$	$1.2 \cdot 10^{-4}$	$1 \cdot 10^{-2}$	$2.1 \cdot 10^{-4}$	$1.8 \cdot 10^{-2}$
R-DR	$5 \cdot 10^{-5}$	$3 \cdot 10^{-3}$	$2 \cdot 10^{-5}$	$9 \cdot 10^{-4}$	$1 \cdot 10^{-5}$	$8 \cdot 10^{-4}$

	V_{DD}=1.5V T=5°		V_{DD}=1.5V T=35°	
	SNR	MI	SNR	MI
CMOS	$8 \cdot 10^{-3}$	$3 \cdot 10^{-1}$	$7 \cdot 10^{-3}$	$2 \cdot 10^{-1}$
R-CM$_{dis}$	$4 \cdot 10^{-3}$	$9 \cdot 10^{-2}$	$3 \cdot 10^{-3}$	$7.6 \cdot 10^{-2}$
R-CM	$9 \cdot 10^{-4}$	$5 \cdot 10^{-2}$	$7 \cdot 10^{-4}$	$2.1 \cdot 10^{-2}$
R-DR$_{dis}$	$4 \cdot 10^{-4}$	$3.2 \cdot 10^{-2}$	$2.3 \cdot 10^{-4}$	$1.1 \cdot 10^{-2}$
R-DR	$3.8 \cdot 10^{-5}$	$2.3 \cdot 10^{-3}$	$1.5 \cdot 10^{-5}$	$1.6 \cdot 10^{-3}$

TABLE III: Security metrics for different external power supply voltage and temperature

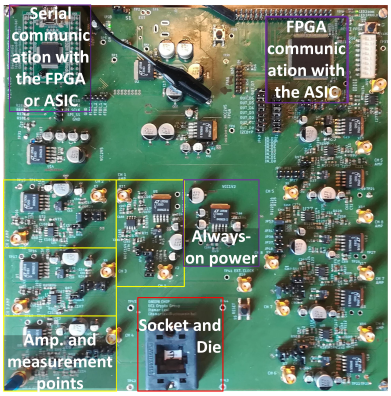


Fig. 18: Test board: test chip is in the socket at the bottom where around the socket are multiple dedicated pre- and post-amplifier measurement points for the different isolated cores.

worst-case scenario, i.e. all power gates are shut down but one. The back-end flow performed was power domain based flow with CPF specifications and finally we perform sign-off stages. Clearly for the entire flow once these 'artificial' elements are inserted we set `dont_touch` to all cells and networks and the optimization is only allowed by placement allocation and buffer insertion if it is a must. These entire steps are fully automated with the standard tools, libraries and flows and can be performed by an experienced designer, making it highly attractive. Furthermore, all verification stages can be performed as part of the normal flow without any custom cells, such as power-grid (IR), signal-integrity steps etc.

APPENDIX B
TEMPERATURE AND VOLTAGE DEPENDENCE

In this Appendix we evaluate security-metrics while varying environmental factors. Such variations may be attributed to malicious parties (e.g. V_{DD} [V] tempering) or due to *honest* environmental conditions (e.g. T in °C). To control the power supply voltage DC, we vary the stable, low-noise power-regulator on-board in the range: $V_{DD} \in \{1, , ..., 1.5\}$ [V] and For temperature control we use the Votsch VT 7004 chamber while sweeping across temperatures of $\{0, ..., 35\}^{\circ}$ C.

In Table. 3 we list the extreme points of our analysis. i.e. low/high voltage with low/high temperature combinations. The main observations are that lowering the supply voltage by 200mV in general reduces the SNR and IT metric which is reasonable due to further signal-reduction with V_{DD}'s lowering and increasing the external supply voltage by 300mV (above which IOs are not guaranteed to function correctly), does the opposite which corresponds to further increasing the signal due to voltage increase. Relating to temperature changes, generally, reducing the temperature for the same voltage level increases the SNR and the IT metric. This is reasonable due to less electronic noise owing to lower temperatures. The specific observations are that the CMOS and R-CM$_{dis}$ designs are more sensitive, and for (e.g.) high voltage and low temperature leak significantly more than the R-DR design. Both the R-CM and R-DR designs does not leak significantly more in those situations. This might be explained due to the noise-generation circuitry which is dominating for those designs. In fact, the R-DR design counter intuitively leak quite similarly even with high-voltage and low temperature to the nominal case.

ACKNOWLEDGMENT

François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research. This work has been funded in parts by the ERC project SWORD and the UCLouvain ARC project NANOSEC.

