Simulatable Leakage, Revisited

Emilie Deprez[1], Charles Momin[1,2], and François-Xavier Standaert[1]

[1] Crypto Group, ICTEAM Institute, UCLouvain, Louvain-la-Neuve, Belgium
{emilie.deprez,charles.momin,fstandae}@uclouvain.be
[2] Belgian National Security Authority (BE-NSA)

Abstract. Finding assumptions that allow reasoning about the leakage-resilience of cryptographic implementations which are at the same time theoretically convenient and practically relevant is notoriously hard. Informally, this is because it requires bounding both the informativeness and the computational power of physical functions for which no simple mathematical expressions are known. The simulatable leakage assumption was introduced as a way to circumvent this problem. Rather than trying to model the leakage function, it posits that it is possible to simulate leakage traces concretely, using the same hardware as a target implementation but without knowledge of the key. Unfortunately, it has also been shown that building concrete simulators is hard, due to the difficulty to capture various types of correlations occurring in physical measurements. In this paper, we revisit the simulatable leakage assumption in two directions. First, we assess the impact of technology (and frequency) scaling on simulatability, and highlight that they are unlikely to provide significantly more positive conclusions. Second, we investigate a hybrid approach combining a concrete simulator for the parts of the traces that are easy to simulate with a theoretical one for the others, leveraging results showing how to obtain security in the (practically relevant) noisy leakage model from security in the (theoretically convenient) bounded leakage model, based on the simulation paradigm also used in the context of masking. We show that such a hybrid approach combined with efficient design tweaks can significantly reduce the amount of bounded leakages needed by theoretical simulators. We also discuss the challenges raised by this approach and their analogy with masking proofs.

Keywords: leakage-resilience, side-channel attacks, simulatable leakage

1 Introduction

Proving the leakage-resilience of cryptographic primitives like Pseudo-Random Generators (PRGs) or Pseudo-Random Functions (PRFs) is known to be challenging [17, 31, 39, 14, 37, 19]. The same holds for authentication and authenticated encryption schemes [25, 29, 1, 13, 5, 4]. One difficulty for this purpose is to restrict the leakage function in a way that is at the same time theoretically convenient and practically relevant. Taking examples from the above references, assuming bounded-length leakages is theoretically convenient but unrealistic,

since side-channel measurements are usually much longer than the key. By contrast, assuming hard-to-invert leakages is practically relevant (and can be used in the context of authentication) but its use for PRGs, PRFs or encryption with leakage so far needs to be combined with idealized assumptions, such as the oracle-free leakages introduced in [39]. As a result, the simulatable leakage assumption was introduced as a promising intermediate [36]. It posits that the leakages of (for example) a block cipher execution can be simulated concretely, using the block cipher's plaintext and ciphertext and the same hardware as a target implementation, but without knowledge of the key. This is a stronger assumption than hard-to-invert leakages [20]. Yet, it avoids the hassle of specifying the computational complexity of the leakage function[3] and it came with a concrete instance of simulator, triggering the hope of establishing a falsifiable assumption on which leakage-resilient cryptography could be rigorously built.

In order to motivate our work, it is worth giving a bit more details about the "split-and-concatenate" simulator proposed in [36]. Say for simplicity we have a 10-cycle hardware implementation of the AES, where one round is performed in one cycle. That means a plaintext x is gradually turned into a ciphertext $z = \mathsf{AES}_k(x)$ by computing intermediate values $y_i = \mathsf{R}^i(x)$, for $1 \leq i \leq 10$, where $\mathsf{R}(.)$ is the round function, $x := y_0$ and $z := y_{10}$. Further assume as a first step that each intermediate value leaks a Hamming weight $\mathsf{HW}(y_i)$, leading to leakage vectors $\boldsymbol{l} = [l_0, l_1, \ldots, l_{10}]$. The simulator works by picking up a random key k^*, computing $x^* = \mathsf{AES}^{-1}(z)$, computing two leakage traces:

$$\boldsymbol{l}_x^{z^*} = [l_0^\dagger, l_1^\dagger, \ldots, l_{10}^\dagger] \leftsquigarrow z^* = \mathsf{AES}_{k^*}(x),$$
$$\boldsymbol{l}_{x^*}^{z} = [l'_0, l'_1, \ldots, l'_{10}] \leftsquigarrow z = \mathsf{AES}_{k^*}(x^*),$$

and producing the simulated trace $\tilde{\boldsymbol{l}}$ by concatenating $\boldsymbol{l}_x^{z^*}(0:5)$ and $\boldsymbol{l}_{x^*}^{z}(6:10)$. Since the leakages of the public plaintext (i.e., $l_0, l_0^\dagger$) and ciphertext (i.e., l_{10}, l'_{10}) are identical, there are two natural strategies for distinguishing the simulated $\tilde{\boldsymbol{l}}$ from a real $\boldsymbol{l} \leftsquigarrow z = \mathsf{AES}_k(x)$: either perform a key recovery or detect an inconsistency between $l_5^\dagger$ and l'_6. For example, the *mathematical correlation* between the Hamming weights of two consecutive rounds could be used for this purpose. The expectation in [36] is that for sufficiently parallel implementations, such correlations are hard to exploit so that the assumption boils down to security against Simple Power Analysis (SPA), i.e., a key recovery attack where the adversary can only observe the leakage of one (or a few) plaintext-ciphertext pairs.[4]

Unfortunately, these positive expectations turned out to be incorrect. As shown by Longo et al., actual leakage traces also exhibit strong *physical correlations* between consecutive samples, which can be exploited to identify simulated

[3] Which is also a hard problem that may impose design tweaks, like the alternating introduced structure in [17, 31], or idealized assumptions, like the oracle-free leakages introduced in [39], in order to avoid so-called "future computation attacks".

[4] A bit more precisely, the number of n-bit values leading to a Hamming weight h is $\binom{n}{h}$ and is heavily concentrated around the mean h. This number rapidly gets very large when n grows, even more if noise is considered. For example, $\binom{128}{64} \approx 2^{124}$.

traces [22]. For example, concrete measurements are generally noisy so that $l_i = \mathsf{HW}(y_i) + r_i$, with r_i some random noise which can be strongly correlated between consecutive rounds. Hence, just estimating the cross-correlation within a noisy trace can be used to falsify the simulatable leakage assumption.

In this paper, we aim to re-evaluate the pros and cons of the simulatable leakage assumption. Our main motivations for this purpose are twofold:

On the one hand, and more than a decade later, no significantly better assumptions have been proposed. In particular, competing alternatives for the analysis of PRGs, PRFs or encryption with leakage remain trivially falsified (e.g., the aforementioned bounded leakage), need to be combined with idealized assumptions (e.g., the aforementioned oracle-free leakages) or imply undesirable design tweaks like the alternating structure mentioned in Footnote 3.

On the other hand, progresses towards simulating (realistic) noisy leakages with bounded leakages are also pushing towards a hybrid approach, where a concrete simulator (based on actual traces generated with a random key) is combined with a theoretical one, using results like [6, 28, 9]. Doing so, the amount of bounded leakage that must be tolerated by a leakage-resilient scheme could be significantly reduced compared to using a theoretical simulator only.[5]

Our main contributions in this respect, which we illustrate with the widespread AES algorithm (but apply similarly to other block ciphers), are twofold:

On the one hand, we revisit the split-and-concatenate simulator against the physical correlation distinguisher put forward by Longo et al. We do that for different, more or less recent, FPGAs (i.e., Xilinx's Spartan-6 and Artix-7) and for different clock frequencies. This allows us to confirm previous results on more recent technologies, and the difficulty to argue that the distinguishing advantage can be made sufficiently small based on an easy-to-control parameter.

On the other hand, we study the aforementioned hybrid approach. More precisely, we consider a setting where $\frac{10}{11}$th of the leakage traces is simulated with a concrete simulator, by running the AES with a random key, and the last cycle (corresponding to the ciphertext) is simulated using the results of [6, 28, 9]. Since the ciphertext is public, this means the amount of bounded leakage needed to simulate this last cycle essentially depends on the residual information on the key that lies in it. In other words, if this last cycle was only depending on the ciphertext y, it would be trivial to simulate.[6] But in case this last cycle also depends on the key, simulation becomes non-trivial. We therefore propose efficient design tweaks to weaken the key dependency of this last cycle in the AES traces, and quantify the residual information using state-of-the-art tools. We conclude by discussing the remaining limitations of the hybrid approach, together with an alternative that could be considered as open problem.

