*cryptography*

MDPI

*Article*

# How Bad Are Bad Templates? Optimistic Design-Stage Side-Channel Security Evaluation and its Cost

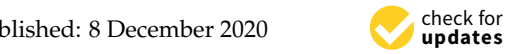
**Rinat Breuer** *,† **and Itamar Levi** †

Faculty of Engineering, Bar-Ilan University (BIU), Ramat-Gan 5290002, Israel; itamar.levi@biu.ac.il

* Correspondence: rinat.breuer@biu.ac.il

† These authors contributed equally to this work.

Received: 6 November 2020; Accepted: 30 November 2020; Published: 8 December 2020




**Abstract:** Cryptographic designs are vulnerable to side-channel analysis attacks. Evaluating their security during design stages is of crucial importance. The latter is achieved by very expensive (slow) analog transient-noise simulations over advanced fabrication process technologies. The main challenge of such rigorous security-evaluation analysis lies in the fact that technologies are becoming more and more complex and the physical properties of manufactured devices vary significantly due to process variations. In turn, a detailed security evaluation process imposes exponential time complexity with the circuit-size, the number of physical implementation corners (statistical variations) and the accuracy of the circuit-simulator. Given these circumstances, *what is the cost of not exhausting the entire implementation space?* In terms of simulation-time complexity, the benefits would clearly be significant; however, we are interested in evaluating the security implications. This question can be formulated for many other interesting side-channel contexts such as for example, how would an attack-outcome vary when the adversary is building a leakage template over one device, i.e., one physical corner, and it performs an evaluation (attack) phase of a device drawn from a different statistical corner? Alternatively, is it safe to assume that a typical (average) corner would represent the worst case in terms of security evaluation or would it be advisable to perform a security evaluation over another specific view? Finally, how would the outcome vary concretely? We ran in-depth experiments to answer these questions in the hope of finding a nice tradeoff between simulation efforts and expertise, and security-evaluation degradation. We evaluate the results utilizing methodologies such as template-attacks with a clear distinction between profiling and attack-phase statistical views. This exemplary view of what an adversary might capture in these scenarios is followed by a more complete statistical evaluation analysis utilizing tools such as the Kullback–Leibler (KL) divergence and the Jensen-Shannon (JS) divergence to draw conclusions.

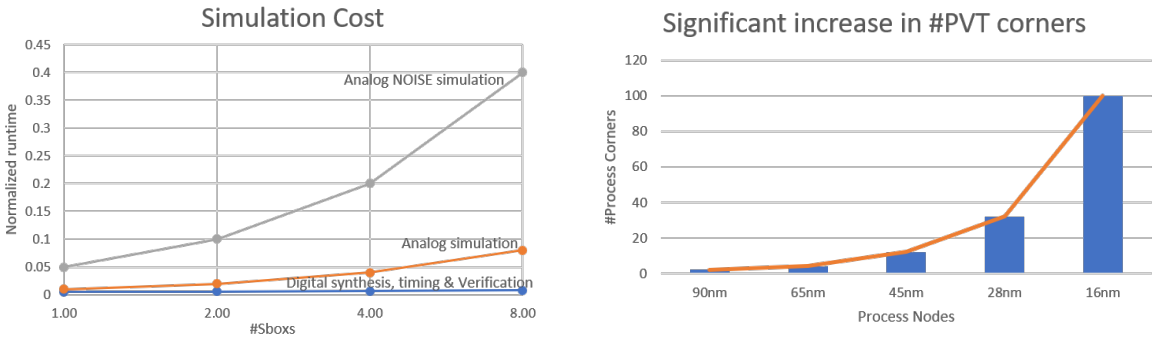
**Keywords:** corners; device mismatch; worst case security evaluation; side-channel analysis; template attacks; simulation; statistical distance

## 1. Introduction

Security evaluation methodologies for cryptographic devices have evolved rapidly to face the rapid rise in side-channel attacks (SCAs). In many organizations they have become mainstream, even in non security-oriented design houses. Specifically, methodologies based on advanced attacks and statistical worst-case evaluation metrics are co-progressing at-speed with countermeasures and the related expertise is expanding in both academia and in industry. However, security evaluation standards have not followed suit, and attacks based on physical information (leakage) are constantly improving through the use of sophisticated attack vectors. The nature of the leakage depends

on the device type but also on the countermeasures embedded within it. Although side-channel attack countermeasures and attacks have attracted considerable attention, a point which is rarely considered in literature is *SCA-security implications related to the statistical nature of the manufactured devices*. This manuscript aims to take a step forward in understanding the security degradation of such statistical behavior, and provide a better understanding of how to approach design-stage security evaluation and its expected time costs.

In the semiconductor industry, companies utilize traditional corner-based signoff methodologies, e.g., by evaluating more standard design metrics such as propagation delay and energy consumption in several distinct manufacturing statistical-corners. Initially, in old manufacturing processes there were only 2 corners, the worst and the best case. As process nodes move to lower geometries, the number of corners has grown exponentially (e.g., consider Figure 1b). Nowadays, for advanced technologies, both analog and digital circuit verification is done by multi-mode/multi-corner (MMMC) analysis, where mode typically implies sets of possible voltages and temperatures and the different corners reflect physical transistors and routing statistical geometry changes. In turn, for advanced system-on-a-chips (SoCs), the verification time increases exponentially while evaluating these sets of conventional design metrics [1].




(**a**) (**b**)

**Figure 1.** Security evaluation time costs: (**a**) Digital, Analog and transient noise Simulation Time vs. circuit size and (**b**) exponential increase in # of physical *corners* versus process technology.

The main difference between a security evaluation and a standard very large scale integration (VLSI) *digital* circuit evaluation, is that in the former case *design-abstraction* is robust and effective. Figure 1a shows the simulation time versus the circuit size verifying a complex design through digital timing/energy checks has a rather low computational (time) cost and this verification is sufficient for traditional electronic designs. However, for security evaluation analysis in the context of SCA attacks, even much slower analog transient simulations are not sufficient to capture the statistical nature of the leakages, the noise level etc. These characteristics play a significant role in security evaluation methodologies and must be accurately evaluated by advanced transient-noise simulations. The gray curve in Figure 1a clearly shows that the cost of this analysis is exponential with the circuit size (x-axis). Here the x-axis is in units of the number of cryptographic 4-bit Sboxes evaluated in this example, i.e., 8 reflects 32 bits of a symmetric-encryption state variable processed by eight 4-bit Sboxes.

Taking both factors into account, i.e., the exponential number of corners (routing corners and devices/transistors corners) and the time complexity of transient noise simulations, it is clear that a detailed analysis would be ***out-of-reach or too expensive*** for many companies and organizations, especially during design stages and given the limited time-to-market constraints.

The specific objective of this manuscript is to initiate a debate relating to the following issues:

I1 What would the hardware-security over/under estimation be in cases of local/global corners?
I2 How bad would a bad template be? That is, if a template was built on one physical realization (corner) and it is being used to attack a device from a different corner, what would the result be?

For a security evaluation would the overly pessimistic approach of using a "perfect" template induce very small security margins which are not statistically meaningful? If the converse is the case it would have unpleasant implications.

I3 During design stages (or pre-fabrication SoC design evaluation), what is the best way to approach a genuine security guarantee? i.e., would a *cornered* adversary be more/less efficient? which corner would typically be more sensitive and would security evaluation results change significantly?

I4 What are the consequences when facing a standard adversary with low means and computational abilities as compared to when the adversary is considered strong and resourceful?

Our experimental results, which can only be simulated, are aimed at providing initial answers to this set of questions. Throughout this manuscript we present the variability effects in the context of SCA gradually, by using more standard template attacks and log-likelihood distinguishers (Although the maximum likelihood distinguisher is not perfect when the noise is not Gaussian), and while doing so we evaluate more statistically informative and visual tools to demonstrate and illustrate different features related to the full distribution of the side channel leakage.