REFERENCES

[1] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *DATE*. IEEE Computer Society, 2004, pp. 246–251.
[2] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *CRYPTO*, ser. LNCS, vol. 1666. Springer, 1999, pp. 398–412.
[3] C. Herbst, E. Oswald, and S. Mangard, "An AES smart card implementation resistant to power analysis attacks," in *ACNS*, ser. Lecture Notes in Computer Science, vol. 3989, 2006, pp. 239–252.
[4] S. Dziembowski and K. Pietrzak, "Leakage-resilient cryptography," in *FOCS*. IEEE Computer Society, 2008, pp. 293–302.
[5] T. Popp and S. Mangard, "Masked dual-rail pre-charge logic: Dpa-resistance without routing constraints," in *CHES*, ser. Lecture Notes in Computer Science, vol. 3659. Springer, 2005, pp. 172–186.
[6] S. Mangard, T. Popp, and B. M. Gammel, "Side-channel leakage of masked CMOS gates," in *CT-RSA*, ser. Lecture Notes in Computer Science, vol. 3376. Springer, 2005, pp. 351–365.
[7] I. Levi, D. Bellizia, and F.-X. Standaert, "Reducing a masked implementation's effective security order with setup manipulations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 293–317, 2019.
[8] N. Veyrat-Charvillon, M. Medwed, S. Kerckhof, and F. Standaert, "Shuffling against side-channel attacks: A comprehensive study with cautionary note," in *ASIACRYPT*, ser. Lecture Notes in Computer Science, vol. 7658. Springer, 2012, pp. 740–757.
[9] I. Levi, D. Bellizia, and F.-X. Standaert, "Beyond algorithmic noise or how to shuffle parallel implementations?" *International Journal of Circuit Theory and Applications*, 2020.
[10] S. Nikova, V. Rijmen, and M. Schläffer, "Secure hardware implementation of nonlinear functions in the presence of glitches," *J. Cryptology*, vol. 24, no. 2, pp. 292–321, 2011.
[11] Y. Ishai, A. Sahai, and D. A. Wagner, "Private circuits: Securing hardware against probing attacks," in *CRYPTO*, ser. Lecture Notes in Computer Science, vol. 2729. Springer, 2003, pp. 463–481.
[12] G. Barthe, S. Belaïd, F. Dupressoir, P. Fouque, B. Grégoire, and P. Strub, "Verified proofs of higher-order masking," in *EUROCRYPT (1)*, ser. LNCS, vol. 9056. Springer, 2015, pp. 457–485.
[13] G. Barthe, S. Belaïd, F. Dupressoir, P. Fouque, B. Grégoire, P. Strub, and R. Zucchini, "Strong non-interference and type-directed higher-order masking," in *ACM Conference on Computer and Communications Security*. ACM, 2016, pp. 116–129.

[14] A. Duc, S. Faust, and F. Standaert, "Making masking security proofs concrete (or how to evaluate the security of any leaking device), extended version," *J. Cryptology*, vol. 32, no. 4, pp. 1263–1297, 2019.
[15] S. Faust, V. Grosso, S. M. D. Pozo, C. Paglialonga, and F. Standaert, "Composable masking schemes in the presence of physical defaults & the robust probing model," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 89–120, 2018.
[16] G. Barthe, S. Belaïd, G. Cassiers, P. Fouque, B. Grégoire, and F. Standaert, "maskverif: Automated verification of higher-order masking in presence of physical defaults," in *ESORICS (1)*, ser. Lecture Notes in Computer Science, vol. 11735. Springer, 2019, pp. 300–318.
[17] S. Belaïd, D. Goudarzi, and M. Rivain, "Tight private circuits: Achieving probing security with the least refreshing," in *ASIACRYPT (2)*, ser. LNCS, vol. 11273. Springer, 2018, pp. 343–372.
[18] W. Yu and S. Köse, "A lightweight masked aes implementation for securing iot against cpa attacks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 11, pp. 2934–2944, 2017.
[19] S. Mangard, "Hardware countermeasures against DPA ? A statistical analysis of their effectiveness," in *CT-RSA*, ser. Lecture Notes in Computer Science, vol. 2964. Springer, 2004, pp. 222–235.
[20] O. Bronchain, J. M. Hendrickx, C. Massart, A. Olshevsky, and F. Standaert, "Leakage certification revisited: Bounding model errors in side-channel security evaluations," in *CRYPTO (1)*, ser. Lecture Notes in Computer Science, vol. 11692. Springer, 2019, pp. 713–737.
[21] K. Tiri and I. Verbauwhede, "Place and route for secure standard cell design," in *Smart Card Research and Advanced Applications VI*. Springer, 2004, pp. 143–158.
[22] S. Guilley, P. Hoogvorst, Y. Mathieu, and R. Pacalet, "The "backend duplication" method," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2005, pp. 383–397.
[23] B. Sunar, W. J. Martin, and D. R. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Transactions on computers*, vol. 56, no. 1, pp. 109–119, 2007.
[24] K. Wold and S. Petrović, "Behavioral model of trng based on oscillator rings implemented in fpga," in *14th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems*. IEEE, 2011, pp. 163–166.
[25] M. Avital, A. Mordakhay, D. Z. Zabib, Y. Weizman, A. Fish, and O. Keren, "Utilization of process and supply voltage random variations for random bit generation," in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. IEEE, 2018, pp. 317–320.
[26] B. Yang, V. Rožic, M. Grujic, N. Mentens, and I. Verbauwhede, "Es-trng: A high-throughput, low-area true random number generator based on edge sampling," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 267–292, 2018.
[27] M. Kar *et al.*, "8.1 improved power-side-channel-attack resistance of an aes-128 core via a security-aware integrated buck voltage regulator," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2017, pp. 142–143.
[28] M. Kar, A. Singh, S. K. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, "Reducing power side-channel information leakage of aes engines using fully integrated inductive voltage regulator," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 8, pp. 2399–2414, 2018.
[29] J. Heyszl, S. Mangard, B. Heinz, F. Stumpf, and G. Sigl, "Localized electromagnetic analysis of cryptographic implementations," in *CT-RSA*, ser. LNCS, vol. 7178. Springer, 2012, pp. 231–244.
[30] W. Yu and S. Köse, "A voltage regulator-assisted lightweight aes implementation against dpa attacks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 8, pp. 1152–1163, 2016.
[31] W. Yu, O. A. Uzun, and S. Köse, "Leveraging on-chip voltage regulators as a countermeasure against side-channel attacks," in *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, p. 115.
[32] F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2009, pp. 443–461.
[33] E. Prouff and M. Rivain, "Masking against side-channel attacks: A formal security proof," in *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, 2013, pp. 142–159.
[34] I. Levi, A. Fish, and O. Keren, "Low-cost pseudoasynchronous circuit design style with reduced exploitable side information," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 1, pp. 82–95, 2017.
[35] I. Levi, K. Osnat, and F. Alexander, "Pseudo-asynchronous digital circuit design," Jul. 18 2019, uS Patent App. 16/312,317.
[36] H. Groß, S. Mangard, and T. Korak, "Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order." *IACR Cryptology ePrint Archive*, vol. 2016, p. 486, 2016.

