Noname manuscript No.
(will be inserted by the editor)

Towards Easy Leakage Certification

Extended Version

François Durvaux, François-Xavier Standaert, Santos Merino Del Pozo

the date of receipt and acceptance should be inserted later

Abstract Side-channel attacks generally rely on the availability of good leakage models to extract sensitive information from cryptographic implementations. The recently introduced leakage certification tests aim to guarantee that this condition is fulfilled based on sound statistical arguments. They are important ingredients in the evaluation of leaking devices since they allow a good separation between engineering challenges (how to produce clean measurements) and cryptographic ones (how to exploit these measurements). In this paper, we propose an alternative leakage certification test that is significantly simpler to implement than the previous proposal from Eurocrypt 2014. This gain admittedly comes at the cost of a couple of heuristic (yet reasonable) assumptions on the leakage distribution. To confirm its relevance, we first show that it allows confirming previous results of leakage certification. We then put forward that it leads to additional and useful intuitions regarding the information losses caused by incorrect assumptions in leakage modeling.

Keywords Side-channel analysis, security evaluations.

1 Introduction

Side-channel attacks are an important threat against the security of modern embedded devices. As a result, the search for efficient approaches to secure cryptographic implementations against such attacks has been an ongoing process over the last 15 years. Sound tools for quantifying physical leakages are a central ingredient for this purpose, since they are necessary to balance the implementation cost of concrete countermeasures with the security improvements they provide. Hence, while early countermeasures came with proposals of security evaluations that were sometimes specialized to the countermeasure, more recent works have investigated the possibility to consider evaluation methods that generally apply to any countermeasure. The unified evaluation framework proposed at Eurocrypt 2009 is a popular attempt in this direction [26]. It suggests to analyze cryptographic implementations with a combination of information theoretic and security metrics. The first ones aim at measuring the (worst-case) information leakage independent of the adversary exploiting it, and are typically instantiated with the Mutual Information (MI). The second ones aim at quantifying how efficiently an adversary can take advantage of this leakage in order to turn it into (e.g.) a key recovery, and are typically instantiated with a success rate.

In this context, an important observation is that most side-channel attacks, and in particular any standard Differential Power Analysis (DPA) attack, require a leakage model [15]. This model usually corresponds to an estimation of the leakage Probability Density Function (PDF), possibly simplified to certain statistical moments. Since the exact distribution of (e.g.) power consumption or electromagnetic radiation measurements is generally unknown, it raises the problem that any physical security evaluation is possibly biased by model errors. In other words, security evaluations ideally require a perfect leakage model (so that all the information is extracted from the measurements). But in practice models are never perfect, so that the quality of the evaluation may highly depend on the quality of the evaluator. This intuition can be captured with the notion of Perceived Information (PI), that is nothing

Institute of Information and Communication Technologies, Electronics and Applied Mathematics / Crypto Group, Université catholique de Louvain, Louvain-la-Neuve, Belgium.

else than an estimation of the MI biased by the side-channel evaluator's model [21]. Namely, the MI captures the worst-case security level of an implementation, as it corresponds to an (hypothetical) adversary who can perfectly profile the leakage PDF. By contrast, the PI captures its practical counterpart, where actual (statistical) estimation procedures are used by an evaluator, in order to profile the leakage PDF.

Picking up on this problem, Durvaux et al. introduced first "leakage certification" methods at Eurocrypt 2014 [9]. Intuitively, leakage certification starts from the fact that actual leakage models are obtained via PDF estimation, which may lead to both estimation and assumption errors. As a result, and since it seems hard to enforce that such estimated models are perfect, the best that one can hope is to guarantee that they are "good enough". For estimation errors, this is easily verified using standard cross–validation techniques (in general, estimation errors can anyway be made arbitrarily small by measuring more). For assumption errors, things are more difficult since detecting them requires to find out whether the estimated model is close to an (unknown) perfect model. Interestingly, the Eurocrypt 2014 paper showed that indirect approaches allow determining if this condition is respected, essentially by comparing the model errors caused by incorrect assumptions to estimation errors. That is, let us assume that an evaluator is given a set of leakage measurements to quantify the security of a leaking implementation. As long as the assumption errors measured from these traces remain small in front of the estimation errors, the evaluator is sure that any improvement of his (possibly imperfect) assumptions will not lead to noticeable degradations of the estimated security level – since the impact of improved assumptions will essentially be hidden by the estimation errors. By contrast, once the assumption errors become significant in front of estimation ones, it means that an improved model is required to extract all the information from the measurements. Hence, leakage certification allows ensuring that the modeling part of an evaluation is sound (i.e. depends on the implementation – not the evaluator).

In practice, the leakage certification test in [9] requires a number of technical ingredients. Namely, the evaluator first has to characterize the leakages of the target implementation with a sampled (cumulative) distance distribution, and to characterize his model with a simulated (cumulative) distance distribution. Working with distances allows exploiting a univariate goodness–of–fit test even for leakages of large dimensionalities (i.e. it allows comparing the univariate distances between multivariate leakages rather than comparing the multivariate leakages directly). The Cramér–von–Mises divergence is used as a comparison tool in the Eurocrypt 2014 paper. Qualitatively, large divergences between the sampled and simulated distributions essentially mean that the assumptions are imperfect. Quantitatively, the evaluator then has to determine whether such divergences are significant, by verifying whether they can be explained by assumption errors. This requires computing the p-values when testing the hypothesis that the estimated model is correct (which again requires computing many simulated cumulative distance distributions). Summarizing, the beauty of this approach lies in the fact that it only relies on non-parametric estimations and requires no assumptions on the underlying leakage distributions. But this also comes at the cost of quite computationally intensive tools.

In this paper, we analyze solutions to mitigate the latter drawback, by investigating whether (computationally) cheaper and (conceptually) simpler certification procedures can be obtained at the cost of mild assumptions on the statistical distributions in hand. Two natural options directly come to mind for this purpose, that both aim to avoid dealing with the (expensive to characterize) cumulative leakage distributions directly. One possibility is to "summarize" the leakage distribution with its MI/PI estimates (since they can be used as indicators of the side-channel security level, as now proven in [7]). Another one is to analyze this distribution "moments by moments", motivated by the recent results in [18]. In both cases, and following the approach in [9], the main idea remains to compare actual leakage samples generated by a leaking implementation with hypothetical ones generated with the evaluator's model. Surprisingly, we show that the first approach cannot work, because of situations where model errors in one statistical moment (e.g. the mean) are reflected in another statistical moment (e.g. the variance), which typically arises when using the popular stochastic models in [22], and actually corresponds to the context of epistemic noise discussed in [13]. More interestingly, we also show that a moment-based approach provides excellent results under reasonable assumptions, and can borrow from the "leakage detection tests" that are already used by evaluation laboratories [11,16,8]. The resulting leakage certification method is significantly faster than the Eurocrypt 2014 one (and allows reproducing its experiments). We also show that it easily generalizes to masked implementations, and enables extracting very useful intuitions on the origin of the leakages. Eventually, our new tools lead to simple heuristics to approximate the information loss due to incorrect leakage models, which remained an open problem in [9].

Summarizing, we simplify leakage certification into a set of easy–to–implement procedures, hopefully more attractive for evaluation laboratories, of which we make the prototype implementations available as open source to facilitate their dissemination [1].

Cautionary note. This paper is about *leakage certification*, which is a different problem than the *leakage detection* one discussed in [11,16,8] (despite we indeed borrow some tools from leakage detection to simplify leakage certification). In this respect, Goodwill et al.'s non specific t-test is a natural approach to leakage detection, and allows determining if there is "some" leakage in an implementation, independent of whether it can be exploited (e.g. how many traces do you need to attack). By contrast, leakage certification aims to guarantee that a leakage model that can be exploited in an attack (and, e.g. can be used to determine a key recovery success rate) is close enough to the true leakage model. That is, it aims to make evaluators confident that their attacks are close enough to the worst-case ones. So leakage detection and certification are essentially complementary. Note that leakage models (and certification) are needed in any attempt to connect side-channel analysis with cryptographic security guarantees (e.g. in leakage resilience [10]), where we will always need an accurate evaluation of the security level, or to build security graphs such as introduced in [30].

Notations. Capital letters are for random variables, small caps for realizations, hats for estimations, sans serif fonts for functions and calligraphic ones for sets.