**Paper organization.** We start with a short presentation of security implications of being non-exhaustive in simulations for security evaluation. The delicate balance between simulation-effort and reliable security evaluation is emphasized (Section 2). In Section 2.2 the tools we use for security analysis are described concisely providing the rationale in context of this research. This section also describes the evaluation setup of our simulation environment. Section 3 discusses results relating to the variability caused by the different corners for a leaking device and their effect in a Maximum Likelihood attack scenario. In Section 4 we explore the differences between the leakage distribution of different corners quantitatively in terms of their differences and qualitatively in terms of the magnitude of the effect. This is done by evaluating statistical distance as a metric and in an attack scenario. Finally Section 5 discusses the possible consequences arising from previous sections. Section 6 provides conclusions and discusses future work.

## 2. Motivation: The Cost of Being Lazy or Non-Exhaustive

This motivational section is aimed at discussing the security implications (and cost) of being non-exhaustive in simulations for security evaluation.

Data growth, the increases incommunication traffic and computational requirements are pushing digital devices to scale-down and increase parallelism to make gains in data-bandwidth.These rapid changes have led to challenges relating to the underlying physical layers of silicon devices. To cope with the variability of devices geometry and interconnects (routing) in advanced nodes, a single design is represented with many so-called physical *views*, where each is characterized in a different process *corner* corresponding to a different type of physical fabrication mismatch than the gold standard design (pre-fabrication). Several characteristics such as max-current, delay and power-dissipation are abstracted and are represented with fewer details and accuracy for each corner view, so they can be integrated within digital design-flows and thus contribute to a much faster design cycle. Similarly, software design tools also need to support the growth of such views to reliably capture the statistical distribution of the design characteristics after fabrication (e.g., Cadence's Multi-Mode-Multi-Corner simulations and testing environment, used for both analog and digital representations [2]).

In essence, most companies use such corners views to verify designs in a ***fully-digital abstraction***, rapidly and accurately [3]. Statistical assurance is guaranteed and provided by such views (libraries). Replicating these abstracted flows for SCA security verification is strongly discouraged, though very attractive. Physical design characteristics such as the ones that need to be evaluated for hardware security are not fully covered by digital views to date [4]. Even if some parameters are characterized, e.g., standard cells current models, they are not specifically tailored to reflect security-related sensitivities nor are provided with the required resolution. For example, consider Side-Channel adversarial threats measuring the device's current/voltage under external parameters variations and

process internal variations; these need to be carefully specified and not only statistically bounded, worst-case estimated, etc.

In addition to accuracy and the increased number of corners, cost- and time-related factors are also crucial; attempting to analog-simulate a large post-layout design is the bottleneck for analog verification. However, analog layout-based circuit simulations are the most accurate and reliable choice to validate security features. To date, Analog Spice/FastSPICE-like platforms [5–8], deliver very accurate results, are very comprehensive and suitable for the high-performance verification required for SCA evaluation. They are foundry-certified, accurate and are constantly put to the test for the most advanced chips and technologies. However, the analog verification-time bottleneck is impossible to meet for most companies and designs as technology progresses and design-size increases. Moreover, considering SCA's characteristics, where the effect of physical noise is crucial [9,10], noise-simulations are required. These are far more time and computationally costly.

Thus analog noise simulations are crucial for security evaluations but are very costly. Another challenge is related to the IP; there are concrete scenarios where it is impossible for a company to perform a full analog evaluation of a design. For example if an IP is embedded into its chip, it forces them to perform some digital verification which fail to capture side-channel characteristics. The last aspect to consider is that physical security evaluation criteria, metrics and tools are constantly evolving (e.g., [11–14], for just a few) and it is almost impossible to specify which criteria are robust for all designs, and are comprehensive and exhaustive enough to evaluate hardware security levels for differently protected and unprotected designs. Therefore, the ability to evaluate designs through analog transient-noise simulations **in-house** is still a must.

Figure 2 illustrates the type of security-assurance possible in terms of the practical constraints which impede a detailed, exhaustive and rigorous simulation campaign. A non-exhaustive view of the simulation-space can be attributed to: (a) economical reasons such as the time-to-market requirements of the devices, the cost of a large number of licences for simulation tools and computational or data limitations for small start-ups, (b) digital/analog-IP issues restricting the abstraction of simulations, which is more apparent in software startups, medium size companies and integration-houses and (c) the required expertise which might not exist in all organizations, e.g., analog simulation/designer or hardware security evaluation abilities or knowledge.

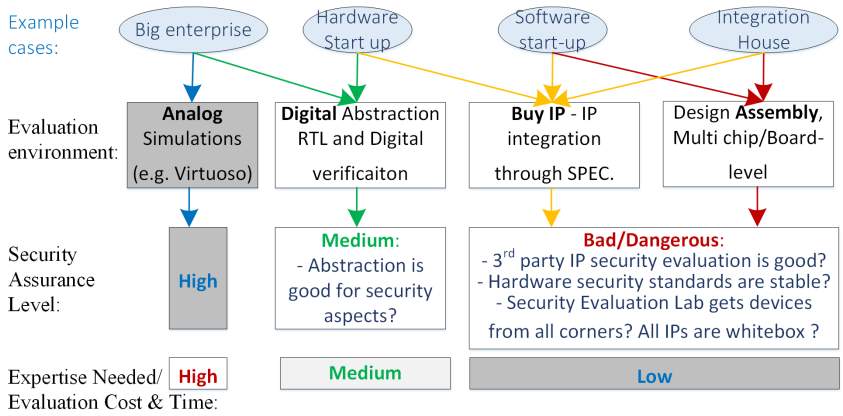



**Figure 2.** Illustration of the SCA security assurance level versus cost and expertise for several 'abstract' example organizations.

***We assess the impact of being* lazy *or non exhaustive in that sense.***

On the one hand large firms may enjoy exhaustive analysis of analog noise simulations across corners using vast resources, knowledge and expertise which yield a high security assurance level. However, most mid-echelon hardware/software companies and start-ups are not able to invest these resources. This in turn can culminate in bad/dangerous security-assurance level, as illustrated in Figure 2.

***From a Security Perspective:***

In sum, by nature, evaluating SCA security requires accurate analog information which can only be obtained by exhaustive statistical simulations (during the design stages). On a positive note, VLSI simulators are ultra-accurate. Theoretically, there is no viable reason why we should not use such accurate analog tools for security analyses. However, as always, cost and resources are limited, it would be preferable to use abstract models for security to reduce these limitations but to date such "hardware-security"-abstraction remain out of reach.

Below we examine whether it is enough to evaluate security but not over the entire statistical-fabrication space. Specifically, (1) how do the security-evaluation results distribute across this space, (2) how would utilizing one device to build a template and attack another device vary statistically, (3) which corners and scenarios would be preferable for security-evaluation with smaller security margins and better assurance; and can we recommend general guidelines for designers.

### *2.1. Security Analysis Tools We Use and Why*

In this subsection we briefly list the main tools we use in this manuscript, as we move from an attack based approach (which is the one most commonly performed in industry) to a statistical analysis view with the aim of visualizing the statistical characteristics of leakages in different corners, etc.

Let $Y$ be an $n$-bit sensitive variable manipulated by a cryptographic algorithm (and let $y$ be its realization). During device operation it leaks information which is associated with data manipulation and physical and/or environmental parameters throughout side-channels. For simplicity we focus on the power-supply current leakage. The security-level of a device corresponds to the abilities of the adversary; thus, as is traditionally done, we assume that the adversary exploits a divide and conquer approach over a sub set of the secret variable of $s$ bits ($s \leq n$). We denote a leakage trace by a set of measurements at $T$ time points, $t_i, i \in \{1...T\}$. The leakage trace is associated with some internal variable manipulated within the hardware, here the manipulation of $y$: $\boldsymbol{L} = \{L_{t1}, L_{t2}, ..., L_{tT}\}$. For simplicity, in what follows we focus on a univariate analysis of the leakage distribution, although, theoretically there are scenarios which strictly require multivariate analysis, e.g., when *shuffling* [10] or Masking [15,16] countermeasures are embedded. In our simulation environment we incorporate no such countermeasures, thus making a univariate analysis natural to respond to the goal of first answering more inherent questions and leaving room for further investigation.