[5] Admittedly, this hybrid approach still does not fix the issues related to the computational complexity of the leakage function mentioned in Footnote 3.

[6] Under perfect knowledge of the leakage distribution, which is always assumed in papers connecting leakage models for leakage-resilience, or for masking [15]. We refer to Section 5 for a discussion of this assumption and its implications.

Overall, we therefore hope the new practical insights that our results provide on the simulatable leakage assumption can trigger further research towards its improvement and its use in the analysis of leakage-resilient designs.

2 Background

Noisy Leakages. We consider a setting where an adversary has access the measurements of a cryptographic implementation running on an electronic device. The measurement of a single execution (denoted as a trace) takes the form of a vector $\boldsymbol{l} = [l_0, l_1, \ldots, l_{N_s-1}]$ of size N_s, where each element is a single sample acquired using an oscilloscope with a fixed sampling rate. An assumption classically encountered in the literature is to model the leakage samples l_t as:

$$l_t = \delta(x) + r_t,$$

where δ is a deterministic function of a value x internally processed at time t by the circuit and r is an independent random (e.g., Gaussian) noise.

Profiled Attacks. Profiled attacks such as Chari et al.'s template attacks [10] are a popular and powerful type of side-channel attack, consisting in two phases: offline training and online attack. During the training phase, the adversary has access to a device similar to the one she aims to attack, and uses it to estimate a (possibly multivariate) statistical model of target's leakages. This model is then used in a second attack phase in order to recover information about the (unknown) sensitive variables that are manipulated by the target device (e.g., S-box outputs in block ciphers). The efficiency of profiled attacks is therefore evaluated based on two metrics: the profiling and the attack complexities, which measure the amount of traces and/or time required during both phases.

Signal-to-Noise Ratio. Leakage traces can be long and building a statistical model using all the time samples is often computationally intensive. As a result, it is a standard practice to first identify so called Points Of Interest (POIs) in the traces and estimate a model for those points only. The Signal-to-Noise Ratio (SNR) is a standard tool for this purpose [23]. For each time sample, it defines the (univariate) signal as the variance of the leakage function's deterministic part and the noise as the variance of the leakage function's probabilistic part. This leads to the following estimate:

$$\mathsf{SNR}(t) = \frac{\hat{\mathsf{Var}}_{x \in \mathcal{X}}(\hat{\mathsf{E}}(L_t^x))}{\hat{\mathsf{E}}_{x \in \mathcal{X}}(\hat{\mathsf{Var}}(L_t^x))},$$

where $\hat{\mathsf{Var}}$ and $\hat{\mathsf{E}}$ denote the sample variance and expected value, $\mathcal{X}$ is the domain of the variable x and L_t^x is a vector of samples at time t for a given x.

Multivariate Gaussian Model in a Linear Subspace. The original template attacks model the Probability Density Function (PDF) of the true leakage function with a multivariate Gaussian distribution. In this context, the training phase boils down to estimating the joint distribution of $N_p \leq N_s$ POIs conditioned to the value x of a target variable, denoted as $\hat{\mathsf{f}}(\boldsymbol{l}|x)$. Since this estimation can be expensive when N_p is large, dimensionality reduction techniques have been introduced to make it more efficient. For example, Linear Discriminant Analysis (LDA) can be used to identify a projection matrix $\mathbf{W}$ maximising the SNR of the $N_d < N_p$ dimensions kept after projection in a linear subspace [35]. When using the latter, the templates are expressed as:

$$\hat{\mathsf{f}}(\boldsymbol{l}|x) = \frac{1}{\sqrt{(2\pi)^{N_d}\,|\hat{\boldsymbol{\Sigma}}|}} \mathsf{exp}\left(-\frac{1}{2}(\mathbf{W}\boldsymbol{l} - \hat{\mu}_x)^\top \hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{W}\boldsymbol{l} - \hat{\mu}_x)\right),$$

where $\hat{\mu}_x$ is the estimated mean vector, $\hat{\boldsymbol{\Sigma}}$ the estimated covariance matrix, $\boldsymbol{l}$ is a trace composed of N_p POIs and $\mathbf{W}$ is the linear projection matrix.

IT Metrics and Model Evaluation. The Mutual Information (MI) theoretically quantifies the amount of information related to a target variable that can be extracted from a leakage function. It is a popular metric bound the complexity of side-channel attacks [34]. However, its direct computation is not possible since it requires the knowledge of the true (and unknown) leakage PDF. The Perceived Information (PI) was proposed as a surrogate that lower bounds the MI and quantifies the amount of information that can be exploited by a given statistical model produced during a training phase [8]. In this setting, considering a training traces' set $\mathcal{L}_t$ used to build the model and an (independent) evaluation traces' set $\mathcal{L}_e$, the PI can be estimated as:

$$\hat{\mathsf{PI}}(X;\mathsf{L}) = \mathsf{H}(X) + \sum_{x\in\mathcal{X}} \mathsf{Pr}(x) \sum_{\boldsymbol{l}\in\mathcal{L}_e^x} \frac{1}{|\mathcal{L}_e^x|} \mathsf{log}_2 \hat{\mathsf{Pr}}(x|\boldsymbol{l}),$$

where $\mathcal{L}_e^x$ is the subset of trace from $\mathcal{L}_e$ for which the target variable X equals x and $\hat{\mathsf{Pr}}(x|\boldsymbol{l})$ is obtained by applying Bayes' rule with trained templates.

The PI increases with the training complexity (i.e., $|\mathcal{L}_t|$). Complementarily, the Training Information (TI) decreases with the training complexity and has been shown to provide an upper bound to the PI [26]. It is estimated as the PI, but evaluating the information in an overfitting manner by using the training set $\mathcal{L}_t$ instead of $\mathcal{L}_e$. Evaluating the convergence of both the PI and the TI as a function of the training complexity is therefore a convenient manner to identify if increasing the training complexity can lead to model improvements.

Correlation-Based Leakage Detection. The ρ-test is a leakage detection test based on the Correlation Power Analysis (CPA) distinguisher [7] that was introduced as an alternative to the classical fixed vs. random t-test, allowing

a better exploitability of the test result at the cost of a larger sampling complexity [16]. To be data efficient, the ρ-test generally takes advantage of k-fold cross-validation and considers k non-overlapping sets of similar size $\mathcal{L}^{(i)}$ (obtained by splitting a traces set $\mathcal{L}$), from which the profiling sets $\mathcal{L}_p^{(j)} = \cup_{i \neq j} \mathcal{L}^{(i)}$ and the evaluation sets $\mathcal{L}_e^{(j)} = \mathcal{L} \backslash \mathcal{L}_p^{(j)}$ are defined. For a target intermediate variable X, the training sets are first used to estimate leakage models $\hat{\mathsf{model}}_t^{(j)}(X)$ for every time sample, and then used to estimate the correlation with the test set $\mathcal{L}_e^{(j)}$ as follows:

$$\hat{r}_t^{(j)} = \hat{\rho}\left(L_{X,t}^{(j)}, \hat{\mathsf{model}}_t^{(j)}(X)\right),$$

where $L_{X,t}^{(j)}$ depicts the traces associated to the variable X. The k cross-validation results are then averaged in a single unbiased result $\hat{r}_t$, from which a sample following a close-to-normal distribution $\mathcal{N}(0,1)$ can be obtained by applying Fisher's Z transform as follow:

$$\hat{r}_t^z = \frac{\sqrt{N-3}}{2} \times \mathsf{ln}\left(\frac{1+\hat{r}_t}{1-\hat{r}_t}\right),$$

where $N = |\mathcal{L}|$. It follows that we can reject the null hypothesis (that assumes no correlation) when the sample is above a threshold such as:

$$\mathsf{Q}_{\mathcal{N}(0,1)}\left(1-\alpha/2\right) \leq |\hat{r}_t^z|,$$

where $\mathsf{Q}_{\mathcal{N}(0,1)}$ is the quantile function of the Gaussian distribution and α the test significance level. In this work, we use the test heuristically to reject obvious equal correlation hypotheses, and consider a threshold of 5 for this purpose.