2 Background

2.1 Measurement setups

Our software experiments are based on measurements of an AES Furious implementation[1] run by an 8-bit Atmel AVR (ATMega644P) microcontroller at a 20 MHz clock frequency. We monitored the voltage variations across a 22 Ω resistor introduced in the supply circuit of our target chip. Acquisitions were performed using a Lecroy WaveRunner HRO 66 oscilloscope running at 625 Msamples/second and providing 8-bit samples. In practice, our evaluations focused on the leakage of the first AES master key byte (but would apply identically to any other enumerable target). Leakage traces were produced according to the following procedure. Let x and s be our target input plaintext byte and subkey, and $y = x \oplus s$. For each of the 256 values of y, we generated 1000 encryption traces, where the rest of the plaintext and key was random (i.e. we generated 256 000 traces in total, with plaintexts of the shape $p = x||r_1||\ldots||r_{15}$, keys of the shape $\kappa = s||r_{16}||\ldots||r_{30}$, and the r_i's denoting uniformly random bytes). In order to reduce the memory cost of our evaluations, we only stored the leakage corresponding to the 2 first AES rounds (as the dependencies in our target byte $y = x \oplus s$ typically vanish after the first round, because of the strong diffusion properties of the AES). We will denote the 1000 encryption traces obtained from a plaintext p including the target byte x under a key κ including the subkey s as: $\mathsf{AES}_{\kappa_s}(p_x) \rightsquigarrow l_y^i$ (with $i \in [1; 1000]$). Eventually, whenever accessing the points of these traces, we will use the notation $l_y^i(\tau)$ (with $\tau \in [1; 10\ 000]$, typically). Subscripts and superscripts are omitted when clear from the context.

Our hardware experiments are based on a similar setup, but consider a masked (threshold) implementation of PRESENT similar to the *Profile-4* design described in [19]. The leakage in such hardware implementations is mostly determined by the distance between two consecutive values in a target register R. Hence, we generated traces l_t^i (with $i \in [1; 100\ 000]$) for the 256 possible transitions $t =: R(x_1 \oplus s) \rightarrow R(x_2 \oplus s)$ between 4-bit intermediate results of the PRESENT S-box computations. This larger evaluation set was motivated by the protected nature and larger noise of this implementation. Because of similar memory constraints as in the software case, we limited our measurements to the first PRESENT round. These measurements were taken at a 500 Msamples/second, using the SAKURA-G board [2]. Our target device is a SPARTAN-6 FPGA.

2.2 PDF estimation methods

Side-channel attacks such as the standard DPA described in [15] require a leakage model. In general, such models correspond to estimations of the leakage PDF (possibly simplified to certain statistical moments). In the following, we will consider two important PDF estimation techniques for this purpose. For convenience, we describe them with a profiling based on intermediate values y's as considered in our software experiments, but these tools can be applied similarly to the transitions t's considered in our hardware experiments.

Gaussian templates. The Template Attack (TA) in [5] approximates the leakages using a set of normal distributions. It assumes that each intermediate computation generates Gaussian-distributed samples. In our typical scenario where the targets follow a key addition, we consequently use: $\hat{\Pr}_{\mathtt{model}}[l_y|s,x] \approx \hat{\Pr}_{\mathtt{model}}[l_y|s \oplus x] \sim \mathcal{N}(\mu_y, \sigma_y^2)$, where the "hat" notation is used to denote

[1] Available at `http://point-at-infinity.org/avraes/`.

the estimation of a statistic. This approach requires estimating the sample means and variances for each value $y = x \oplus s$ (and mean vectors / covariance matrices in case of multivariate attacks). We denote the construction of such a model with $\hat{\Pr}^{\texttt{ta}}_{\texttt{model}} \leftarrow \mathcal{L}^p_Y$, where $\mathcal{L}^p_Y$ is a set of N_p traces used for profiling.

Regression-based models. To reduce the data complexity of the profiling, an alternative approach proposed by Schindler et al. is to exploit Linear Regression (LR) [22]. In this case, a stochastic model $\hat{\theta}(y)$ is used to approximate the leakage function and built from a linear basis $\mathsf{g}(y) = \{\mathsf{g}_0(y), ..., \mathsf{g}_{B-1}(y)\}$ chosen by the adversary/evaluator (usually $\mathsf{g}_i(y)$ are monomials in the bits of y). Evaluating $\hat{\theta}(y)$ boils down to estimating the coefficients α_i such that the vector $\hat{\theta}(y) = \sum_i \alpha_i \mathsf{g}_i(y)$ is a least-square approximation of the measured leakages L_y. In general, an interesting feature of such models is that they allow trading profiling efforts for online attack complexity, by adapting the basis $\mathsf{g}(y)$. That is, a simpler model with fewer parameters will converge for smaller values of N_p, but a more complex model can potentially approximate the real leakage function more accurately. Compared to Gaussian templates, another feature of this approach is that only a single variance (or covariance matrix) is estimated for capturing the noise (i.e. it relies on an assumption of homoscedastic errors). Again, we denote the constructions of such a model with $\hat{\Pr}^{\texttt{lr}}_{\texttt{model}} \leftarrow \mathcal{L}^p_Y$.

2.3 Evaluation metrics

In this subsection, we recall a couple of useful evaluation metrics that have been introduced in previous works on side-channel attacks and countermeasures. For convenience, we again express these metrics for software (value-based) profiling. But they can straightforwardly adapted to the transition-based case.

Correlation coefficient. In view of the popularity of the Correlation Power Analysis (CPA) distinguisher in the literature [4], a natural candidate evaluation metric is Pearson's correlation coefficient. In the non-profiled setting, an a-priori (e.g. Hamming weight) model is used for computing the metric. The evaluator then estimates the correlation between his measured leakages and the modeled leakages of a target intermediate value. In our AES example, it leads to $\hat{\rho}(L_Y(\tau), \mathsf{model}_{\mathsf{cpa}}(Y))$. In practice, this estimation is performed by sampling (i.e. measuring) N_t test traces from the leakage distribution (we denote the set of these N_t test traces as $\mathcal{L}^t_Y$). Next, and in order to avoid possible biases due to an incorrect a-priori choice of leakage model, a natural solution is to extend the previous proposal to the profiled setting. In this case, the evaluator will start by estimating a model from N_p profiling traces: $\hat{\mathsf{model}}_{\mathsf{cpa}} \leftarrow \mathcal{L}^p_Y$ (with $\mathcal{L}^p_Y \perp\!\!\!\perp \mathcal{L}^t_Y$). In practice, $\hat{\mathsf{model}}_{\mathsf{cpa}}$ can be seen as a simplification of the previous Gaussian templates, that only includes estimates for the first-order moments of the leakages. That is, for any time sample τ, we have $\hat{\mathsf{model}}_{\mathsf{cpa}}(y) = \hat{m}^1_y(\tau) = \hat{\mathsf{E}}_i(L^i_y(\tau))$, with $\hat{m}^1_y$ a first-order moment and $\hat{\mathsf{E}}$ the sample mean operator.

Mutual and perceived information. In theory, the worst-case security level of an implementation can be measured with a MI metric. Taking advantage of the notations in Section 2.1 and considering the standard case where a key byte S is targeted, it amounts to estimate the following quantity:

$$\mathrm{MI}(S;\,X,L) = \mathrm{H}[S] + \sum_{s\in\mathcal{S}} \Pr[s] \sum_{x\in\mathcal{X}} \Pr[x] \cdot \sum_{l^i_y\in\mathcal{L}_t} \Pr_{\texttt{chip}}[l^i_y|s,x].\log_2 \Pr_{\texttt{chip}}[s|x,l^i_y]. \tag{1}$$

When summing over all s and x values, and a sufficiently large number of leakages, the estimation tends to the correct MI. Yet, as mentioned in introduction, the chip distribution $\Pr_{\texttt{chip}}[l^i_y|s,x]$ is generally unknown to the evaluator. So in practice, the best that we can hope is to compute the following PI:

$$\hat{\mathrm{PI}}(S;\,X,L) = \mathrm{H}[S] + \sum_{s\in\mathcal{S}} \Pr[s] \sum_{x\in\mathcal{X}} \Pr[x] \cdot \sum_{l^i_y\in\mathcal{L}_t} \Pr_{\texttt{chip}}[l^i_y|s,x].\log_2 \hat{\Pr}_{\texttt{model}}[s|x,l^i_y], \tag{2}$$

where $\hat{\Pr}_{\texttt{model}} \leftarrow \mathcal{L}^p_Y$ is typically obtained using the previous Gaussian templates or LR-based models. Under the assumption that the model is properly estimated, it is shown in [15] that the CPA and PI metrics are essentially equivalent in the context of standard univariate side-channel attacks (i.e. exploiting a single leakage point $l^i_y(\tau)$ at a time). By contrast, only the PI naturally extends to multivariate attacks. It can be interpreted as the amount of information leakage that will be exploited by an adversary using an estimated model. So just as the MI is a good predictor for the success rate of an ideal TA exploiting the perfect model $\Pr_{\texttt{chip}}$, the PI is a good predictor for the success rate of an actual TA exploiting the "best available" model $\hat{\Pr}_{\texttt{model}}$ obtained thanks to profiling.