*Signal to Noise Ratio, SNR:*

The SNR in the side-channel sense, which was first proposed by Mangard [17] and utilized in numerous works, indicates 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) for each $y$ state, and averages over the states. Formally, the SNR is defined as:

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

Note that because the univariate SNR and DPA attacks utilize some simplifying statistical assumptions regarding the leakage distribution, it still serves as a viable tool to identify points-of-interest (POI) in time when the manipulation of a secret variable takes place. In turn, it constitutes a tool for valuable speedup in security evaluations.

#### 2.1.1. Template Attacks

Template attacks [18] are performed in two subsequent (or interleaved) phases of *profiling* and *attack*. It is assumed that the adversary got hold of one device for which it can program (or control) the

secret key and therefore profile the leakages and another target device from which it tries to extract information on the underlying key. Note that an attack procedure can be done simultaneously with measurements (on-line).

Let $l \mid y_i$ (or $l_{x_i,k}$ for e.g., an Sbox output, $y_i$) be a leakage trace measured while the value of $y_i$ is processed within the device. To perform a template attack, one chooses a target intermediate variable to template the probability density function (PDF), $\mathrm{f}(x_i, k)$, associated with a known/chosen plaintext chunk $x_i$ and secret key subset $k$. A set of $\mathcal{L}_p$ profiling traces of size $N_p$ is first used in order to estimate the leakage distribution parameters for each intermediate value of $y$, denoted as $\hat{M}_y$. For the attack phase, a fresh set of new traces from a different device $\mathcal{L}_{att}$ of size $N_{att}$ are used. In most cases in the literature (and in practice) the PDF is assumed to follow a Gaussian distribution, $\mathrm{f}(l \mid y) = \mathcal{N}\left(\widehat{\mu}_{l|y}, \widehat{\sigma}_{l|y}\right)$. In practice, distributions from ASICs are not normal and do not follow nice leakage functions (e.g., Hamming-weight HW); we illustrate such cases in Section 3. However, the simplicity of such a model, low computational effort and speed makes it a very popular *de-facto* tool. Finally, the secret key (byte/chunk) $k^*$ which maximizes the univariate Maximum Likelihood (denoted by LH) is chosen:

$$k^* = \underset{k}{\operatorname{argmax}}\, \mathrm{LH}(k) = \underset{k}{\operatorname{argmax}} \prod_{j=1}^{N_{att}} \left(\mathrm{f}\left(l_i \mid y_i\right)\right). \tag{1}$$

For practical computational reasons and numerical errors, the log-likelihood (LLH) is used [19] $k^* = \underset{k}{\operatorname{argmax}}\, \mathrm{LLH}(k) = \underset{k}{\operatorname{argmax}} \sum_{j=1}^{N_{\mathrm{att}}} \log\left(\mathrm{f}\left(l_i \mid y_i\right)\right)$.

Note that any other {Distinguisher, Model}-pair can be used instead of the ML distinguisher and the a-priori Gaussian modeled probabilities, e.g., a correlation distinguisher with a Hamming-Distance distinguisher or a Moments-Correlating Profiled DPA (MCP-DPA) which are very useful when masked implementations are considered with leakages stemming from higher statistical moments [20].

As done commonly, in this manuscript we build templates for manipulation of $y$; however, group leakage measurements which correspond to the $y$ value, e.g., the Hamming-Weight $\mathrm{W}(y)$, instead of $y$ (identity) can considerably reduce time, memory and complexity of the analysis.

#### 2.1.2. Mutual Information

The Mutual-Information (MI) metric [21,22] is traditionally used to quantify the amount of information an adversary can capture from a leakage trace. It can be used to approximate the attack success rate (SR) with multiple measurements.

$$\mathrm{MI}(Y; \boldsymbol{L}_Y) = \mathrm{H}(\mathrm{Y}) - \sum_{y_i \in \mathrm{Y}} \Pr(y_i) \cdot \sum_{\mathbf{l} \in \boldsymbol{L}_\mathrm{Y}} \tilde{\mathrm{f}}_{dua}(\mathbf{l}|y_i) \cdot log_2(\mathrm{f}_{prof}(y_i|\mathbf{l})). \tag{2}$$

where $\mathrm{H}(\mathrm{Y})$ is the entropy of the sensitive variable Y. The conditional probability, $\Pr(y_i|\mathbf{l})$, can be computed by Bayes' theorem as done in [21,23]. The subscripts *prof* and *dua* are used to distinguish the profiled setting in which the leakage distribution is characterized over one device, to obtain a profiled model, *prof*; however, the attack phase (secret extraction), is done over the other device under evaluation, *dua*. Because leakage distributions are continuous (can never be exhausted), there are model-estimation errors (e.g., histograms and kernels) the theoretical MI can never compute (only estimated), so that the perceived information is what we typically compute [14].

However, even-though this an analysis is the most "theoretically" sound approach, information theoretic based metrics are computationally hard to evaluate. Therefore the main tool we use in this work is a measure of the statistical-distance which is highly associated with the MI but easier to compute and provides a more visually easy interpretation. We admit that for an actual rigorous analysis information theoretic metrics can be evaluated. However, we stress that according to our

examination here, this should only be done for specific scenarios and corners so as to reduce the computational effort associated with it.

#### 2.1.3. Probability Distance Measures

The main questions we tackle relate to the differences between leakage distributions of the profiling versus attack trace-sets and the different process-corners of the device in each of them. Therefore, the most natural tool to use for evaluation is a probability-distance measure. The relative entropy or the Kullback–Leibler (KL) divergence [24] is one measure of the probability distance between two distributions, though it is not a metric and is not symmetric:

$$D(\mathrm{X}\|\mathrm{Y}) = \sum_{z} \Pr(\mathrm{X} = z) \log_2 \left( \frac{\Pr(X = z)}{\Pr(Y = z)} \right)$$

where X and Y represent two PDFs over the shared support ($z \in Z$).

The Jensen–Shannon divergence (JSD) is a smoothed and symmetric transform of the KL divergence: $D_{JS}(\mathrm{X}\|\mathrm{Y}) = D(\mathrm{X}\|\mathrm{M}) + D(\mathrm{Y}\|\mathrm{M})$, where M is the arithmetic mean of X and Y, M = 0.5 (X + Y). As the measure is weighted with probabilities, it is better suited here than distance measures such as the norm-1 distance (or total variation distance) (Note that other metrics exist for the Statistical-Distance (SD) (e.g., see the discussion in [25,26] and in the side-channel context in [12]).). Note that the probability distance and more generally, the Statistical Distance (SD) are actually tightly related to the MI, $\mathrm{MI}(Y; L) = D(\Pr(y, l) \| \Pr(y)\Pr(l))$.

In this manuscript we use the divergence and the visual representations of the PDFs at a time point-of-interest (POI) to inspect, justify and exemplify the challenges of using one simulated scenario for generating profiles (e.g., a corner), versus possibly a different one inflicted upon an adversary/evaluator while conducting an evaluation or executing an attack campaign.

These measures are very efficient to compute and fast for design-stage evaluation (i.e., to understand which corner we need to focus on). They also provide intuitive and visual insights for designers.

### *2.2. Evaluation Setup*

In order to extract the experimental dataset a simple simulation environment is required to target a circuit that is as small as possible; as discussed above, transient noise simulations over a very large number of statistical corners and experimental tests take quite long time for high accuracy. The chosen synthesized design embeds a simple 4 bit Sbox of the present cipher [27] along with peripheral and control circuitry. The chosen 65 nm technology is highly mature and commonly used. Standard-cells from a library characterized for supply voltage of 1.2 V were chosen. The design was imported to an analog simulation environment and the current consumption of the main supply was measured for each simulation. In each simulation a key-addition followed by the Sbox was presented with different pairs of keys and plain texts. For the analysis presented in this paper we ran the simulation with 5 representative corners, each with a different process corner $\in$ {TT, FF, FS, SF, SS} due to deviations in the semiconductor fabrication process. Here, {T,S,F} stands for the typical, slow and fast statistical corners of devices, respectively. The first element represents n-MOS devices and the second p-MOS devices.