3 Split-and-Concatenate Simulator

We first revisit the split-and-concatenate simulator together with the correlation attack that falsified it. We do that for different FPGA technologies (and clock frequencies) and use our experimental results to argue about the difficulty of building concrete simulators based on leakage trace manipulations (split, concatenate or others), motivating the investigations in the next section.

3.1 The Crypto 2013 Proposal

The concept of simulatable leakage was introduced at Crypto 2013 [36]. It relies on the q-sim game, where an adversary has to distinguish real leakage traces obtained with q different block cipher plaintexts from simulated traces that must be produced without knowledge of the secret key but with the same hardware as the real traces. Leakages are said to be (q, t, ϵ)-simulatable if the probability of determining whether the traces are real or simulated for adversaries with time complexity t is smaller than ϵ. This definition came with a first proposal of concrete "split-and-concatenate" simulator, outlined in introduction.

The main challenges that this simulator faces is to stay consistent with the plaintext and the ciphertext, which are public knowledge, without knowing the key. For this purpose, and say one has to simulate a leakage trace $\boldsymbol{l} \leftsquigarrow z = \mathsf{AES}_k(x)$, the proposal was to use a random key k^* and to concatenate half a trace starting with the correct plaintext x and half a trace ending with the correct ciphertext z. That is $\boldsymbol{l}_x^{z^*} \leftsquigarrow z^* = \mathsf{AES}_{k^*}(x)$, $x^* = \mathsf{AES}_{k^*}^{-1}(y)$, $\boldsymbol{l}_{x^*}^{z} \leftsquigarrow z = \mathsf{AES}_{k^*}(x^*)$ and $\tilde{\boldsymbol{l}} = \boldsymbol{l}_x^{z^*}(0:5)||\boldsymbol{l}_{x^*}^{z}(6:10)$, where $\boldsymbol{l}_x^{z^*}(0:5)$ and $\boldsymbol{l}_{x^*}^{z}(6:10)$ respectively denote the first and the second halves of the traces $\boldsymbol{l}_x^{z^*}$ and $\boldsymbol{l}_{x^*}^{z}$.

Unfortunately, this split and concatenate simulator was later falsified by Longo et al. [22], who exploited the absence of cross-correlation across the point in time of the leakage traces where the split takes place to distinguish them.

3.2 Measurement Setup and Target Implementations

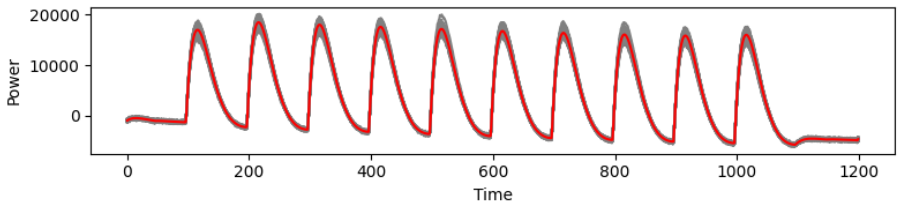
For our experiments, we considered a round-based hardware implementation of the AES with a 128-bit architecture, which relies on 20 instances of S-boxes (i.e., 16 for the rounds, 4 for the key scheduling), 4 MixColumn units (each of them operating on a single column) and ShiftRow implemented at no cost through routing. A dedicated mux is used to bypass MixColumn at the last round. The key scheduling operations are performed in parallel to the round computations with dedicated logic, which results in the computation of a round in a single clock cycle. The full encryption process is achieved by sequentially computing round operations, by routing the results of the latter to the two 128-bit registers storing the input of the computing logic (apart from the first cycle, during which a MUX is used to fetch both the 128-bit input plaintext and key instead).

Concretely, we rely on two different target FPGAs for our analyzes: a Xilinx Spartan-6 XC6SLX75 mounted on a Sakura-G board and a Xilinx Artix-7 XC7A100T mounted on a CW305 board. We acquire power traces using a Tektronix CT1 current probe connected to a PICO6242E oscilloscope using a running frequency of 1.526MHz provided by a Keysight E36102B as external power source (resp., connected on EXTVIN for the Sakura-G and banana jack for the CW305). For both cases, the clock signal used by the DUT is generated using the AWG feature of the oscillocope and provided as an external clock through dedicated SMA IOs (i.e., J6 on the Sakura-G and CLKIN on the CW305).

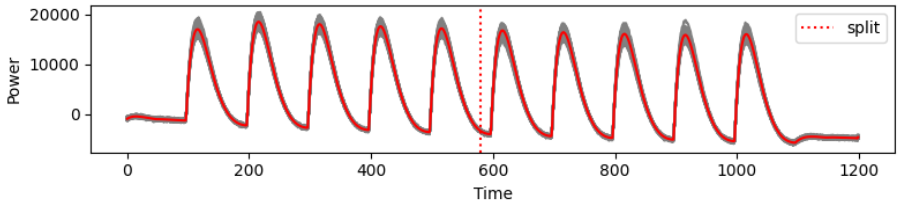
3.3 Correlation Distinguisher Results

Before discussing the results of the correlation distinguisher, we show the shapes of the (real and simulated) leakage traces for both the Spartan-6 and the Artix-7 targets. They are reported in Figures 1 and 2, respectively. Traces look essentially similar for both boards, though their amplitude slightly differs, suggesting a higher power consumption for the (older) Spartan-6 technology. There is also no obvious pattern allowing to distinguish real and simulated traces.

As a natural next step, Figures 3 and 4 show the results of Longo et al.'s cross-correlation distinguisher applied to the Spartan-6 and Artix-7 target. They

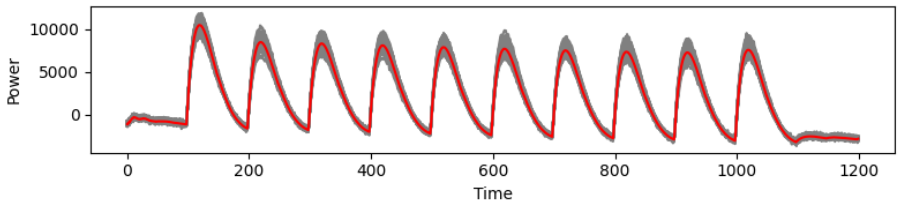


(a) Real leakage traces.

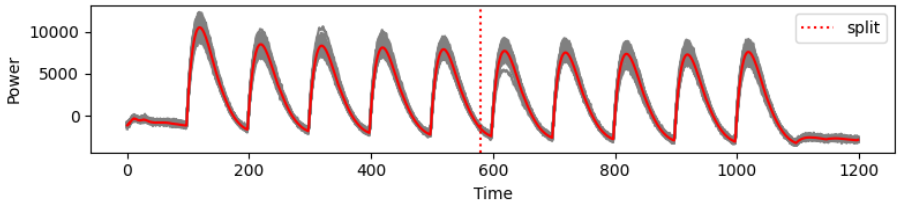


(b) Simulated leakage traces.

Fig. 1: Real and simulated leakage traces acquired from the Spartan-6 target with different shades of gray corresponding to different traces and average in red. DUT running at 1562500.0 Hz and sampling frequency 156250000.0 Hz.



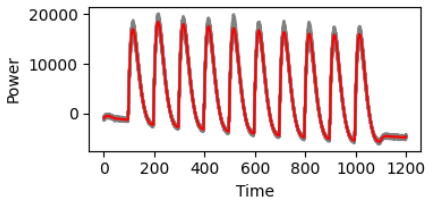
(a) Real leakage traces.



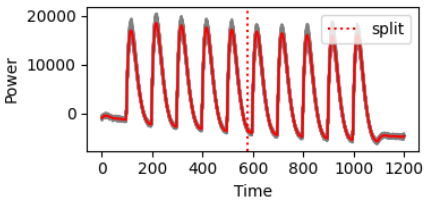
(b) Simulated leakage traces.