Moments-correlating DPA. Eventually, and in order to extend the CPA distinguisher to higher-order moments, the Moments-Correlating Profiled DPA (MCP-DPA) has been introduced in [18]. It features essentially the same steps as a profiled CPA. The only difference is that the adversary first estimates dth-order statistical moments with his profiling traces, and then uses $\hat{\mathsf{model}}^{d}_{\texttt{mcp-dpa}}(y) = \hat{m}^d_y(\tau)$, with $\hat{m}^d_y$ a dth-order moment. For concreteness, we will consider d's up to four (i.e. the sample mean for $d = 1$, variance for $d = 2$, skewness for $d = 3$ and kurtosis for $d = 4$), which allows us discussing the relevant case-study of a masked implementation with two shares. Yet, the tool naturally extends to any d. One useful feature of this distinguisher is that it embeds the same "metric" intuition as CPA: the higher the correlation estimated with MCP-DPA, the more efficient the corresponding attack exploiting a moment of given order.

2.4 Estimating a metric with cross–validation

Estimating a metric α from a leaking implementation holds in two steps. First, a model has to be estimated from a set of profiling traces $\mathcal{L}_p$: $\hat{\texttt{model}} \leftarrow \mathcal{L}_p$. Second, a set of test traces $\mathcal{L}_t$ (following the true distribution $\Pr_{\texttt{chip}}$) is used to estimate the metric: $\hat{\alpha} \leftarrow (\mathcal{L}_t, \hat{\texttt{model}})$. As a result, two main types of errors can arise. First, the number of traces in the profiling set may be too low to estimate the model accurately (which corresponds to estimation errors). Second, the model may not be able to accurately predict the distribution of samples in the test set, even after intensive profiling (which then corresponds to assumption errors).

In order to verify that estimations in a security evaluation are sufficiently accurate, the solution used in [9] is to exploit cross–validation. In general, this technique allows gauging how well a predictive (here leakage) model performs in practice. For k-fold cross–validations, the set of evaluation traces $\mathcal{L}$ is first split into k (non overlapping) sets $\mathcal{L}^{(i)}$ of approximately the same size. Let us define the profiling sets $\mathcal{L}_p^{(j)} = \bigcup_{i\neq j} \mathcal{L}^{(i)}$ and the test sets $\mathcal{L}_t^{(j)} = \mathcal{L} \setminus \mathcal{L}_p^{(j)}$. The sample metric is then repeatedly computed k times for $1 \leq j \leq k$ as follows. First, we build a model from a profiling set: $\hat{\texttt{model}}^{(j)} \leftarrow \mathcal{L}_p^{(j)}$. Then we estimate the metric with the associated test set $\hat{\alpha}^{(j)} \leftarrow (\mathcal{L}_t^{(j)}, \hat{\texttt{model}}^{(j)})$. Cross–validation protects evaluators from obtaining too optimistic sample metric values due to over-fitting, since the test computations are always performed with an independent data set. Finally, the k outputs can be averaged in order to get an unbiased metric estimate, and their spread characterizes the result's accuracy.

3 A motivating negative result

As mentioned in introduction, detecting assumption errors is generally more challenging than detecting estimation errors (which is easily done with the previous cross–validation). Intuitively, it requires to investigate the likelihood that samples obtained from a leaking device can indeed be explained by an estimated model, which requires a (multivariate) goodness–of–fit test. Since such tests are computationally intensive, an appealing alternative would be to check whether the samples obtained from the leaking device lead to a PI that is at least close enough to the MI: this would guarantee a good estimation of the security level. But we again face the problem that the MI is unknown, which imposes trying indirect approaches. That is, we would need an metric counterpart to the sampled simulated distance distribution in [9], which would typically correspond to the following definition of Hypothetical (mutual) Information (HI):

$$\hat{\mathrm{HI}}(S;\, X, L) = \mathrm{H}[S] + \sum_{s\in\mathcal{S}} \Pr[s] \sum_{x\in\mathcal{X}} \Pr[x] \cdot \sum_{l_y^i\in\mathcal{L}_t} \hat{\Pr}_{\texttt{model}}[l_y^i|s,x].\log_2 \hat{\Pr}_{\texttt{model}}[s|x,l_y^i]. \qquad (3)$$

Intuitively, this HI corresponds to the amount of information that would be extracted from an hypothetical implementation that would exactly leak according to the model $\hat{\Pr}_{\texttt{model}}$. In itself, the HI is useless to the evaluator, as it is actually disconnected from the chip distribution. For example, even a totally incorrect model (i.e. leading to a negative PI) would lead to a positive HI. By contrast, we could hope that as long as the HI and PI are "close", the assumption errors are "small enough" for the number of measurements considered in the security evaluation. Furthermore, we could use a simple hypothesis test to detect non-closeness. For a number of traces N in the evaluation set, this would require to compute estimates $\hat{\mathrm{PI}}(S;X,L)^{(j)}$ and $\hat{\mathrm{HI}}(S;X,L)^{(j)}$ with cross–validation, and to check whether these estimates come from different (univariate) distributions. If they significantly differ, we would conclude that the model exhibits assumption errors that degrade the estimated security level, in a similar fashion as in [9].

Unfortunately, and despite it can detect certain assumption errors, this approach cannot succeed in general. A simple counter–example can be explained in the context of LR. Say an adversary estimates a model with a linear basis, which leads to significant differences between the actual (mean) leakages and the ones suggested by the model. Then, because of the homoscedastic error assumption, the single variance of the LR-based model will reflect this error (i.e. capture both physical noise and model error). As a result, whenever this type of error increases, the PI will decrease (as expected) but the HI will also decrease (contrary to the MI). So testing the consistency between the PI and HI estimates will not reveal the inconsistencies between the PI estimates and the true MI.

4 A new method to detect assumption errors

Despite negative, the previous counter–example suggests two interesting tracks for simplifying leakage certification tests. First, summarizing a complete distribution into representative metrics (e.g. such as the PI) allows taking advantage of simpler statistical tests. Second, since the fact that the homoscedastic errors assumption is not fulfilled implies that errors made in the estimation of certain statistical moments (or more generally, parameters) of a distribution are reflected in other statistical moments of this distribution, a natural approach is to test the relevance of a model "moment by moment". That is, for a number of traces N in an evaluation set, one could verify that the moments estimated from actual leakage samples are hard to tell apart from the moments estimated from the model (with the same number of samples N). Based on this idea, our simplified method to detect assumption errors will be based on the following two hypotheses (one strictly necessary and the other optional but simplifying).

1. *The leakage distribution is well represented by its statistical moments.* This corresponds to the classical "moment problem" in statistics, for which there exist counter-examples (e.g. the log-normal distribution is not uniquely characterized by its moments). So our (informal) assumption is that these counter-examples will not be significant for our experiments.
2. *The sampled estimates of our statistical moments are approximately Gaussian-distributed.* This directly derives from the central limit theorem and actually depends on the number of samples used in the estimations (which will become sufficient as the leakages become more noisy, e.g. in the case of protected implementations that are most relevant in practice).

Let us add a few words of motivation for those assumptions. First recall that we know from the previous results in [9] that leakage certification is possible without such assumptions, at the cost of somewhat involved statistical reasoning and estimations. So it seems natural to investigate alternative (heuristic) paths allowing to reach similar conclusions. As will be shown next, this is indeed the case of our simplified approach for a couple of relevant scenarios. Second, statistical moments are at the core of the reasoning regarding the masking countermeasure. That is, the security order of an implementation is generally defined as the lowest informative moment in the leakage distribution (minus one) – see [7] for an extensive discussion of this issue. Besides, many concrete (profiled and non-profiled) side-channel attacks are based (implicitly or explicitly) on parametric PDF estimation techniques that rely on the estimation of moments (e.g. the Gaussian templates and LR-based models in Section 2.2, but also second-order attacks such as [6, 20]). So a certificative approach based on an analysis of moments seems well founded in these cases.[2] Eventually, contradictions of this first hypothesis imply potential false negatives in leakage certification, but no false positives. So it remains that any detected assumption error requires model improvement by the evaluator. Overall, and maybe most importantly, we believe that the following tools open interesting research avenues regarding the intuitive evaluation of leaking devices based on their moments.