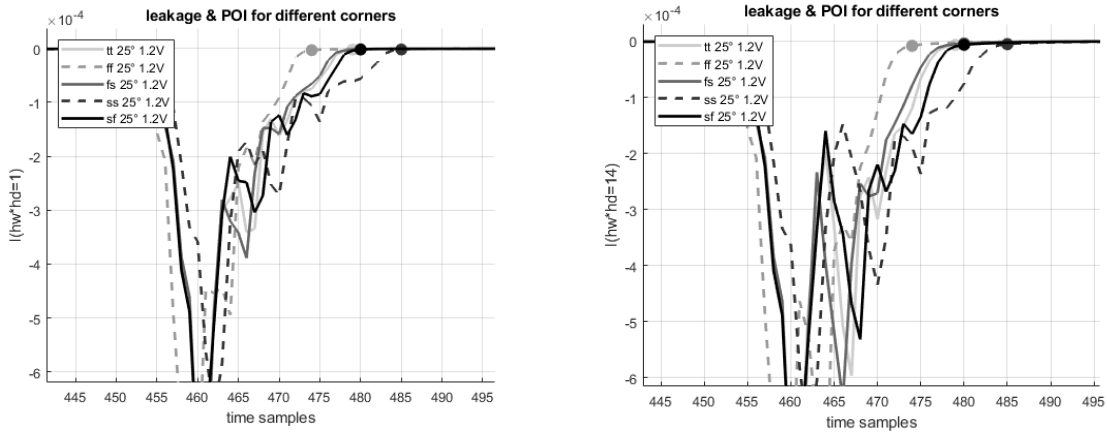
For template creation and the *dua* attack campaigns discussed in this manuscript we collected 3000 traces in each run, and 50 multiple runs (repetitions) with transient-noise simulations with very high accuracy and low tolerances. This yielded a total of 150,000 traces (per corner). Each trace of 900 time samples incorporated the serial processing of different $\{x_i, k\}$ pairs. Each of the pairs was processed within a pipeline of 3 clock cycles where the Sbox output is computed in one stage of the pipe. The clock frequency was set to 1Ghz and the sampling frequency was set to 100 GHz, yielded 100 samples per cycle. Of the 9 cycles in a trace, 7 contained computations of the Sbox (after pipeline-fill). Note that with 65 nm and a relatively small 4-bit Sbox design 1GHz represents a not-too-tight timing,

and that very large timing slacks were present to enable correct functionality even in the worst possible corner. Overall, our data set had $1.05 \times 10^6$ traces per corner, and $5.2 \times 10^6$ traces in total. Note that producing such a high-accuracy data-set from simulations is quite a different task than doing so from actual measurements, since the latter can be done in a day/s whereas the former can easily take months even for such small circuitry (depending on the resources: #simulator licences and servers strength).

The profiling phase involved classification according to multiple grouping functions; namely, the Hamming-Weight, HW, Hamming Distance, HD, and their multiplication, HW·HD of the Sbox's output for each of the intermediate variables in each trace. The leakages were categorized by these classes as traditionally done and the templates were evaluated per category. Note that we only illustrate results from the HW·HD classification which was the most informative for the process-technology discussed.

## 3. Variability of Template Attacks

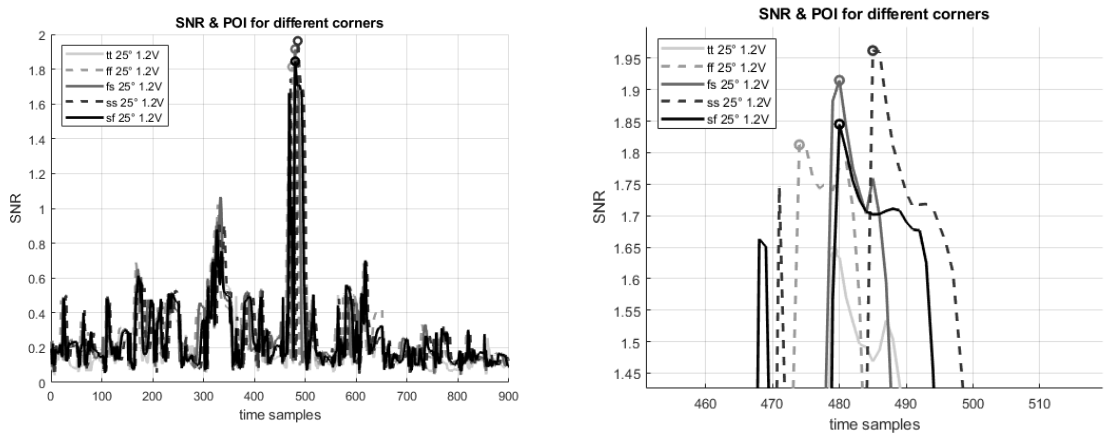
After post-processing the leakage traces from the simulation, the POI of each corner could be extracted by using the SNR. When looking at a trace with HW·HD = 1 ($l|HW \cdot HD = 1$) Figure 3a, as expected, it is clear that the leakage was significantly different for each corner, there are corners for which the leakage was greater than others owing to the different physical realization of the devices in each. In Figure 3b a leakage traces of HW·HD = 14 ($l|HW \cdot HD = 14$) is shown for the different corners; it is apparent that there are much greater differences between the corners' traces due to the greater logical activity which sums to greater differences between corners. That is, there were 3 signals changing from 0 to 1, and each drew a current affected by more process mismatched elements, whereas in the previous case only 1 signal changes from 0 to 1.




(**a**) (**b**)

**Figure 3.** Leakage for different y values for each corner, with marked POI values: (**a**) leakage trace for HW·HD = 1, (**b**) leakage trace for HW·HD = 14.

The SNR shown in Figure 4a illustrates the peak at the SNR, which was used as our POI; these values are also presented in Figure 3. It is clear in Figure 4b, with a zoomed-in view of the SNR peak, that the POI of the corner $\{FF\}$ appears in earlier time samples compared to the other corners, and that the POI of $\{SS\}$ corner is naturally the last (the slowest devices), and that the other corners' POI locations only vary slightly. The results are exactly as anticipated since in the $\{FF\}$ (resp. $\{SS\}$) corner we expected to capture faster (resp. slower) responses of the circuits to input changes. Note as well that the highest SNR value is different for each corner, and that for corners $\{SS\}$ and $\{FS\}$ the SNR(POI) provides a higher value as compared to the other corners. This implies that these corners leaks more information (if the noise is actually normally distributed) in these specific points.



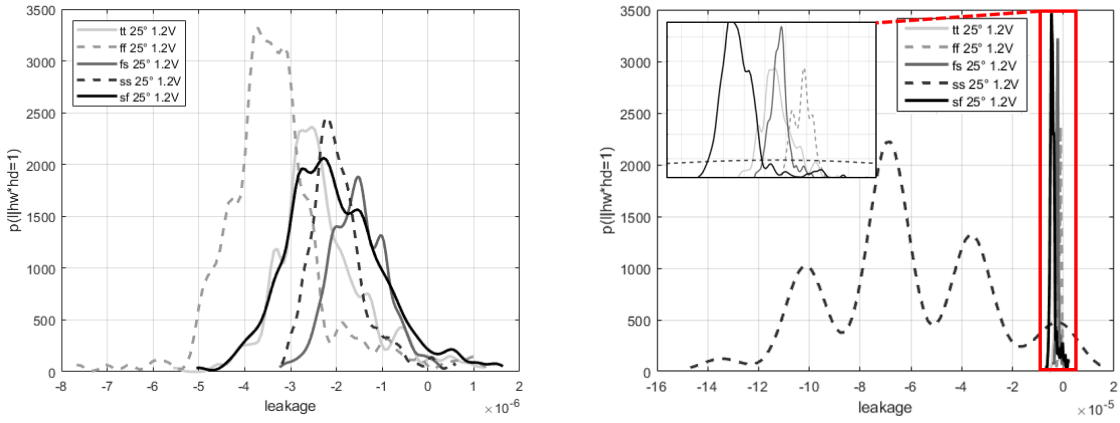

(**a**) (**b**)

**Figure 4.** SNR for each corner with marked POI: (**a**) zoom out, (**b**) zoom in.

To identify the variability between the different corners let us examine the leakage distribution graph. Clearly, this is a more informative representation as it captures the full distribution ($\Pr(l|HW \cdot HD)$) and not only anticipates informativeness from simplified metrics like the SNR (which only makes sense when the noise is Gaussian). We can see that for HW·HD = 1, when taking the distribution for each corner in the corner's POI Figure 5a, the leakage values and distributions are very different from each other. Furthermore, when looking at the leakage distribution for the same value of HW·HD = 1, but now using only one corner's POI (in this example the $\{TT\}$ corner), the distribution map varies significantly. That is, Figure 5b shows that the distributions are completely different now. This view represents a case where an adversary utilizes the POI from one device, used for a template, in an attack campaign against another device from a different corner. Clearly, it faces a much tougher scenario. Relatively speaking, the POIs vary considerably as well as the leakage "model".