Fig. 2: Real and simulated leakage traces acquired from the Artix-7 target with different shades of gray corresponding to different traces and average in red. DUT running at 1562500.0 Hz and sampling frequency 156250000.0 Hz.

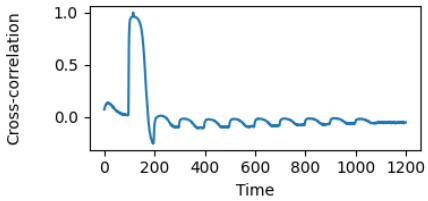
highlight that the cross-correlations of the real and simulated traces confidently differ after the split (the threshold on the graphs indicate whether the hypothesis of equal cross-correlation is rejected). Furthermore, we can observe that technology scaling is not leading to easier simulation as detection succeeds easily with both the Spartan-6 and the Artix-7 FPGAs. For completeness, we also ran similar experiments for the same devices running at lower frequencies. The results depicted in Figures 5 and 6 show that this can only help marginally.



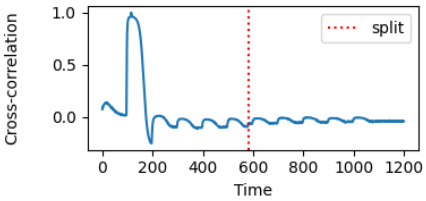
(a) Real leakage traces.



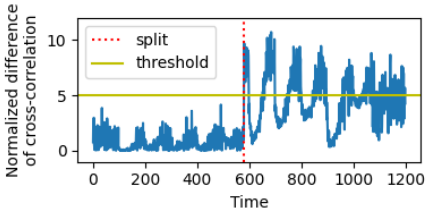
(b) Simulated leakage traces.



(c) Cross-correlation of real traces.



(d) Cross-correlation of simulated traces.

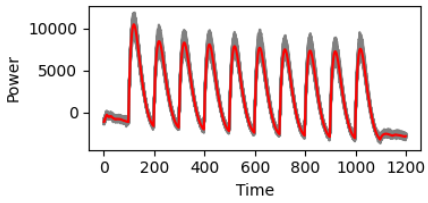


(e) Difference of correlations.

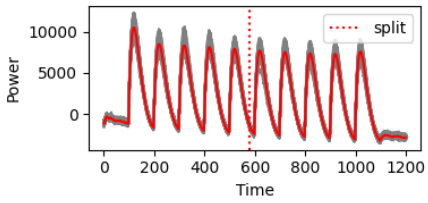
Fig. 3: Comparison of the cross-correlations between real and simulated leakage traces acquired from the Spartan-6 target, and result of a ρ-test. DUT running at 1562500.0 Hz and sampling frequency 156250000.0 Hz.

3.4 Discussion

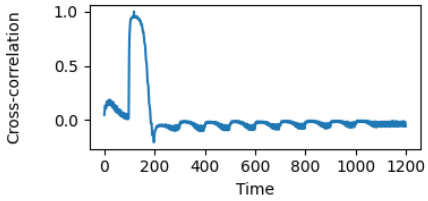
The experiments in this section confirm the results of Longo et al. They also highlight that technology scaling is unlikely to help significantly. This matches the fact that as transistor sizes shrink, the coupling between the physical effects of close operations (whether in space or time) are expected to be strength-



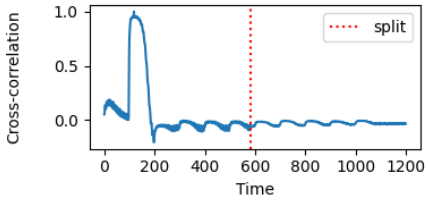
(a) Real traces.



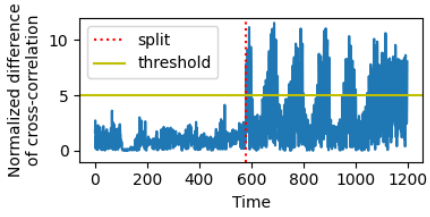
(b) Simulated traces.



(c) Cross-correlation real traces.

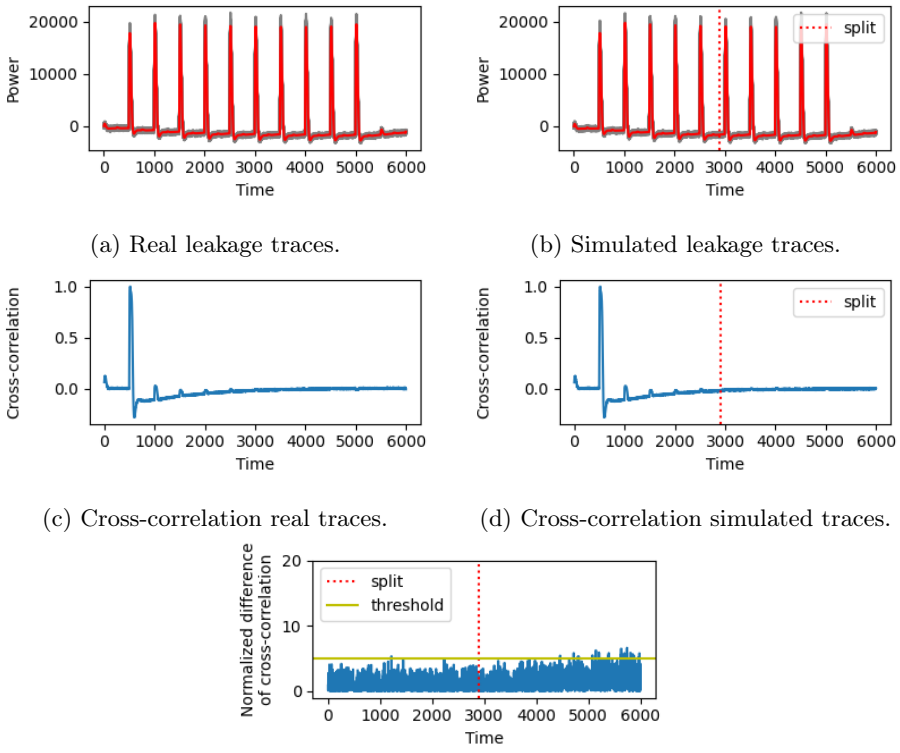


(d) Cross-correlation simulated traces.



(e) Difference of correlations.

Fig. 4: Comparison of the cross-correlations between real and simulated leakage traces acquired from the Artix-7 target, and result of a ρ-test. DUT running at 1562500.0 Hz and sampling frequency 156250000.0 Hz.



(a) Real leakage traces.

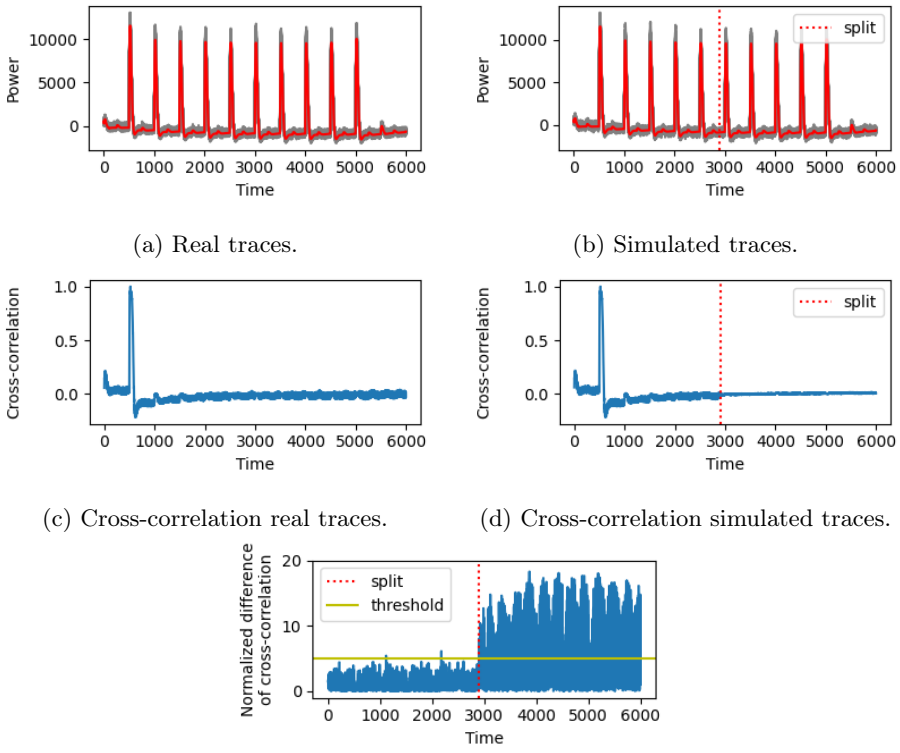
(b) Simulated leakage traces.