Itamar Levi received his M.Sc. degree in Electrical and Computer Engineering from Ben-Gurion University 2013. He completed his Ph.D. at Bar-Ilan University, Israel, in 2017. He was a research-associate in UCLouvain, Belgium until 2019 with the Crypto.-Group. Currently, he is a computer engineering faculty member at Bar-Ilan University (BIU) and a member of the Emerging nanoscaled Integrated Circuits and Systems (EnICs) labs, in Ramat Gan, Israel. His current research interests are digital circuit design, embedded systems security, security evaluation and countermeasures and cryptographic implementations.

Davide Bellizia received the M.D. (summa cum laude) and the Ph.D. in Electronics Engineering from University "La Sapienza" of Rome (Italy), respectively in 2014 and 2018. In 2014 he received the "Laureato Eccellente" award for the best graduate student of the year. In 2017, he joined to the Crypto.-Group of UCLouvain, Louvain-la-Neuve, Belgium, as postdoc researcher. His research interests include SCA countermeasures, design and test of cryptographic ICs, VLSI design and implementation of DSP algorithms.

David Bol is an assistant professor at UCL. He received the Ph.D degree in Engineering Science from UCLouvain in 2008 in the field of ultra-low power digital nanoelectronics. In 2005, he was a visiting Ph.D student at the CNM, Sevilla, Spain, and in 2009, a postdoctoral researcher at intoPIX, Louvain-la-Neuve, Belgium. In 2010, he was a visiting postdoctoral researcher at the UC Berkeley Lab for Manufacturing and Sustainability, Berkeley, CA. In 2015, he participated to the creation of e-peas semiconductors spin-off company, Louvain-la-Neuve, Belgium. He leads the Electronic Circuits and Systems (ECS) research group focused on ultra-low-power design of smart-sensor integrated circuits for the IoT and biomedical applications with a specific focus on environmental sustainability. His personal IC interests include computing, power management, sensing and wireless communications. Prof. Bol has authored more than 100 papers and conference contributions and holds three delivered patents. He (co-)received three Best Paper/Poster/Design Awards in IEEE conferences (ICCD 2008, SOI Conf. 2008, FTFC 2014). He serves as a reviewer for various IEEE journals/conferences and presented several keynotes in international conferences. On the private side, Prof. Bol pioneered the parental leave for male professors in his faculty, to spend time connecting to nature with his family.

François-Xavier Standaert received the Electrical Engineering degree and PhD degree from UCLouvain, respectively in 2001 and 2004. In 2004-2005, he was a Fulbright visiting researcher at Columbia University and at the MIT Medialab. In 2006, he was a founding member of IntoPix s.a. From 2005 to 2008, he was a post-doctoral researcher of the Belgian Fund for Scientific Research (FNRS-F.R.S.) at the UCL Crypto Group and a regular visitor of the two aforementioned laboratories. Since 2008 (resp. 2017), he is associate researcher (resp. senior associate researcher) of the Belgian Fund for Scientific Research (FNRS-F.R.S). In 2011, he was awarded a Starting Independent Research Grant by the European Research Council. In 2016, he has been awarded a Consolidator Grant by the European Research Council. His research interests include cryptographic hardware and embedded systems, low power implementations, the design and cryptanalysis of symmetric cryptographic primitives, physical security issues and side-channel analysis.