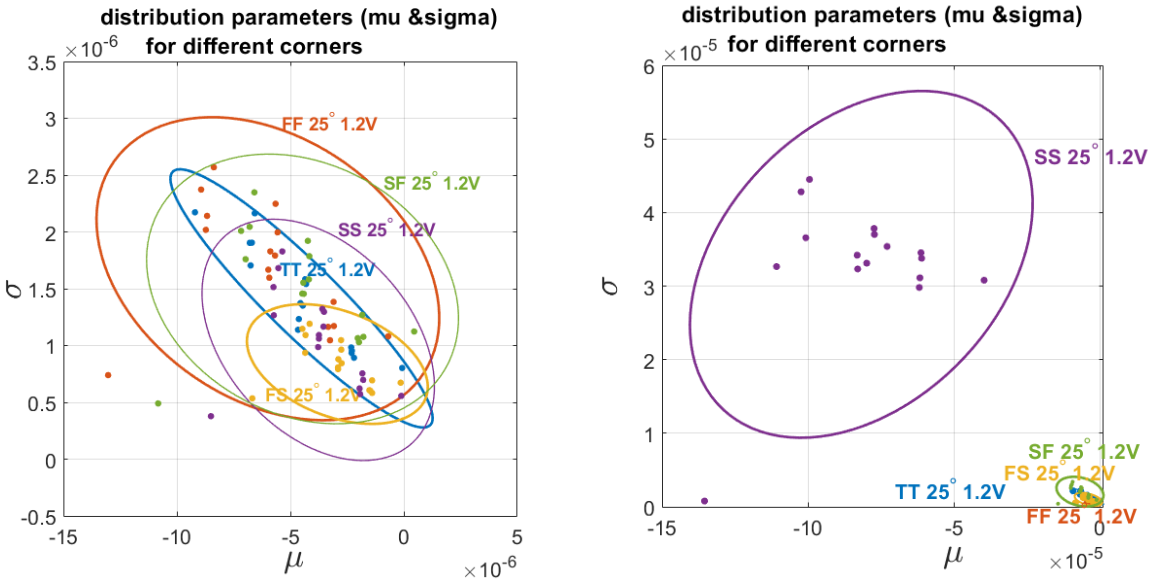
This phenomenon can be explained by the SNR depicted in Figure 4b, in which the POI of each corner is marked. It is clear that for $\{FS, SF, TT\}$ corners, the POIs approximately appear in the same time sample, so for those corners we expect to get similar results following an attack. For example for the $\{SS\}$ corner, the POI appears later on the graph, but it still correspond to the $\{FS, SF, TT\}$ peaks. This in turn implies that when using the template of the $\{SS\}$ for those corners, we still expect to achieve a decent outcome in the attack results. The converse is not identical since the POIs of these corners does not match the peak of the $\{SS\}$ corner's SNR which implies that we will fail or obtain poor results in an attack campaign of a $\{SS\}$ device with a template of a $\{FS, SF\}$ or $\{TT\}$ device. In terms of the $\{FF\}$ corner, the POI appears earlier in the SNR graph, and is not correlated with the peaks of the other corners, so when using a $\{FF\}$ device template to attack a device of any other corner we expect to obtain poor results, etc.

We further examine the variability between the corners by looking at the mean, $\mu$, and standard deviation, $\sigma$, of the leakages for each corner, at the POIs of each corner. We do so for different values of $HW \cdot HD$, as shown in Figure 6a. These values are used to create templates for the maximum-likelihood attack. Figure 6b illustrates the parameter distributions while using the $\{FF\}$ corner POI to profile devices drawn from other corners. It is worth noting that there are situations in which there is no overlap of the model parameters with the different corners. This can result in inaccurate results when attacking a device using one template with a different implementation realized statistically in different corners.




(**a**) (**b**)

**Figure 5.** leakage distribution for HW·HD = 1, for each corner with different POI sources: (**a**) using the corner's POI, (**b**) using the $\{TT\}$ corner's POI.




(**a**) (**b**)

**Figure 6.** Mean and variance of the leakage for each y value: (**a**) using each corner's POI, (**b**) using the $\{TT\}$ corner's POI.

Thus overall, the variability between the corners is significant and concerning. The leakage distributions for the different corners are not the same, and leads to a greater effect when examining distribution for higher HW·HD values. Temporal effects also makes it significantly more challenging in the SCA context; POIs used for one corner may easily result in unsuccessful attacks. But the more alarming feature is the concrete change in the shape of the leakage distribution, implying that a simple univariate sweep with one model (corner) over the attack-set leakages of a different corner will be futile.

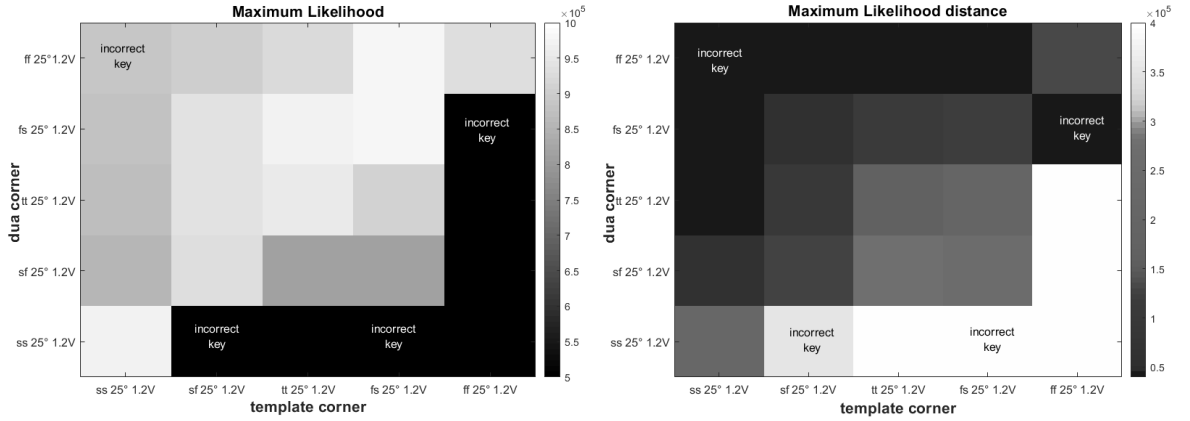
The next section discusses the perspective of an SCA adversary but also the perspective of a design-stage security evaluator that aims at reduce the cost of an exhaustive examination of the entire simulation space. In other words, we aim to capture what would reflect a minimal evaluation set with high certainty.

### *3.1. An Attack Perspective: Maximum Likelihood*

In this subsection we describe an exemplary attack campaign utilizing the ML distinguisher using our generated profiles over the entire corner-space.

In each attack campaign we took a different pair of template-corners and an attacked device corner, under the assumption of a normal distribution for the template's leakage distribution which was generated by using the standard procedure of profiling mean-variance pairs for the template devices (corners). For this experiment, the POI of the device under evaluation, *dua*, was chosen to be the POI of each of the templates used to attack; namely a (simplified) space of $5 \times 5$ scenarios. Note that we only illustrate a subset of the actual results for readability and to better convey our main message. In fact, we performed evaluations with different voltages, temperatures and parasitic RC corner scenarios as well.