(c) Cross-correlation real traces.

(d) Cross-correlation simulated traces.

(e) Difference of correlations.

Fig. 5: Comparison of the cross-correlations between real and simulated leakage traces acquired from the Sakura-G board, and result of a ρ-test. DUT running at 312500.0 Hz and sampling frequency 156250000.0 Hz.



(a) Real traces.

(b) Simulated traces.

(c) Cross-correlation real traces.

(d) Cross-correlation simulated traces.

(e) Difference of correlations.

Fig. 6: Comparison of the cross-correlations between real and simulated leakage traces acquired from the CW305 board, and result of a ρ-test. DUT running at 312500.0 Hz and sampling frequency 156250000.0 Hz.

ened [12, 33, 38]. As a result, (concrete) simulators exclusively based on manipulating traces such as the split-and-concatenate simulator, or extensions thereof as suggested in [30], are unlikely to provide confident security guarantees. This is because ultimately, a perfect simulation would require an extremely accurate modeling of the leakage function, which is precisely what is lacking for complex circuits. In the next section, we therefore step back to an alternative hybrid approach, trying to leverage concrete simulators for what they are good for, and to rely on theoretical simulation approaches to remedy their weaknesses.

4 Hybrid Simulation Approach

The previous section confirms that the main issue with concrete simulators is the need to combine leakage traces produced separately in a hard-to-distinguish way. For example, the split-and-concatenate simulator leads to distinguishable patterns at the point in time where the concatenation takes place.

An alternative approach is to move from such concrete simulators to theoretical ones, as vastly used in the masking literature. For example, Duc et al. first showed how to simulate (practically relevant) noisy leakages from (theoretically convenient) probing leakages [15]. Many follow-up works extended and improved this simulation paradigm – see [32, 2] for recent ones. Interestingly, a similar connection has been made for the bounded leakage model, so that it is theoretically possible to simulate noisy leakages from bounded leakages [6, 28, 9]. Quite naturally, the more samples must be simulated and the less noisy they are, the more bounded leakages is needed, which then has to be tolerated by the leakage-resilient construction under investigation. So to make stronger security claims, there is a general incentive to minimize the amount of samples to simulate and to make them noisy. The natural (and for now main) approach for this purpose is to leverage parallel computations in hardware implementations.

Practicaly, this means that even in parallel implementations, the amount of bounded leakages needed to simulate a leakage trace grows linearly with the amount of operations (e.g., rounds in our case study with an FPGA implementation of the AES). In this section, we therefore propose an intermediate approach, trying to combine the pros and cons of concrete and theoretical simulators. The idea is to simulate most of the trace (i.e., all the cycles but the final one manipulating the ciphertext) with a concrete leakage simulator. This easy since up to the ciphertext, the traces will be perfectly consistent. We then simulate the last cycle with a theoretical simulator. Assuming that the leakage at this stage mostly depends on the public ciphertext, this is expected to be easier than simulating an inner cycle of the implementation. Yet, it naturally raises the question of making this final cycle as independent as possible from the secret key. In the rest of this section, we therefore start by describing a minimally modified implementation of the AES that aims to satisfy this goal (in Section 4.1), before analyzing the residual key leakage making the simulation non-trivial (in Section 4.2).

We note that theoretical simulators admittedly raise concerns with respect to their connection to practice. We discuss remaining gaps and potential remedies in Section 5. Meanwhile, the next results should be understood as a way to substantially improve the guarantees offered by theoretical simulators.

4.1 Output Isolating Implementation

As a first attempt towards minimizing the key dependency of the last cycle of an AES implementation, we consider a slightly tweaked version of the previous round-based architecture. Our goal is to enforce that the power consumption of this last cycle depends mostly (if not exclusively) on the ciphertext z. For this purpose, we will focus on correlations in the dynamic power consumption, assuming that the implementation is running at a sufficiently high frequency so that it dominates over static leakage [27], the investigation of which is left as an interesting topic for further research. To ease our architecture description, we denote the cycle indexes of a single execution as c_i, where $0 \leq i \leq 10$.

As depicted in Figure 7, the specificity of the core lies in the implementation of the two last rounds. During the penultimate cycle (i.e., c_9), the outputs of the S-boxes from the previous round (coming from the round datapath) as well as the last 128-bit round key, are XORed with the same 128-bit random word r (fetched as an additional input of the core and loaded in R4 by enabling it during c_8).[7] This results in two 128-bit shares z_0 and z_1 of the output ciphertext. At the cycle end, these are fetched by 2 dedicated 128-bit registers (enabled only during c_9), while the remaining registers of the core are stalled. The stalling mechanism is used to limit the dynamic power consumption to the impact of the two shares manipulated in parallel, which are the two only words transitioning during c_{10}, in the registers R2 and R3. The ciphertext is finally computed during an extra clock cycle where the 2 shares are recombined with dedicated XORs. The core then remains stalled until it is reset for a new execution to start over. This tweaked version consumes 150% more area than the standard implementation.

Such an architecture is expected to contribute to our goal since all the logic (apart from R2 and R3) is stalled during c_{10} and no combinational path other than the XOR between z_0 and z_1 located at the output exists in the circuitry. The extent to which it reduces (hard-to-model) capacitive and inductive effects that may still lead key dependencies to show up in the last cycles nevertheless needs to be confirmed experimentally, which is the goal of the next section.

4.2 Residual Leakage Analysis

As a first step towards investigating the residual key dependencies in the last cycle of our prototype implementation, we evaluate the SNR. Figures 8a and 10a show the leakage traces of our two targets, where we see that the final cycle (where only the XOR between z_0 and z_1 is executed) consumes significantly

[7] The ShiftRow routing is not depicted in the figure for clarity.

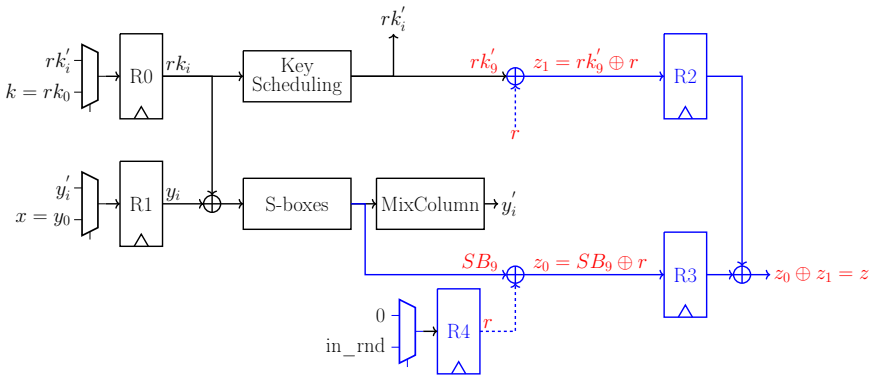


Fig. 7: Tweaked round-based AES architecture for hybrid simulation. The output isolating dedicated logic is depicted in blue. The busses are all 128-bit wide. The computation performed during the penultimate and last cycles are depicted in red. Registers R0 and R1 are enabled during $c_{0 \leq i < 9}$. Registers R2 and R3 are enabled during c_9 and R4 is enable during c_8.

less power, presumably due to the small proportion of the FPGAs' logic that is dedicated to the computations of the final cycle.[8] Complementarily, Figures 8b and 10b depict the SNR of the 16 S-boxes outputs in the last AES round (i.e., cycle c_9).[9] We see that the residual SNR on cycle c_{10} is significantly lower than the actual one on cycle c_9. To given a numerical flavor, the average SNR (over the 16 S-boxes) on cycle c_9 is worth 6.4×10^{-3} (resp., 1.8×10^{-3}) with the Spartan-6 (resp., Artix-7) and drops to 0.3×10^{-3} (resp., 0.4×10^{-3}) on cycle c_{10}.