As for the Gaussian assumption, our motivation is even more pragmatic, and relates to the observation that simple t-tests are becoming de facto standards in the preliminary evaluation of leaking devices [11, 16, 23]. So we find it appealing to rely on statistical tools that are already widespread in the CHES community, and to connect them with leakage certification. As will be clear next, this allows us to use the same evaluation method for statistical moments of different orders. However, we insist that it is perfectly feasible to refine our approach by using a well adapted test for each statistical moment (e.g. F-test for variances, . . .).

4.1 Test specification

The main idea behind our new leakage certification method is to compare (actual) dth-order moments $\hat{m}_y^d$ estimated from the leakages with (simulated) dth-order moments $\tilde{m}_y^d$ estimated from the evaluator's model $\hat{\Pr}_{\texttt{model}}$ (by sampling this model). Thanks to our second assumption, this comparison can simply be performed based on Student's t-test. For this purpose, we need multiple estimations of the moments $\hat{m}_y^d$ and $\tilde{m}_y^d$, that

[2] Note that theoretical approches to guarantee that a distribution is well characterized by its moments (such as Carleman's condition [25]) typically apply when considering an infinite number of them and in general, no distribution is determined by a finite number of moments. So the restriction of our reasoning to specific classes of meaningful distributions is in fact necessary for our approach to be sound. Besides, note also that non-parametric PDF estimations may not suffer from assumption errors (at the cost of a significantly increased estimation cost), so are out of scope here.

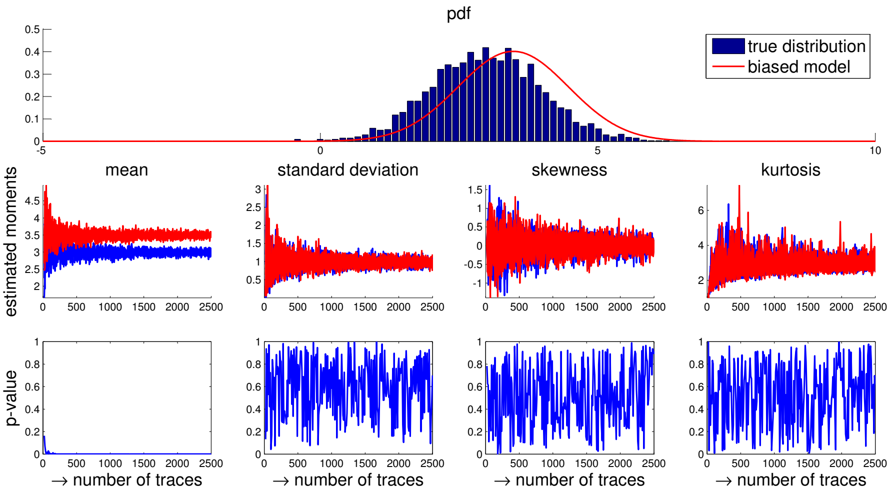

Fig. 1: Gaussian leakages, Gaussian model, error in the estimated mean.

we obtain thanks to an approach inspired from Section 2.4 (although there is no cross–validation involved here). More precisely, we start by splitting the full set of evaluation traces $\mathcal{L}$ into k (non overlapping) sets of approximately the same size $\mathcal{L}^{(j)}$, with $1 \leq j \leq k$. From these k subsets, we produce k estimates of (actual) d^{th}-order moments $\hat{m}_y^{d,(j)}$, each of them from a set $\mathcal{L}^{(j)}$. We then produce a set of simulated traces $\tilde{\mathcal{L}}$ that has the same size and corresponds to the same intermediate values as the real evaluation set $\mathcal{L}$, but where the leakages are sampled according to the model that we want to evaluate. In other words, we first build the model $\hat{\Pr}_{\texttt{model}} \leftarrow \mathcal{L}$, and then generate a simulated set of traces $\tilde{\mathcal{L}} \leftarrow \hat{\Pr}_{\texttt{model}}$. Based on $\tilde{\mathcal{L}}$, we produce k estimates of (simulated) d^{th}-order moments $\tilde{m}_y^{d,(j)}$, each of them from a set $\tilde{\mathcal{L}}^{(j)}$, as done for the real set of evaluation traces. From these real and simulated moments estimates, we compute the following quantities:

$$\hat{\mu}_y^d = \hat{\mathsf{E}}_j(\hat{m}_y^{d,(j)}), \qquad \hat{\sigma}_y^d = \sqrt{\hat{\mathsf{var}}_j(\hat{m}_y^{d,(j)})},$$

$$\tilde{\mu}_y^d = \hat{\mathsf{E}}_j(\tilde{m}_y^{d,(j)}), \qquad \tilde{\sigma}_y^d = \sqrt{\hat{\mathsf{var}}_j(\tilde{m}_y^{d,(j)})},$$

where $\hat{\mathsf{var}}$ is the sample variance operator. Eventually, we simply estimate the t statistic (next denoted with Δ_y^d) as follows:

$$\Delta_y^d = \frac{\hat{\mu}_y^d - \tilde{\mu}_y^d}{\sqrt{\frac{(\hat{\sigma}_y^d)^2 + (\tilde{\sigma}_y^d)^2}{k}}}.$$

The p-value of this t statistic within the associated Student's distribution returns the probability that the observed difference is the result of estimations issues:

$$p = 2 \times (1 - \mathsf{CDF}_{\mathsf{t}}(|\Delta_y^d|, d_f)),$$

where $\mathsf{CDF}_{\mathsf{t}}$ is the Student's t cumulative distribution function, and d_f is its number of freedom degrees.[3] In other words, a small p-value indicates that the model is incorrect with high probability. Concretely, the only parameter to set in this test is the number of non overlapping sets k. Following [9], we used $k = 10$ which is a rather standard value in the literature. Note that increasing k has very limited impact on the accuracy of our conclusions since all variance estimates in the t-test are normalized by k. By contast it increases the time complexity of the test (so keeping k reasonably small is in general a good strategy).

5 Simulated experiments

In order to validate our moment-based certification, we first analyze a couple of simulated experiments, where we can control the assumption errors. In particular, and in order to keep these simulations reasonably close to concrete attacks, we consider four distinct scenarios. In the first one (reported in Figure 1) we investigate errors in the mean of the model distribution. The upper part of the figure represents a non-parametric estimate of the true leakage distribution (with histograms) and a leakage model $\hat{\Pr}_{\texttt{model}}$ following a Gaussian distribution. The middle part of the figure represents the estimated moments $\hat{m}_y^{d,(j)}$ (in blue) and $\tilde{m}_y^{d,(j)}$ (in red), in function of the number of traces used for their estimation, from which we clearly see the error in the

[3] Student's t distribution is a parametric probability density function whose only parameter is its number of freedom degrees, that can be directly derived from k and the previous σ estimates as: $d_f = (k-1) \times [(\hat{\sigma}_y^d)^2 + (\tilde{\sigma}_y^d)^2]^2 / [(\hat{\sigma}_y^d)^4 + (\tilde{\sigma}_y^d)^4]$.

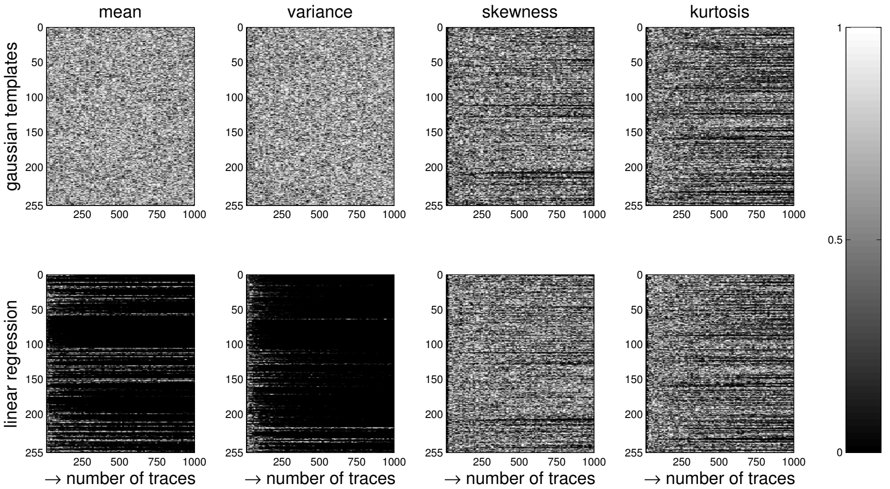

Fig. 2: Results of the new leakage certification test for software measurements.

mean. The lower part of the figure represents the evolution of our test's p-value in function of the number of traces used for certification. As expected, we directly detect an error in the mean (reflected by a very small p-value for this moment), whereas the p-values of the other moments remain erratic, reflecting the fact that (hypothetical) assumption errors are not significant in front of estimation errors (i.e. do not lead to significant information losses) for those moments. Similar figures corresponding to model errors in the variance, skewness and kurtosis are reported in Appendix A, Figures 9, 10 and 11. The last two cases typically correspond to the setting of a masked implementation for which the true distribution is a mixture [28].