Figure 7a shows the maximum values for the ML attack, where labels on the figure indicate whether the attack succeeded. For some pairs whose corner pairs are at extremes to each other, the attack failed. For example, for $\{SS\}$ template corner and $\{FF\}$ *dua* corner we obtained an incorrect key result (black region). Note the non-symmetric map of failures: one might expect that "remote" corners would fail equally but, sometimes mismatch effects were stronger for faster *vs*. slower corners than for slower *vs*. faster corners.




(**a**) Maximum likelihood for the correct key

(**b**) Maximum likelihood distance from the correct to the second-best key

**Figure 7.** Maximum Likelihood results for the different profiling *vs*. *dua* corners.

The figure illustrates a rather systematic behavior where mostly maximum values appear on the diagonal of the graph, which is reasonable since the distribution of a template with a specific corner is the most similar to the distribution of the *dua* in the same corner. There is one special case where a device is attacked from a $\{FF\}$ corner with the template of the $\{FS\}$ corner: it indicates that there may be cases where attacking a device from a different corner may give away more information than when attacking with the same corner. This is a problematic scenario from an evaluator's perspective, and these are exactly the cases we would like to pinpoint.

In Figure 7b the ML distance between the value achieved with the correct key and the second-best key is shown. Figure 7a shows that the distance on the diagonal is high for the lowest ML maximum value, which makes sense. However, there are some extreme distance values on the right column and the lowest row, but these also achieve the worst ML max value.

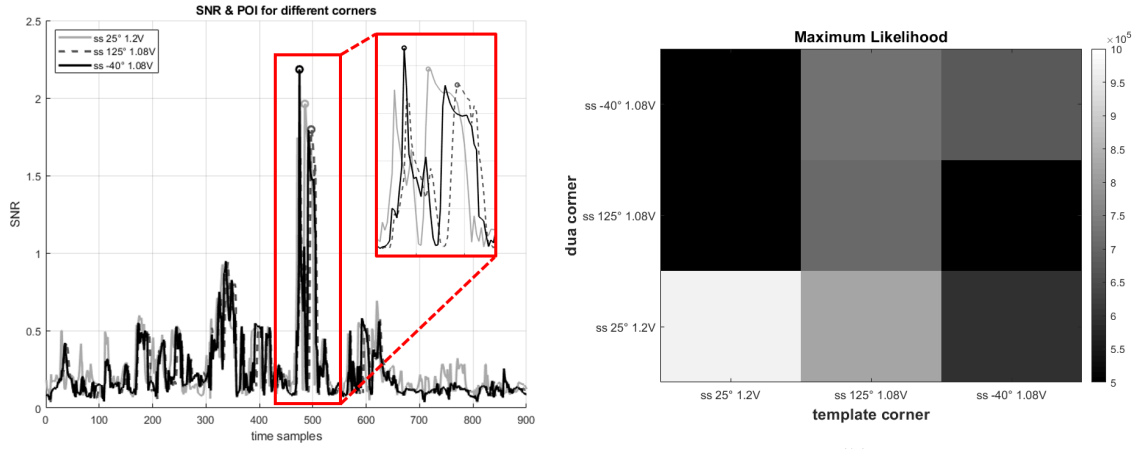
### *3.2. Environmental Views*

It is clear that environmental views play a major role in increasing the side-channel signal, for example by increasing the power-supply voltage of a device which leads to larger currents and

thus a larger SCA-signal. In addition it is also clear that it is possible to tailor environmental factors to concretely reduce the noise level of the measurements, for example by lowering the temperature.

As will be shown next, our investigations indicate that the $\{SS\}$ corner is perhaps the most sensitive statistical *view*. Since we would like pre-select specific profiling-*dua* pairs for security evaluation, inquired whether the intuition from the previous paragraph would be borne out through experimentation.

Figure 8a shows the $\{SS\}$ corner. We first evaluate the SNR for different *modes*; i.e., temperatures and voltage levels. The POIs of each mode is depicted in the figure. Counter-intuitively, the low-temperature and low-voltage $\{SS\}$ mode resulted in the highest SNR(POI) value, and the lowest value was achieved for high-temperature with low-voltage which actually fits the intuition above. Figure 8b presents the results of a ML attack campaign for these different modes. These results fit nicely on the diagonal to what is captured by the SNR values in Figure 8a. However, off the diagonal it is shown that generally increasing the voltage increases ML values as well as reducing the ambient environmental temperature.




(**a**) (**b**)

**Figure 8.** Maximum Likelihood results for various temperature and external voltage conditions: (**a**) SNR for each *mode* (**b**) Maximum-Likelihood values.

The most crucial point to discuss is that monitoring environmental parameters by an evaluation lab is indeed an easy task (e.g., by using temperature chamber and regulated power supply), regardless of the fact that in many scenarios this type of control is beyond the abilities of the adversary. For concrete security-evaluations, the significant effects of these factors must be considered both in design stages and post-fabrication. Therefore, reducing the evaluation-space complexity to only a few *modes* is a key challenge.

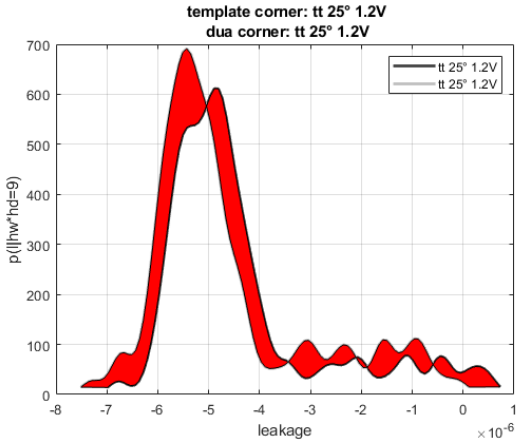
## 4. Statistical Evaluation Utilizing Statistical Distance Measures

In this section we report statistical distance measures to examine the complete distribution differences among corners. In this form of evaluation we do not need any assumptions relating to the shape of the distribution (unlike in maximum likelihood). This should provide a better and more accurate understanding of the variability and serve to evaluate the effect on the informativeness of one corner-view on another, regardless of the statistical tools or assumptions used by the adversary. Before we look at the JS-distance results we examine the distribution of the leakages for different pairs of corners for template- and *dua*-pairs.

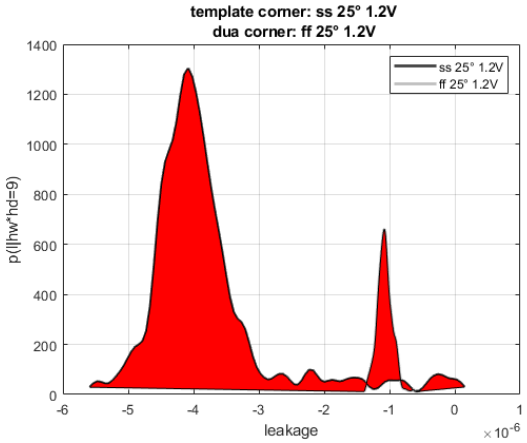
- Figure 9a represents the baseline, where we illustrate both typical-typical (nominal) distributions from the attack and profiling sets. It is clear that the distributions are almost identical.

The differences are natural and due to the different numbers of traces in each set and noise factors which can be improved by longer phases (more measurements).