Theoretically though, the relevant SNR is the one corresponding to the full AES state. Indeed, in order to obtain provable security guarantees, it is necessary to simulate the full traces (not the sub-parts depending on eight bits).[10] We therefore also estimated a state SNR, computed for a fixed amount of random plaintext-key pairs (x_i, k_i) that satisfy $z = \mathsf{AES}_k(x)$ with a constant value z, so that only the residual key dependency (and not the ciphertext dependency) should show up in cycle c_{10}. It is represented in Figure 8c and 10c, where we see both a significant increase of the SNR for all cycles (since its corresponds to a 128-bit signal rather than an 8-bit signal) and a significant drop for the final cycle, which corresponds to the residual dependency we want to estimate.

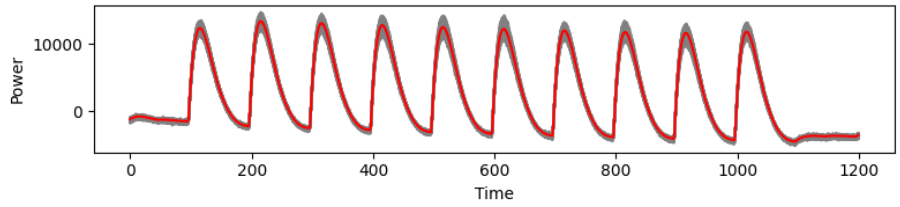
We finally extended our analysis of the residual key dependency in cycle c_{10} from univariate (with the SNR) towards multivariate, by quantifying the PI and TI obtained with a LDA distinguisher. Precisely, we estimated the PI and TI

[8] The output XOR layer requires ≈ 22 times less logic than the full round based on synthesis results obtained with Yosys and the 45nm Nangate FreePDK.

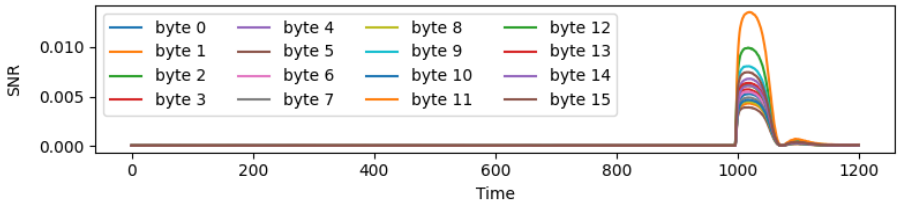
[9] The S-boxes have different SNRs, presumably due to their placement and routing.

[10] Unless making the heuristic assumption that a side-channel adversary is computationally limited to extract information about 8- or 32-bit intermediate variables.

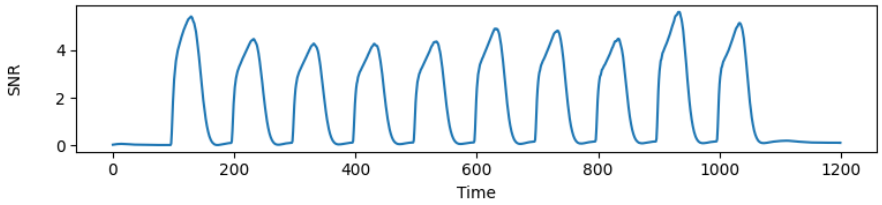
between the leakage and the pairs (x_i, k_i) that can be extracted from the last cycle. We also report the information extracted for a middle round (the third one) for comparison purposes. As depicted on Figures 9 and 11, the third cycle is ≈ 4 (resp., 3.5) times more informative than the last one when the Spartan-6 (resp., Artix-7) target is considered, for models that have converged (i.e., PI and TI values are close), confirming that the residual key dependencies in the final round are significantly lower than the direct ones in the middle rounds.



(a) Power traces, with the average trace depicted in red.



(b) SNR for 10-th round S-boxes output (10e6 traces with random plaintext and key).

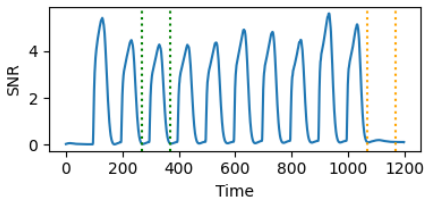


(c) SNR for uniform pairs $p_{0 \leq i < 256}$ and fixed output $z = \mathsf{AES}_{k_i}(x_i)$ (1e6 traces).

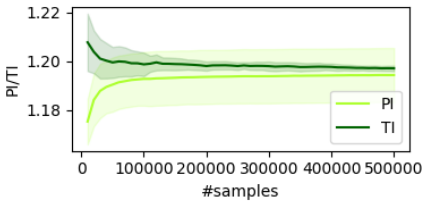
Fig. 8: S-box, state and residual SNR results for the Spartan-6 target. DUT running at 1.5625MHz and sampling frequency of 156.25MHz.

4.3 Discussion

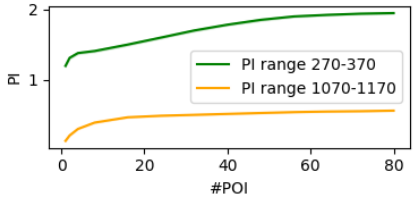
Compared to using theoretical simulators like [6, 28, 9] only, the hybrid approach in this section leads to significant quantitative improvements. First, $\frac{10}{11}$th of the leakage trace can be simulated concretely (so does not require bounded leakage).



(a) 3rd cycle (green), last cycle (orange).



(b) PI/TI convergence.



(c) PI/TI vs. number of POIs.

Fig. 9: Residual (multivariate) key information in cycle c_{10} for the Spartan-6 target. DUT running at 1.5625MHz and sampling frequency of 156.25MHz.

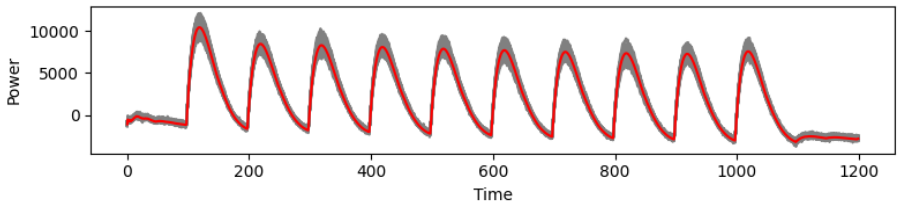
Second the residual information lying in the final cycle of our implementation that must be theoretically simulated from bounded leakage is approximately 4 times lower than the information of a state in the middle AES rounds.

Note that this hybrid approach is also applicable for a standard round-based implementation (i.e., without the output isolating tweak of Section 4.1). In this case, the reduction of the amount of bounded leakage needed to simulate by a factor 11 (corresponding to the AES rounds) remains, and the additional factor 4 corresponding to the reduction of the residual information would be lost.

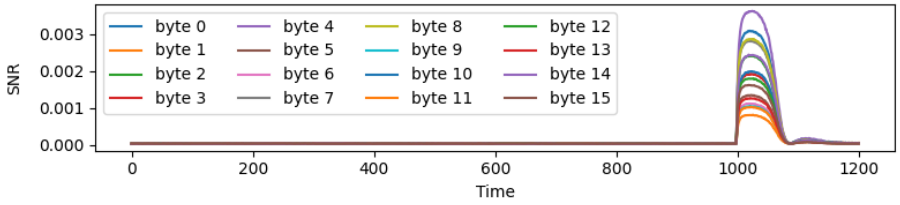
5 Conclusions and Open Problems

Considering the results in the two previous sections, we finally discuss the remaining gaps between the theoretical and the concrete approaches, as it may sound like things that are hard to achieve concretely are possible in theory. We consider the main assumptions of the theoretical approach for this purpose.