These results confirm the simplicity of the method. As the number of measurements in the evaluation set increases, we are able to detect the assumption errors in all cases. The only difference between the applications to different moments is that errors on higher-order moments may be more difficult to detect as the noise increases. This difference is caused by the same argument that justifies the relevance of the higher-order masking countermeasure. Namely, the sampling complexity when estimating the moments of a sufficiently noisy distribution increases exponentially in d. However, this is not a limitation of the certification test: if such errors are not detected for a given evaluation set, it just means that their impact is still small in front of assumption errors at this stage of the evaluation. Besides, we note that the respective relevance of the model errors on different moments will be further discussed in Section 7.

6 Software experiments

In order to obtain a fair comparison with the results provided in [9], we first applied our new leakage certification method to the same case-study. Namely, we used the measurement setup from Section 2.1 and evaluated the relevance of two important profiling methods, namely the Gaussian TA and LR, for the most informative time sample in our leakage traces (i.e. with maximum PI).

The main difference with the previous simulated experiments is that we now have to test 256 models independently (each of them corresponding to a target intermediate value $y = x \oplus s$). Our results are represented in Figure 2, where we plot the p-values output by our different t-tests in greyscale, for four statistical moments (i.e. the mean, variance, skewness and kurtosis). That is, each line in this plot corresponds to the lower part of the previous figures (1, 9, 10, 11). A look at the first two moments essentially confirms the conclusions of Durvaux et al. More precisely, the Gaussian templates capture the measured leakages quite accurately (for the 256,000 traces in our evaluation set). By contrast, the linear regression quickly exhibits inconsistences. Interestingly, assumption errors appear both in the means and in the variances, which corresponds to the expected intuition. That is, errors in the means are detected because for most target intermediate values, the actual leakage cannot be accurately predicted by a linear combination of the S-box output bits.[4] And errors in the variances appear because the LR-based

[4] This happens for the selected time sample because of pipelining effects in the AVR microcontroller. Note that as

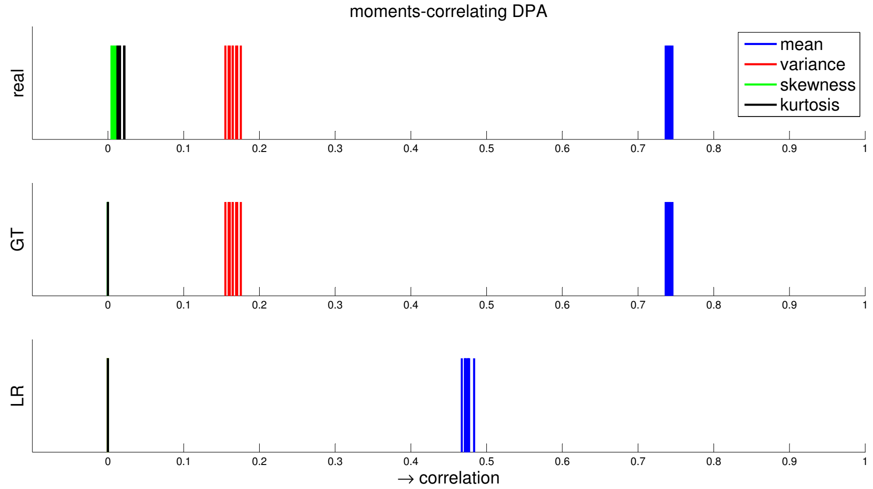

Fig. 3: MCP-DPA results for software measurements (with 256 × 1000 traces).

models rely on the homoscedastic error assumption and capture both physical noise and noise due to assumption errors in a single term.

By contrast, and quite intriguingly, a look at the last two moments (i.e. skewness and kurtosis) also shows some differences with the results in [9]. That is, we remark that even for Gaussian templates, small model errors appear in these higher-order moments. This essentially corresponds to the fact that our measured leakages do not have perfectly key-independent skewness and kurtosis, as we assume in Gaussian PDF estimations. This last observation naturally raises the question whether these errors are significant, i.e. do they contradict the results of the Eurocrypt 2014 leakage certification test? In the next section, we show that it is not the case, and re-conciliate both approaches by investigating the respective informativeness of the four moments in our new test.

7 Quantifying the information loss

Since Figure 2 suggests the existence of (small) model errors in our Gaussian templates, that are due to an incorrect characterization of the third- and fourth-order moments in our leakage traces, we now want to investigate whether these errors are leading to significant information losses. Fortunately, our "per-moment" approach to leakage certification also allows simple investigations in this direction (which heuristically answers one of the open questions in [9], about the information loss due to model errors). In particular, we can simply use the MCP-DPA mentioned in Section 2.3 for this purpose. Roughly, this tool computes the correlation between a simplified model (that corresponds to dth-order moments of the leakage distribution) to samples raised to the power d (centered or standardized if we consider centered and standardized moments). As discussed in [18], the resulting estimated correlation features a "metric intuition": the higher the value of the MCP-DPA distinguisher computed for an order d, the more efficient the MCP-DPA attack exploiting this statistical order of the leakage distribution. Hence, computing the value of the MCP-DPA distinguisher for different values of d should solve our problem, i.e. determine whether the moments for which assumption errors are detected are (among) the most informative ones.

Concretely, we start by applying MCP-DPA in the traditional sense and exploit cross–validation for this purpose, this time following exactly Section 2.4. That is, the set of evaluation traces $\mathcal{L}$ is again split into k (non overlapping) sets $\mathcal{L}^{(i)}$ of approximately the same size, and we use profiling sets $\mathcal{L}_p^{(j)} = \bigcup_{i \neq j} \mathcal{L}^{(i)}$ and test sets $\mathcal{L}_t^{(j)} = \mathcal{L} \setminus \mathcal{L}_p^{(j)}$. We then repeatedly compute the dth-order moments $\hat{m}_y^{d,(j)} \leftarrow \mathcal{L}_p^{(j)}$, and the dth-order MCP-DPA distinguisher:

$$\text{MCP-DPA}^{(j)}(d) = \hat{\rho}\Big(\hat{M}_Y^{d,(j)}, (L_y)^d \leftarrow \mathcal{L}_t^{(j)}\Big).$$

As previously mentioned, it corresponds to the sample correlation between the random variable representing the estimated moments $\hat{M}_Y^d$, and the random variable corresponding to the leakage samples coming from the test set $L_y \leftarrow \mathcal{L}_t^{(j)}$, raised to power d (possibly centered

in [9], the linear model did not exhibit any assumption error for other time samples given the amount of measured traces.

or standardized if we consider centered and standardized moments). The $k = 10$ estimates for this MCP-DPA metric are represented in the top part of Figure 3. We additionally considered two slightly tweaked versions of MCP-DPA, where we rather estimate Gaussian TA (resp. LR-based) models $\hat{\Pr}^{\mathtt{ta}}_{\mathtt{model}}$ (resp. $\hat{\Pr}^{\mathtt{lr}}_{\mathtt{model}}$), and consider the two (resp. one) key-dependent moments from these models to compute the metric. These tweaked MCP-DPAs are represented in the middle (resp. lower) part of the figure. Our main observations are as follows. First, the upper part of the figure suggests that the most informative moments in our leakage traces are the mean and variance. There is indeed a small amount of information in the skewness and kurtosis. But by considering the classical rule–of–thumb that the number of samples N_s required to perform a successful correlation-based attack is inversely proportional to the square of its correlation coefficient, that is:

$$N_s \approx \frac{c}{\hat{\rho}\left(\hat{M}_Y^d, (L_y)^d\right)^2},$$

with c a small constant, we can see that the additional information gain in these higher-order moments is very limited in our context. For example the value of the mean-based MCP-DPA distinguisher (for which no assumption errors are detected) is worth ≈ 0.74 in the figure, and the value of the kurtosis-based MCP-DPA distinguisher (for which assumption errors are detected) is worth ≈ 0.02. Considering these two moments as independent information channels, the loss caused by the assumption errors on the kurtosis can be approximated as $\frac{0.74^2}{0.74^2+0.02^2} \approx 0.999$, meaning that improving the model so that the kurtosis is well characterized could only (and ideally) lead to an attack requiring this fraction of N_s to succeed (that is close to 1). This observation backs up the conclusions of the generic leakage certification test in [9] that Gaussian templates are sufficiently accurate for our evaluation set. Next, we see that TA-based and LR-based MCP-DPA yield no information in the higher-order moments, which trivially derives from the fact that they rely on a Gaussian assumption. Eventually, and quite interestingly, we note that the information loss between LR-based models and TA-based models can be approximated thanks to the correlation between their moments. For example, and considering the means in Figure 3, we can compute the value of the LR-based MCP-DPA distinguisher – worth ≈ 0.48 in the figure – by multiplying the value of the TA-based MCP-DPA distinguisher – worth ≈ 0.74 – by $\hat{\rho}(\hat{M}_Y^{d,\mathsf{ta}}, \hat{M}_Y^{d,\mathsf{lr}})$ – worth ≈ 0.65 in our experiments (i.e. by taking advantage of the “product rule” for the correlation coefficient in [27]).