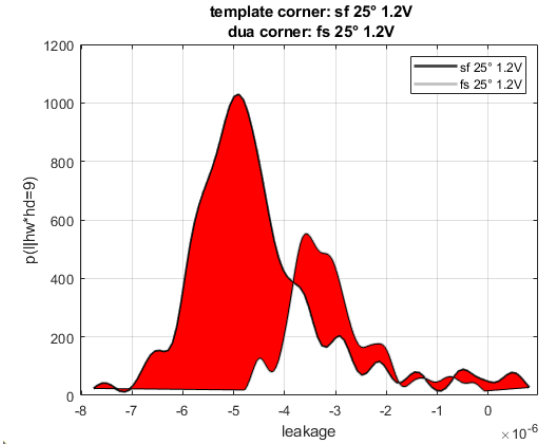
- Figure 9b illustrates a case where the profiled device is in the $\{SS\}$ realization corner, and the attacked device is in $\{FF\}$. These corners manifest in extremely different (remote) distributions, both in shape and amplitude.
- Figure 9c illustrates a case where the profiled device is in $\{SF\}$, and the attacked device is in $\{FS\}$. We expect the distributions to only vary slightly since these corners behave relatively the same e.g., when using tools such as the SNR. However, the statistical distance (difference between the plots) is also considerable.
- Finally, Figure 9d illustrates a $\{TT\}$ corner and a $\{FF\}$ corner distribution. Clearly and intuitively this case reflects a difference midway from both extreme cases in Figure 9a,b.



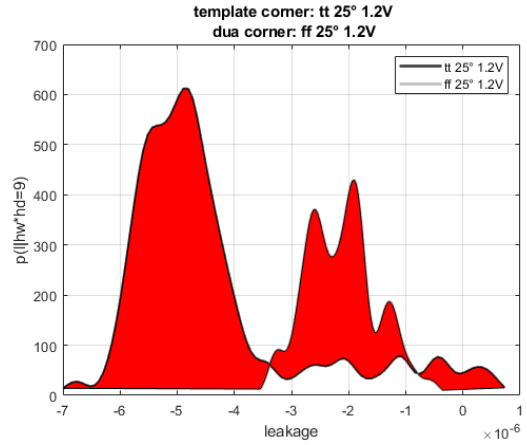

(**a**) distribution of HW·HD = 9 for TT vs TT corners




(**b**) distribution of HW·HD = 9 for SS vs FF corners



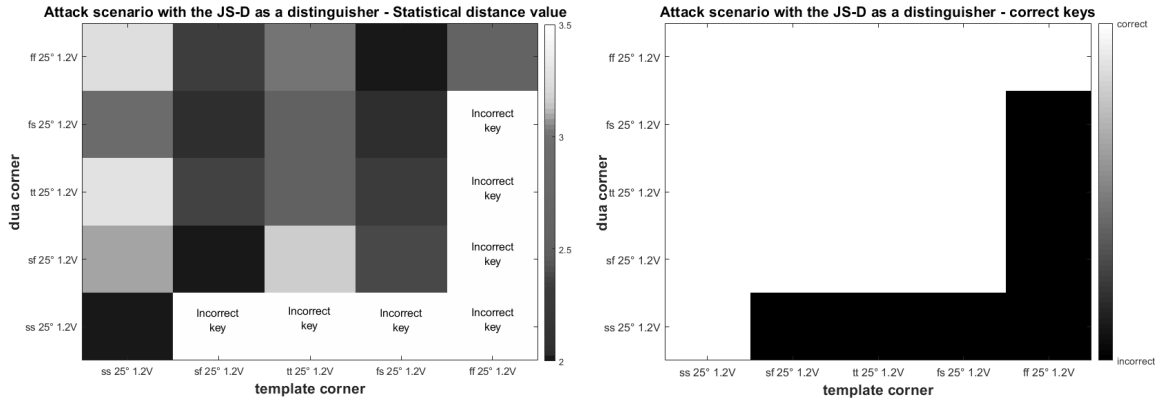

(**c**) distribution of HW·HD = 9 for SF vs FS corners




(**d**) distribution of HW·HD = 9 for FF vs TT corners

**Figure 9.** Comparison of leakage distributions between different corners. The statistical distance between the distributions is shown in red.

In the following we consider Figure 10a, where the JS statistical divergence results for the same $\{\text{profile, dua}\}$-coordinates are illustrated. In addition the JS-Divergence, $D_{JS}$, was also used as a distinguisher in an attack scenario. In other words, we looked for the key $k^*$ which minimizes the distance $k^* = \underset{k}{\text{argmin}} D_{JS}(\Pr(l|k)||\Pr(l|k_{correct}))$. Figure 10b shows whether the attack succeeded or not for the different pairs. An attack that failed clearly indicates that $D_{JS}$ for the correct key was greater than for the incorrect key. This information is used to complement Figure 10a.




(**a**) (**b**)

**Figure 10.** JS-Divergence results: (**a**) JS-Divergence per {profile, *dua*} corner (**b**) $D_{JS}$ used as a distinguisher - successful attack indicated in black.

Figure 10b indicates that the distinguisher always fails when utilizing any non $\{SS\}$-template with an $\{SS\}$-*dua*. It also always fails when utilizing $\{FF\}$-templates to tackle a non $\{FF\}$ realization. This extreme effect corresponds to the explanation relating to the quite substantial drift of the POI and the distribution share for 'remote' corners (i.e., it relates to the SNR and distribution considerations above). Figure 10a shows that as we get closer to the upper left corner or the bottom right corner the values of $D_{JS}$ get higher. There are some cases though in which the minimum values are not on the diagonal. Recall that by intuition an adversary would want to use the same profile corner as the *dua* corner, and as an evaluator this is typically what is done (especially in academic pre-fabrication studies). However, there are several contradictory cases: the usage of an $\{FS\}$ or $\{SF\}$ profile when attacking a $\{TT\}$ device, and another example, when using an $\{FS\}$ profile to attack an $\{FF\}$ device.

The last interesting point relates to the location of the lowest value which corresponds to the softest-point for an adversary to exploit and the choice of interest for an evaluator to perform investigation. It turns out that in fact that point is found using an $\{SS\}$-profile and the *dua* pair. Recall that in the zoomed-in SNR Figure 4b, the $\{SS\}$ corner exhibited the highest SNR value in its POI, which corresponds to the $D_{JS}$ results.

Hence an evaluator that evaluated security with $\{TT\}$-leakages would obtain an optimistic evaluation whereas the (e.g.,) $\{SS\}$ corner is the wanted pessimistic worst-case evaluation point.

## 5. Discussion and Risk Taking

These findings have a number of implications in the context of physical security-evaluation during design-stages or during post-fabrication testing, prior to device-dispatch to the field (e.g., testing machines). The discussion below is linked to the list of questions from the introduction section ({I1:I4}).

### *5.1. Optimistic Security Evaluation: A Bad Template Might Be Really Bad*

I1 and I2: As shown above, different corners can exhibit very different POIs, considerably different leakage distributions and different signal levels. However the most alarming point is that it is hard to anticipate which set will serve as the worst-case for evaluation. ***The last thing a manufacturer would want is that the security of a device will depend on the realization corner it was manufactured in***. Therefore, a bad template used for security evaluation might conclude in a considerably easier attack. Practically, the results indicate that some profiles used for security evaluation can overestimate security and that quantitatively the differences are not small or negligible.

### *5.2. False-Negatives*

I4: Consider the case where a device used for profiling was from an $\{FF\}$ distribution while the attack campaign was performed on any other *dua*-corner. An evaluator might think the system is secured when producing wrong most-likely keys (say with a ML distinguisher); however, it will in-fact represent a false-negative for a large set of other devices.

### *5.3. Cost of Design-Stage Security-Evaluation and Security Margins*

I3: In both evaluation scenarios we investigated, i.e., maximum likelihood and the Jensen-Shannon Divergence, for some cases we achieved a better result for the attacking devices in different corners than devices in the same corner. However clearly the results were not always consistent for both attacks, and in-fact we cannot guarantee that an adversary using another statistical distinguisher, another technology or implementing countermeasures within the device would identify the same sensitivities we have seen, as indicated in the same profile-*dua* pairs. However, it is evident that even if all the elements over the "diagonals" of both the ML and $D_{JS}$ distinguishers are not equally successful, checking all their elements will cover most security-critical points (i.e., a template corner versus the same *dua* corner). Clearly, the complexity is only $c$ (where $c$ denotes the number of corners), as compared to an exhaustive analysis of $c^2$ complexity. If a safe guard is implemented to capture cases such as the case where $\{SF\}$ extracts information better from $\{FF\}$ (e.g., consider Figure 10a), our results indicate that elements close to the diagonal by single shifts will cover all thoese cases. The complexity in this scenario would be $3c$ (for large $c$s'). Note that standard $c$ values can reach 100 for an advanced technology (e.g., 16 nm).