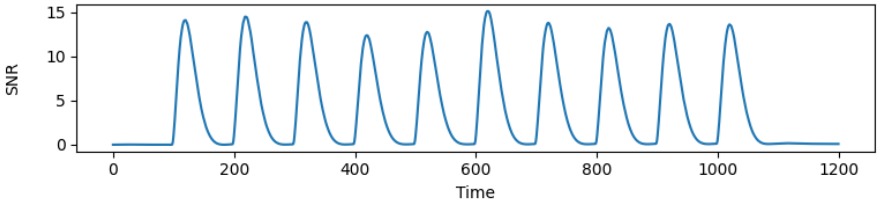
1. Perfect Knowledge of the Leakage Distribution. The theoretical simulators we leverage [6, 28, 9], just as the theoretical simulators of noisy leakages from random probing leakages in the context of masking [15, 32, 2], assume a perfect knowledge of the leakage distribution. This is surprising at first, given that the difficulty to characterize the capacitive and inductive effects in our exemplary implementations is precisely what prevents the split-and-concatenate simulator (and extensions thereof) to succeed. The reason why this modeling difficulty does



(a) Power traces, with the average trace depicted in red.

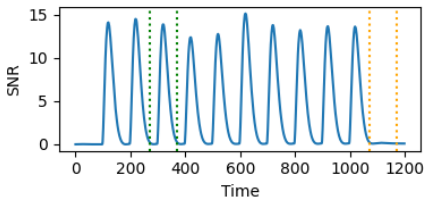


(b) SNR for 10-th round S-boxes output (10e6 traces with random plaintext and key).

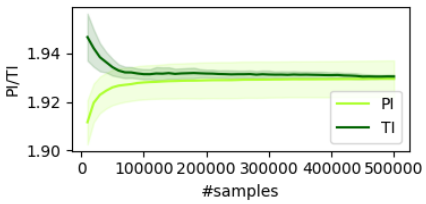


(c) SNR for uniform pairs $p_{0 \leq i < 256}$ and fixed output $z = \mathsf{AES}_{k_i}(x_i)$ (1e6 traces).

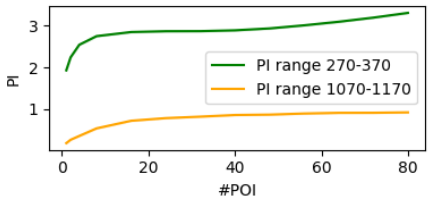
Fig. 10: S-box, state and residual SNR results for the Artix-7 target. DUT running at 1.5625MHz and sampling frequency of 156.25MHz.



(a) 3rd cycle (green), last cycle (orange).



(b) PI/TI convergence.



(c) PI/TI vs. number of POIs.

Fig. 11: Residual (multivariate) key information in cycle c_{10} for the Artix-7 target. DUT running at 1.5625MHz and sampling frequency of 156.25MHz.

not affect the theoretical and concrete simulation approaches to the same extent relates to their fundamentally different nature: concrete simulators as proposed in [36] are constructive whereas theoretical ones are existential. These theoretical simulators guarantee that security in the bounded leakage model ensures security in the noisy leakage model. For this, they assume an exact description of the leakage function is available (despite it may be hard to estimate concretely) and show that any noisy leakage function with limited informativeness (which is easier to estimate) can be simulated. In other words, the hybrid approach ensures that a simulator for the last cycle of our implementation exists given some amount of bounded leakage, despite we cannot exhibit it explicitly.[11]

2. Independence Assumption. Besides, the aforementioned theoretical approaches simulate the leakages of all the operations in a cryptographic implementation independently, and rely on composition results to capture their combination. Yet, it is well known that neither the side-channel signal nor the side-channel noise of consecutive samples in a leakage trace are independent.

In the context of masking, signal dependencies due to glitches [24] or transitions [11] can be strongly detrimental and are usually integrated in the models [18], motivating design tweaks causing implementation overheads. Besides, integrating noise dependencies (over different shares) remains a challenge [3].

[11] This also explains why the two approaches are evaluated quite differently, using a concrete distinguisher in Section 3 and information theoretic metrics in Section 4.

Interestingly, dependencies within leakage traces are better captured by simulators using bounded leakage for leakage-resilience as we consider in this work. First, signal dependencies actually reduce the amount of bounded leakage needed to simulate [9]. This strongly differs from the context of the probing model for masking, but is nevertheless intuitive. In the context of masking, signal dependencies can lead shares to recombine (breaking a security order condition). In the context of leakage-resilience, there is no security order at stakes and signal correlations just enable the same bounded leakage to be useful for simulating multiple clock cycles.[12] In practice, this for example limits the adversaries' incentive to increase their sampling frequencies. Second, correlated noise can be detrimental (as when masking), but its impact can be integrated in the simulation theorems [9], something that can be done per share but remains an open problem over the shares in reductions from probing leakages [3]. Combined with the fact that the granularity of the analyzes differ between masking (finer-grain) and leakage-resilience (coarser-grain), we can conclude that physical dependencies in the side-channel signal and noise are significantly less critical in the context leakage-resilience we consider in this work than in the context of masking.

Eventually, it is worth mentioning that small deviations from an independence assumption may not always affect security. For example in the context of masking, it is in general hard to rule out that actual leakage traces may exhibit (small) recombinations of the shares. Yet, as long as the best attacks do not benefit from such imperfections, this is usually tolerated (e.g., if a 2-share implementation leaks at order one with negligible amplitude). This matches the situation in symmetric cryptography and the concept of non-hermetic design strategy, where distinguishers against building blocks (e.g., permutations) are tolerated as long as they do not lead to improved attacks against the modes.[13] It suggests the investigation of such non-hermetic design strategies in the context of leakage-resilience as an interesting research direction. That is, one could try to show that despite simulated traces can be distinguished, such distinguishers do not significantly affect the security. First steps in this direction have been made in [21] based on a Hamming weight model for the measurements, and could be extended to actual implementations. It would allow getting rid of bounded leakage in the simulations (and the design tweaks and overheads it causes).

Acknowledgments. François-Xavier Standaert is a research director of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in part by the Walloon Region through the project CyberExcellence (convention number 2110186) and by the ERC Advanced Grant project 101096871 (acronym BRIDGE).

Views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

[12] In other words, correlated signal may help theoretical simulators despite making concrete simulators harder to build because making the estimation of the leakage function more complex. As mentioned in the above Point 1, this is not contradictory because the nature of these simulators is different (concrete vs. existential).

[13] See https://keccak.team/files/CSF-0.1.pdf for a discussion.

References

1. Barwell, G., Martin, D.P., Oswald, E., Stam, M.: Authenticated encryption in the face of protocol and side channel leakage. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10624, pp. 693–723. Springer (2017). https://doi.org/10.1007/978-3-319-70694-8_24
2. Béguinot, J., Cheng, W., Guilley, S., Rioul, O.: Formal security proofs via doeblin coefficients: - optimal side-channel factorization from noisy leakage to random probing. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 14925, pp. 389–426. Springer (2024). https://doi.org/10.1007/978-3-031-68391-6_12
3. Belaïd, S., Cassiers, G., Mutschler, C., Rivain, M., Roche, T., Standaert, F., Taleb, A.R.: Sok: A methodology to achieve provable side-channel security in real-world implementations. IACR Commun. Cryptol. **2**(1), 4 (2025). https://doi.org/10.62056/AEBNGY4E-
4. Bellizia, D., Bronchain, O., Cassiers, G., Grosso, V., Guo, C., Momin, C., Pereira, O., Peters, T., Standaert, F.: Mode-level vs. implementation-level physical security in symmetric cryptography - A practical guide through the leakage-resistance jungle. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12170, pp. 369–400. Springer (2020). https://doi.org/10.1007/978-3-030-56784-2_13
5. Berti, F., Guo, C., Peters, T., Standaert, F.: Efficient leakage-resilient macs without idealized assumptions. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 13091, pp. 95–123. Springer (2021)
6. Brian, G., Faonio, A., Obremski, M., Ribeiro, J., Simkin, M., Skórski, M., Venturi, D.: The mother of all leakages: How to simulate noisy leakages via bounded leakage (almost) for free. In: Canteaut, A., Standaert, F. (eds.) Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12697, pp. 408–437. Springer (2021). https://doi.org/10.1007/978-3-030-77886-6_14
7. 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: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings. Lecture Notes in Computer Science, vol. 3156, pp. 16–29. Springer (2004). https://doi.org/10.1007/978-3-540-28632-5_2
8. Bronchain, O., Hendrickx, J.M., Massart, C., Olshevsky, A., Standaert, F.: Leakage certification revisited: Bounding model errors in side-channel security evaluations. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA,