Those last tools are admittedly informal. In particular, it is important to insist that the use of the correlation coefficient as an information theoretic metric only holds for relatively noisy distributions (i.e. low correlation coefficient values) [15]. Yet, we believe they provide a useful variety of heuristics allowing evaluators to analyze the results of their certification tests. In particular, they lead to easy–to–exploit intuitions regarding the impact of model errors detected in moments of a given order. As discussed in the beginning of Section 4, further formalizing these findings, and possibly putting forward relevant scenarios where our simplified approach leads to significant shortcomings, is an interesting scope for further research. Meanwhile, the next section describes an open source code to demonstrate the implementation efficiency of our new certification tests, and Section 9 complements these findings by showing that the proposed certification method applies too in the more challenging context of (unprotected and) masked hardware implementations.

8 Open source code

The previous experiments can be carried out thanks to five scripts and function files (in the Matlab format **.m**) available from [1] and described next:

1. **main.m**. This top-level script loads the leakage samples, changes their format, and calls the certification and display functions. The samples need to be formatted because the code is vectorised for efficiency: samples usually come as a disordered vector, i.e. regardless of the target intermediate values. We reshape this sample vector into a matrix of which each column corresponds to an intermediate value. That is, a N_t-by-256 matrix is created if 8-bit values are investigated (with N_t samples per target value).
2. **moment_based_analysis.m**. This function detects assumption errors with our new certification test. For a given number of samples, the first four moments are computed from the leakage samples and then compared to the moments simulated from the two considered models, i.e. Gaussain templates and LR-based. In order to avoid overfitting, we use cross-validation each iteration. This function produces the results reported in Figure 2.
3. **plot_grey_graphs.m**. This script displays the p-values as in Figure 2.
4. **mcpdpa.m**. This function performs the MCP-DPA part of our analysis, which is only computed for the maximum number of samples per target intermediate values (i.e. N_t). Cross-validation is again ex-

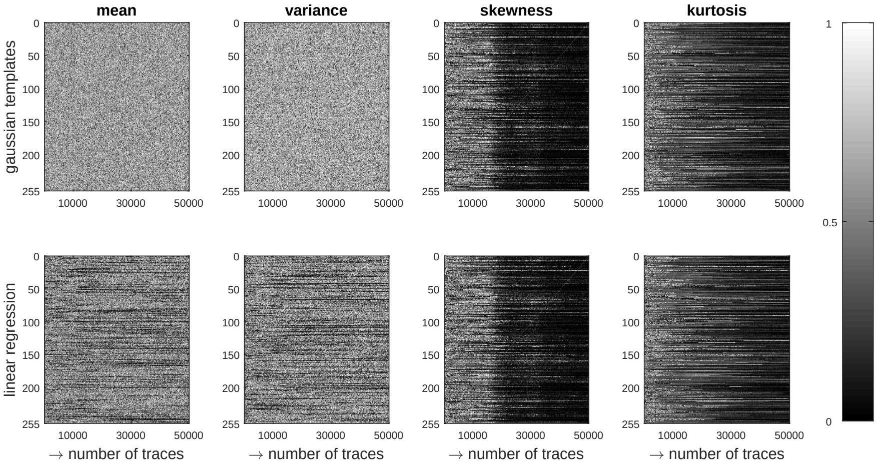

Fig. 4: Results of the new leakage certification test for masked hardware.

ploited in order to avoid overfitting. This function produces the results reported in Figure 3.

5. **plot_mcpdpa.m**. This last script displays the MCP-DPA analysis of Figure 3.

Two files in the Matlab data format **.mat** are included in the demonstration code. The first file, **aes_sbox.mat**, is a table corresponding to the AES S-box execution, and is solely used to build the linear regression model. The second file, **traces.mat**, is a file containing the leakage samples in a vector **traces**, and the associated target intermediate value $y = x \oplus k$ in a vector $\mathbf{y}$.

From the time complexity point–of–view, this code is considerably more efficient than the previous solution from [9]. Strict comparisons are hard to obtain since our current implementations are prototype ones, and further optimizations could be investigated. But roughly speaking, generating leakage certification plots for 256 leakage models as in Figure 2 is completed in seconds of computations on a standard desktop computer, whereas it typically took hours with the Eurocrypt 2014 tools. Since the cost of our heuristic leakage certification method is essentially similar to the one of a CPA, it can easily be applied on full leakage traces, in particular if some high performance computing can be exploited to take advantage of the parallel nature of the certification problem [17].

9 Hardware experiments

As usual in the evaluation of masked implementations, we first ran a preliminary test by setting the masks to constant null values, which actually corresponds to the case of an unprotected FPGA implementation of PRESENT. As mentioned in Section 2, the main difference between this hardware case study and the previous software one is that the leakages now depend on transitions between consecutive values in a target register. For the rest, the details about such attacks and their relation with the underlying architecture (that can be found in [18,19]) are not necessary to understand our following discussions. As expected, the results of this preliminary test were essentially similar to the ones of the unprotected software case. That is, we did not detect assumption errors for the Gaussian templates with up to 256,000 measurements, while some errors could be detected in the LR-based attacks. The only interesting bit of information from this context is the lower MCP-DPA values observed in Appendix A, Figure 12, which can be associated to a higher noise level.

We next moved to the more meaningful case with random masks activated, for which the leakage certification results are given in Figure 4. Two main observations can be extracted from these plots. First, and as previously, LR-based attacks exhibit model errors in the first two moments, that are not detected with Gaussian templates. Second, and more importantly, we see that strong errors are detected for the skewness and kurtosis, already quite early in our evaluation set. This is expected since these two moments are not captured at all, neither by our Gaussian templates, nor by LR-based attacks. However, since the information in a (first-order) threshold implementation should lie in higher-order (at least > 1) statistical moments, it naturally raises the question whether this model imperfection is critical from a security evaluation point–of–view.

In order to answer this question, we again performed MCP-DPA attacks for different statistical orders, as represented in Figure 5. Interestingly, the upper plot shows that, while there is no information in the first-order moments (as guaranteed by the first-order secu-

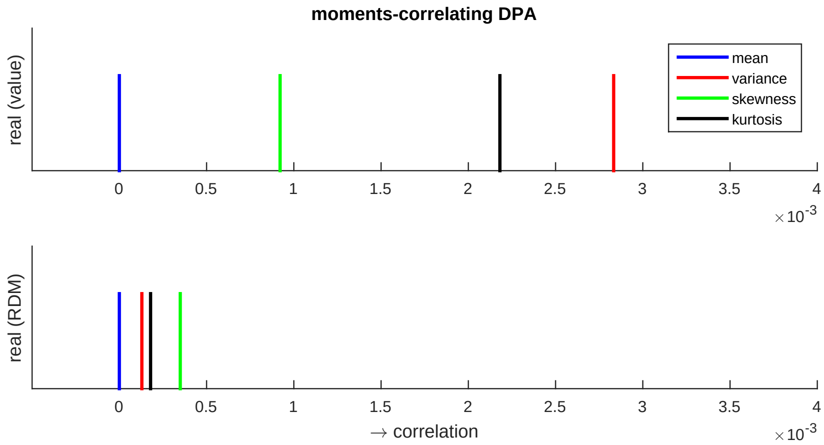

Fig. 5: MCP-DPA results for masked hardware (with 256 × 50,000 traces).

rity property of threshold implementations), there is indeed information in all the other moments. So we are actually in a case where the leakage certification test suggests improvements, and tells the evaluator that his (Gaussian) templates are not sufficient to extract all the information, while LR-based attacks could not succeed at all (since they do estimate a single variance for all the profiled transitions). This raises interesting scopes for further research, since profiling methods that easily incorporate such higher-moments have not been much explored in the side-channel literature so far [3, 24].