### *5.4. Cost of Post Fabrication and Pre-Distribution Testing*

On testing machines, there are methodologies to screen devices from the worst corners which jeopardize standard electronic characteristics such as performance (or timing constraints). However, it would also be possible to screen devices which correspond to a security-critical corner which might be entirely legitimate from a digital performance perspective (e.g., $\{FS\}$ or $\{FF\}$ corners).

It is crucial to note that examining a TT-corner *vs*. TT-corner alone (as typically reported in the literature), is not enough. Recall the low SNR value compared to other corners.

## 6. Conclusions and Future Work

In this paper, the consequences of statistical variations of manufactured devices on SCA security evaluation were investigated. As process nodes make technological advances the effect of different process corners on a device is greater, making it hard to ignore when discussing SCA security evaluation; the tradeoff between simulation efforts and security evaluation must be taken into account. We evaluated security estimations considering different device corners, to better understand the effect of using a bad template when evaluating security, and assessing whether there are sensitive corners exist which will give an adversary a significant advantage or lull an evaluator into a false sense of security.

After examining the results of several attack scenarios, we showed that using a bad template for security evaluation can paint an optimistic view that departs from reality, depending on the corner of the attacked device. Furthermore an attacker might have much easier job when attacking a device if it has access to a device with a leakier corner, or that the template device and attacked device are from a pair of corners which extract better information than the corner pair used for the security evaluation.

Many different adaptations, tests, and experiments could be investigated in the future. Future work could look at evaluating security with protected devices, by conducting multi variate analysis in cases which require them, and more importantly by examining the effects for different technologies and scaling effects, e.g., older nodes as well as more advanced nodes such as 16 nm and 7 nm and predictive 5 nm models, and finally with different process technologies such as FD-SOI.

**Author Contributions:** Conceptualization, I.L. and R.B.; methodology, I.L.; software, R.B.; validation, R.B.; formal analysis, R.B.; investigation, R.B.; data curation, R.B.; writing–original draft preparation, I.L. and R.B.; writing–review and editing, R.B. and I.L.; visualization, R.B.; supervision, I.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lewyn, L.L.; Ytterdal, T.; Wulff, C.; Martin, K. Analog circuit design in nanoscale CMOS technologies. *Proc. IEEE* **2009**, *97*, 1687–1714. [CrossRef]
2. Cadence Multi-Mode Multi-Corner Timing Closure. Available online: https://www.cadence.com/content/cadence-www/global/en_US/home/multimedia.html/content/dam/cadence-www/global/en_US/videos/tools/digital_design_signoff/multi_mode_multi_corner_timing_closure (accessed on 1 September 2020).
3. Cadence Digital Full Flow. Available online: https://www.cadence.com/en_US/home/tools/digital-design-and-signoff.html (accessed on 1 September 2020).
4. Knechtel, J.; Kavun, E.B.; Regazzoni, F.; Heuser, A.; Chattopadhyay, A.; Mukhopadhyay, D.; Dey, S.; Fei, Y.; Belenky, Y.; Levi, I.; et al. Towards secure composition of integrated circuits and electronic systems: On the role of EDA. *arXiv* **2020**, arXiv:2001.09672.
5. Cadence Analog Modeling and Simulation with SPICE. Available online: https://www.cadence.com/en_US/home/training/all-courses/85086.html (accessed on 1 September 2020).
6. Cadence Spectre Simulation Platform for Block-level, Chip-Level, and Mixed-Signal Simulation. Available online: https://www.cadence.com/ko_KR/home/tools/custom-ic-analog-rf-design/circuit-simulation/spectre-simulation-platform.html (accessed on 1 September 2020).
7. Cadence Spectre eXtensive Partitioning Simulator (XPS) Delivers up-to 10X Faster Simulation Throughput. Available online: https://www.cadence.com/ko_KR/home/tools/custom-ic-analog-rf-design/circuit-simulation/spectre-extensive-partitioning-simulator-xps.html (accessed on 1 September 2020) .
8. Analog FastSPICE (AFS) Platform from Mentor. Available online: https://www.mentor.com/products/ic_nanometer_design/analog-mixed-signal-verification/analog-fastspice-platform (accessed on 1 September 2020).
9. Levi, I.; Bellizia, D.; Bol, D.; Standaert, F.X. Ask Less, Get More: Side-Channel Signal Hiding, Revisited. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 4904–4917. [CrossRef]
10. Levi, I.; Bellizia, D.; Standaert, F.X. Beyond algorithmic noise or how to shuffle parallel implementations? *Int. J. Circuit Theory Appl.* **2020**, *48*, 674–695. [CrossRef]
11. The Common Criteria for Information Technology Security Evaluation. Available online: https://www.commoncriteriaportal.org (accessed on 1 September 2020)
12. Veyrat-Charvillon, N.; Standaert, F.X. Mutual information analysis: How, when and why? In *International Workshop on Cryptographic Hardware and Embedded Systems*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 429–443.
13. Ding, A.A.; Chen, C.; Eisenbarth, T. Simpler, faster, and more robust t-test based leakage detection. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 163–183.
14. Bronchain, O.; Hendrickx, J.M.; Massart, C.; Olshevsky, A.; Standaert, F. Leakage Certification Revisited: Bounding Model Errors in Side-Channel Security Evaluations. In *Annual International Cryptology Conference*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11692, pp. 713–737.
15. Cassiers, G.; Grégoire, B.; Levi, I.; Standaert, F.X. Hardware Private Circuits: From Trivial Composition to Full Verification. *IACR Cryptol. EPrint Arch.* **2020**, *2020*, 185.
16. Bilgin, B.; De Meyer, L.; Duval, S.; Levi, I.; Standaert, F.X. Low AND Depth and Efficient Inverses: A Guide on S-boxes for Low-latency Masking. *IACR Trans. Symmetric Cryptol. (ToSC)* **2020**, *2020*, 144–184. [CrossRef]

17. Mangard, S. Hardware Countermeasures against DPA? A Statistical Analysis of Their Effectiveness. In Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, 23–27 February 2004; Springer: Berlin/Heidelberg, Germany, 2004; Volume 2964, pp. 222–235.
18. Chari, S.; Rao, J.R.; Rohatgi, P. Template attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 13–28.
19. Fei, Y.; Ding, A.A.; Lao, J.; Zhang, L. A Statistics-based Fundamental Model for Side-channel Attack Analysis. *IACR Cryptol. EPrint Arch.* **2014**, *2014*, 152.
20. Moradi, A.; Standaert, F. Moments-Correlating DPA. In *Proceedings of the ACM Workshop on Theory of Implementation Security, TIS@CCS 2016, Vienna, Austria, October 2016*; Bilgin, B., Nikova, S., Rijmen, V., Eds.; ACM: New York, NY, USA, 2016; pp. 5–15.
21. Standaert, F.X.; Malkin, T.G.; Yung, M. A unified framework for the analysis of side-channel key recovery attacks. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, 26–30 April 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 443–461.
22. Duc, A.; Faust, S.; Standaert, F.X. Making masking security proofs concrete. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, 26–30 April 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 401–429.
23. Duc, A.; Faust, S.; Standaert, F. Making Masking Security Proofs Concrete (Or How to Evaluate the Security of Any Leaking Device), Extended Version. *J. Cryptol.* **2019**, *32*, 1263–1297. [CrossRef]
24. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
25. Abou-Moustafa, K.T.; Ferrie, F.P. A note on metric properties for some divergence measures: The Gaussian case. In Proceedings of the Asian Conference on Machine Learning, Singapore, 4–6 November 2012; pp. 1–15.
26. Lin, J. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* **1991**, *37*, 145–151. [CrossRef]
27. Bogdanov, A.; Knudsen, L.R.; Leander, G.; Paar, C.; Poschmann, A.; Robshaw, M.J.; Seurin, Y.; Vikkelsoe, C. PRESENT: An ultra-lightweight block cipher. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Vienna, Austria, 10–13 September 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 450–466.