August 18-22, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11692, pp. 713–737. Springer (2019). https://doi.org/10.1007/978-3-030-26948-7_25

9. Béguinot, J., Mukherjee, A., Obresmki, M., Ribeiro, J., Roy, L., Standaert, F.X., Venturi, D.: Simulating noisy leakage with bounded leakage: Simpler, better, faster. Cryptology ePrint Archive, Paper 2026/357 (2026), https://eprint.iacr.org/2026/357
10. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Jr., B.S.K., Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2523, pp. 13–28. Springer (2002). https://doi.org/10.1007/3-540-36400-5_3
11. Coron, J., Giraud, C., Prouff, E., Renner, S., Rivain, M., Vadnala, P.K.: Conversion of security proofs from one leakage model to another: A new issue. In: Schindler, W., Huss, S.A. (eds.) Constructive Side-Channel Analysis and Secure Design - Third International Workshop, COSADE 2012, Darmstadt, Germany, May 3-4, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7275, pp. 69–81. Springer (2012). https://doi.org/10.1007/978-3-642-29912-4_6
12. Dally, W.J., Poulton, J.W.: Digital Systems Engineering. Cambridge University Press (2001)
13. Dobraunig, C., Mennink, B.: Leakage resilience of the duplex construction. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III. Lecture Notes in Computer Science, vol. 11923, pp. 225–255. Springer (2019). https://doi.org/10.1007/978-3-030-34618-8_8
14. Dodis, Y., Pietrzak, K.: Leakage-resilient pseudorandom functions and side-channel attacks on feistel networks. In: Rabin, T. (ed.) Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6223, pp. 21–40. Springer (2010). https://doi.org/10.1007/978-3-642-14623-7_2
15. Duc, A., Dziembowski, S., Faust, S.: Unifying leakage models: From probing attacks to noisy leakage. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441, pp. 423–440. Springer (2014). https://doi.org/10.1007/978-3-642-55220-5_24
16. Durvaux, F., Standaert, F.: From improved leakage detection to the detection of points of interests in leakage traces. In: Fischlin, M., Coron, J. (eds.) Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9665, pp. 240–262. Springer (2016)
17. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, Philadelphia, PA, USA, October 25-28, 2008. pp. 293–302. IEEE Computer Society (2008). https://doi.org/10.1109/FOCS.2008.56
18. Faust, S., Grosso, V., Pozo, S.M.D., Paglialonga, C., Standaert, F.: Composable masking schemes in the presence of physical defaults & the robust prob-

ing model. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2018**(3), 89–120 (2018). https://doi.org/10.13154/TCHES.V2018.I3.89-120
19. Faust, S., Pietrzak, K., Schipper, J.: Practical leakage-resilient symmetric cryptography. In: Prouff, E., Schaumont, P. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7428, pp. 213–232. Springer (2012). https://doi.org/10.1007/978-3-642-33027-8_13
20. Fuller, B., Hamlin, A.: Unifying leakage classes: Simulatable leakage and pseudoentropy. In: Lehmann, A., Wolf, S. (eds.) Information Theoretic Security - 8th International Conference, ICITS 2015, Lugano, Switzerland, May 2-5, 2015. Proceedings. Lecture Notes in Computer Science, vol. 9063, pp. 69–86. Springer (2015). https://doi.org/10.1007/978-3-319-17470-9_5
21. Grosso, V., Standaert, F.: Algebraic side-channel attacks against isap's re-keying: one ascon round may not be enough for serial implementations. IACR Commun. Cryptol. **2**(1), 34 (2025). https://doi.org/10.62056/AESGVURZN
22. Longo, J., Martin, D.P., Oswald, E., Page, D., Stam, M., Tunstall, M.: Simulatable leakage: Analysis, pitfalls, and new constructions. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 223–242. Springer (2014). https://doi.org/10.1007/978-3-662-45611-8_12
23. Mangard, S.: Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In: Okamoto, T. (ed.) Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings. Lecture Notes in Computer Science, vol. 2964, pp. 222–235. Springer (2004). https://doi.org/10.1007/978-3-540-24660-2_18
24. Mangard, S., Popp, T., Gammel, B.M.: Side-channel leakage of masked CMOS gates. In: Menezes, A. (ed.) Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3376, pp. 351–365. Springer (2005). https://doi.org/10.1007/978-3-540-30574-3_24
25. Martin, D.P., Oswald, E., Stam, M., Wójcik, M.: A leakage resilient MAC. In: Groth, J. (ed.) Cryptography and Coding - 15th IMA International Conference, IMACC 2015, Oxford, UK, December 15-17, 2015. Proceedings. Lecture Notes in Computer Science, vol. 9496, pp. 295–310. Springer (2015). https://doi.org/10.1007/978-3-319-27239-9_18
26. Masure, L., Cassiers, G., Hendrickx, J.M., Standaert, F.: Information bounds and convergence rates for side-channel security evaluators. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2023**(3), 522–569 (2023). https://doi.org/10.46586/TCHES.V2023.I3.522-569
27. Moos, T., Moradi, A., Richter, B.: Static power side-channel analysis - an investigation of measurement factors. IEEE Trans. Very Large Scale Integr. Syst. **28**(2), 376–389 (2020). https://doi.org/10.1109/TVLSI.2019.2948141
28. Obremski, M., Ribeiro, J., Roy, L., Standaert, F., Venturi, D.: Improved reductions from noisy to bounded and probing leakages via hockey-stick divergences. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 14925, pp. 461–491. Springer (2024). https://doi.org/10.1007/978-3-031-68391-6_14

29. Pereira, O., Standaert, F., Vivek, S.: Leakage-resilient authentication and encryption from symmetric cryptographic primitives. In: Ray, I., Li, N., Kruegel, C. (eds.) Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015. pp. 96–108. ACM (2015). https://doi.org/10.1145/2810103.2813626
30. Pessl, P., Standaert, F.X., Mangard, S., Durvaux, F.: Towards Leakage Simulators that Withstand the Correlation Distinguisher. In: ASIACRYPT 2014, Rump Session
31. Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5479, pp. 462–482. Springer (2009). https://doi.org/10.1007/978-3-642-01001-9_27
32. Prest, T., Goudarzi, D., Martinelli, A., Passelègue, A.: Unifying leakage models on a rényi day. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11692, pp. 683–712. Springer (2019). https://doi.org/10.1007/978-3-030-26948-7_24
33. Rabaey, J., Chandrakasan, A., Nikolic, B.: Digital Integrated Circuits, vol. 2. Prentice hall Englewood Cliffs (2002)
34. Standaert, F.X.: Side-Channel Analysis and Leakage-Resistance. Version 1.2 (September 2024)
35. Standaert, F., Archambeau, C.: Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In: Oswald, E., Rohatgi, P. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5154, pp. 411–425. Springer (2008). https://doi.org/10.1007/978-3-540-85053-3_26
36. Standaert, F., Pereira, O., Yu, Y.: Leakage-resilient symmetric cryptography under empirically verifiable assumptions. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8042, pp. 335–352. Springer (2013). https://doi.org/10.1007/978-3-642-40041-4_19
37. Standaert, F., Pereira, O., Yu, Y., Quisquater, J., Yung, M., Oswald, E.: Leakage resilient cryptography in practice. In: Sadeghi, A., Naccache, D. (eds.) Towards Hardware-Intrinsic Security - Foundations and Practice, pp. 99–134. Information Security and Cryptography, Springer (2010). https://doi.org/10.1007/978-3-642-14452-3_5
38. Uchino, T., Cong, J.: An interconnect energy model considering coupling effects. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **21**(7), 763–776 (2002). https://doi.org/10.1109/TCAD.2002.1013890
39. Yu, Y., Standaert, F., Pereira, O., Yung, M.: Practical leakage-resilient pseudorandom generators. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010. pp. 141–151. ACM (2010). https://doi.org/10.1145/1866307.1866324