Besides, another useful observation arises if, rather than simply plotting the asymptotic MCP-DPA values, we also plot the Relative Distinguishing Margin (RDM), defined in [31] as the distance between the correct key distinguisher value and the value for the highest ranked alternative. As illustrated by the lower plot of Figure 5, this RDM is larger for the skewness than for the variance. This means that while the variance is the most informative moment overall (i.e. assuming some enumeration is possible as a post-processing after the attack [29]), the skewness is more useful in case the adversary has to recover the key thanks to side-channel measurements exclusively (since the nearest rival captured by the RDM is usually the most difficult to distinguish from the good key).

Summarizing, these experiments confirm the applicability of our easy leakage certification tests in the practically-relevant case study of a threshold implementation (that is representative of state–of–the–art masking schemes). They also put forward that combining MCP-DPA evaluations with the estimation of a RDM metric allows extracting additional intuitions regarding the information vs. computation tradeoff that is inherent to any side-channel attack.

10 More simulations

The previous results put forward that our certification test is handy to detect assumption errors in unprotected and masked implementations. Yet, we were also considering simple modeling tools (i.e. Gaussian templates and linear regression) which essentially work by estimating statistical moments. Before to conclude this work, we finally wanted to investigate what happens in case more complex models are considered.

For this purpose, we again investigated masked implementation, but this time exploiting the Gaussian mixture modeling described in [28]. Namely we considered a simulated setting where a secret s is split into 2 or 3 shares (i.e. $s = s_0 \oplus s_1$ or $s = s_0 \oplus s_1 \oplus s_2$). During profiling, the adversary gets a noisy sum of (Hamming weight) leakages for all the shares *and* the shares values (i.e. the randomness s_0, s_1, s_2), so that he can estimate the exact model distribution as:

$$\hat{\Pr}_{\texttt{model}}[l_s|s] = \sum_{s_0,s_1,s_2 \in \mathcal{S}} \hat{\Pr}_{\texttt{model}}[l_s|s, s_0, s_1, s_2].$$

Then, during the attack, he only has access to the noisy sum of leakages (which emulates the parallel implementation of the previous section). One interesting feature of this setting is that there are less secrets (namely 2^n) than there are Gaussian templates to estimate when building the mixture (namely 2^{2n} or 2^{3n} for the 2-share and 3-share examples). So we can expect that estimating the Gaussian mixture model will be more complex than just estimating moments (as with the simpler Gaussian templates).[5]

[5] We considered simulated measurements for two main reasons. First, and as in Section 5, it allows us to control, and therefore to accurately understand, the observed leakages (e.g. we are sure that the Gaussian mixture modeling

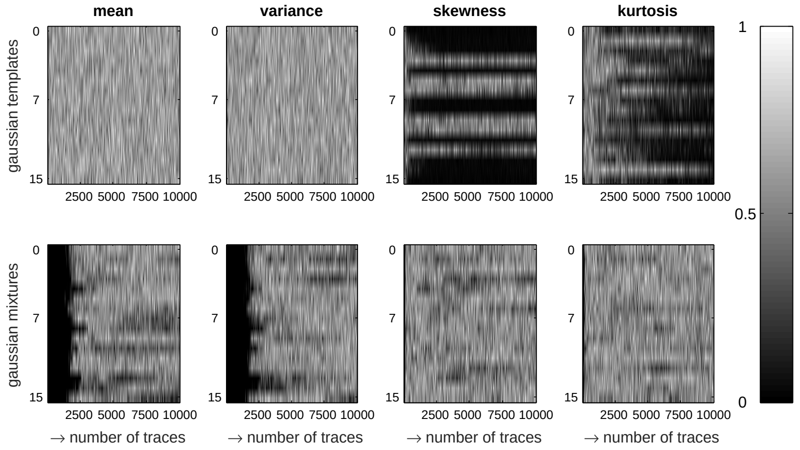

Fig. 6: Results of the new leakage certification test with masked simulations (3 shares).

The results of our new certification test for a 4-bit secret split in 3 shares are in Figure 6. As previously, the Gaussian profiling exhibits model errors in the skewness and kurtosis that are not captured by its templates. Since the only informative moment in this case is the skewness (see the lower part of Figure 8), it directly implies that this model is unable to extract information from the simulated masked implementation. This is also witnessed by plotting the convergence of the PI metric, in the lower part of Figure 7, and intuitively by looking at the distribution in Appendix A, Figure 13.

More interestingly, we see that for the Gaussian mixture profiling, there are actually model errors detected in the mean and variance for low number of traces, that vanish as this number of traces increases. This in fact corresponds to the aforementioned expectation that the estimation of a complex (here, Gaussian mixture) model may be more measurement intensive than the estimation of its statistical moments. In such cases, the leakage certification tool even allows detection estimation errors. Quite naturally, this observation is confirmed by looking at the (lower) PI plot of Figure 7. It in fact quite nicely matches it since the saturation of the PI curve for the Gaussian mixture model (roughly) corresponds to the number of traces for which no errors are detected anymore in Figure 6.

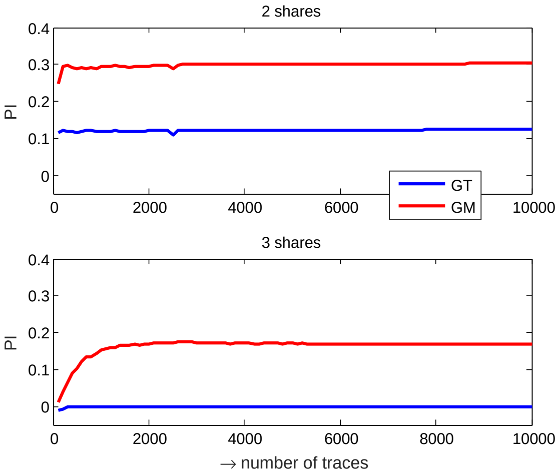

Fig. 7: Convergence of the PI for the Gaussian and Gaussian mixture profiling (simulated measurements).

We performed similar experiments for the 2-share masking, with similar intuitions. The most striking observation in this case relates to the information lying in the variance of the leakage distribution that is captured by the Gaussian templates. Since our experiments were carried out for a low noise level, such that the Gaussian and Gaussian mixture models significantly differ (see again Figure 13 in Appendix A), it implies that the latter model allows extracting more information (see the top of Figure 7). This is in line with the previous results in [12] where such an improved information extraction difference was also exhibited for low noise levels (while it typically vanishes with larger noise levels, since the Gaussian mixture then becomes close to Gaussian). Eventually, the result of the leakage certification for the 2-share experiment is also reported in

is perfect / without assumption errors). Second, concretely estimating Gaussian mixtures for our hardware masked implementations with transition-based leakages would be measurement-intensive (since we would typically need to build templates for $2^{12} \times 2^{12}$ transitions). Note that an alternative would be to consider the LR-based profiling from [14], which we leave as an interesting scope for further research.

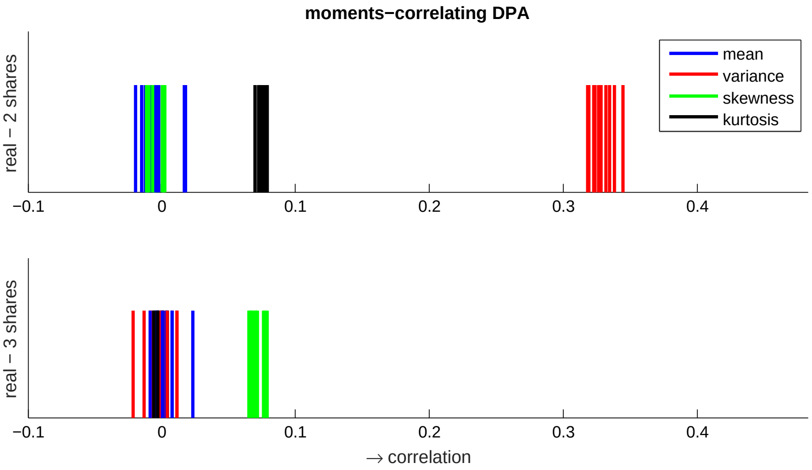

Fig. 8: MCP-DPA results for masked simulations.

Appendix A, Figure 14, and the corresponding MCP-DPA result is in the top of Figure 8. In this respect, note that the correlation-based rule-of-thumb given in Section 7 is not accurate here, as reflected by the larger PI difference between the Gaussian and Gaussian mixture profiling than suggested by the MCP-DPA values. As previously mentioned, this is simply due to the (too) low noise levels for the equivalence between the PI and correlation metrics to be accurate.

11 Conclusion

The evaluation of leaking devices against DPA attacks exploiting statistical models of leakage distributions implies answering two orthogonal questions: (1) is the model used in the attack/evaluation correct? (2) how informative is the model used in the attack/evaluation? The second question is highly investigated. It relates to the concrete security level of an implementation *given* a model, e.g. measured with a number of samples needed to recover the key. The first question is much less investigated and relates to the risk of a “false sense of security”, i.e. evaluations based on non-informative models despite informative leakages. Leakage certification allows evaluators to guarantee that the models used in their DPA attacks are sufficiently accurate. The simple tests we provide in this paper makes it possible to easily integrate leakage certification in actual toolchains. We hope these results open the way towards globally sound evaluations for leaking devices, where one first guarantees that the models used in the attacks are correct, and then evaluates their informativeness, which boil downs to compute their corresponding PI [7].

Interesting scopes for further research include the extension of the tools in this paper to more case studies of protected implementations with higher-order and multivariate leakages, and the investigation of the profiling errors due to the characterization of different devices, possibly affected by variability [21].

Acknowledgements. François-Xavier Standaert Standaert is a research associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in parts by the European Commission through the ERC project 280141 (acronym CRASH) and by the ARC project NANOSEC.

A Supplementary material

References

1. http://perso.uclouvain.be/fstandae/PUBLIS/171.zip.
2. http://satoh.cs.uec.ac.jp/sakura/index.html.
3. Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual information analysis: a comprehensive study. *J. Cryptology*, 24(2):269–291, 2011.
4. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.
5. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.
6. Guillaume Dabosville, Julien Doget, and Emmanuel Prouff. A new second-order side channel attack based

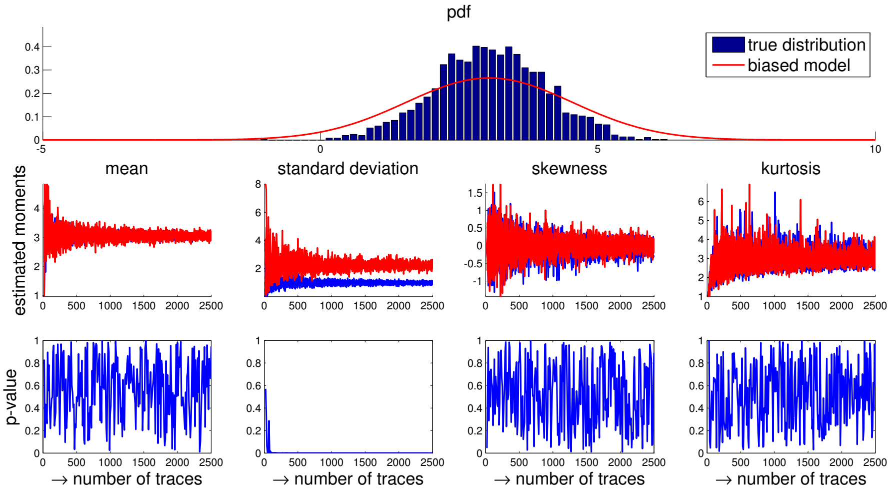

Fig. 9: Gaussian leakages, Gaussian model, error in the estimated variance.

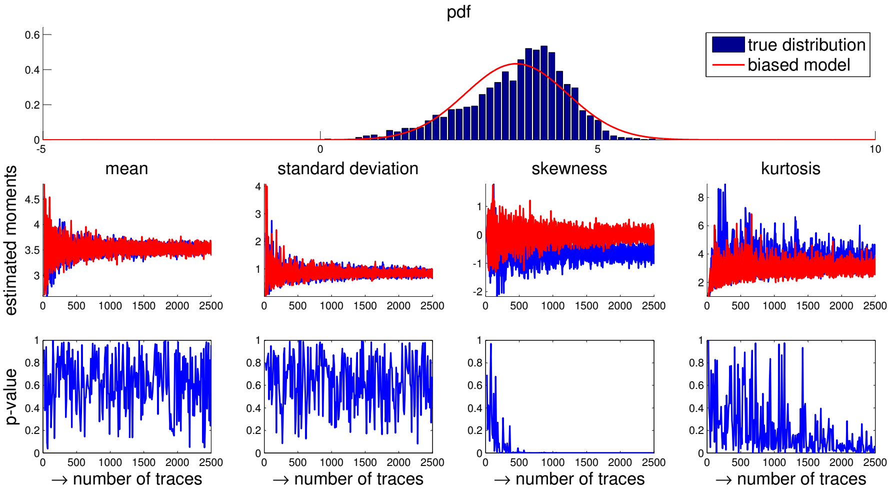

Fig. 10: Gaussian mixture leakages, Gaussian model, error in the estimated skewness.

on linear regression. *IEEE Trans. Computers*, 62(8):1629–1640, 2013.

7. Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 401–429. Springer, 2015.
8. François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In Marc Fischlin and Jean-Sébastien Coron, editors, *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*, volume 9665 of *Lecture Notes in Computer Science*, pages 240–262. Springer, 2016.
9. François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to certify the leakage of a chip? In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 459–476. Springer, 2014.
10. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 293–302. IEEE Computer Society, 2008.
11. Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for side channel resistance validation. NIST non-invasive attack testing workshop, 2011. `http://csrc.nist.gov/news_events/`

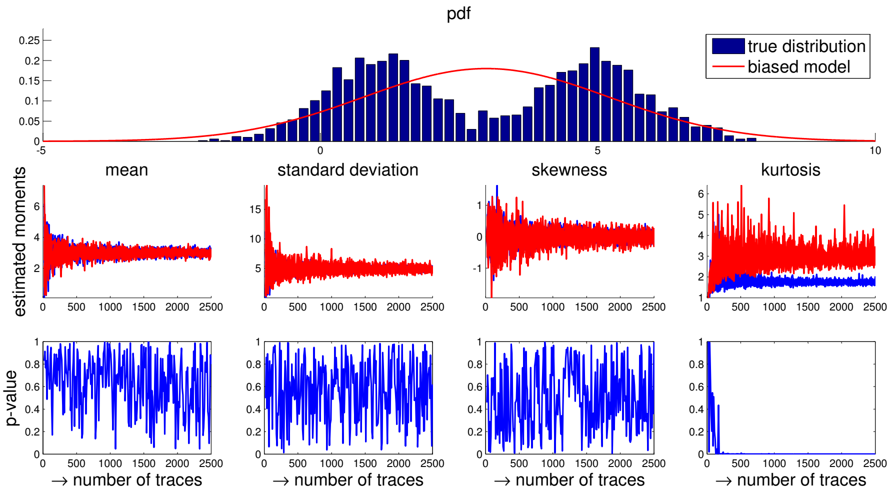

Fig. 11: Gaussian mixture leakages, Gaussian model, error in the estimated kurtosis.

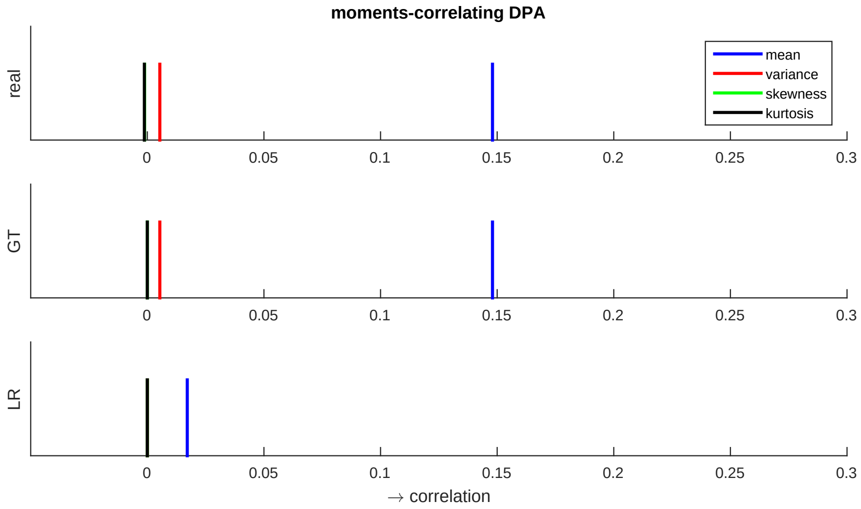

Fig. 12: MCP-DPA results for unprotected hardware.

`non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf`.

12. Vincent Grosso, François-Xavier Standaert, and Emmanuel Prouff. Low entropy masking schemes, revisited. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, volume 8419 of *Lecture Notes in Computer Science*, pages 33–43. Springer, 2013.
13. Annelie Heuser, Olivier Rioul, and Sylvain Guilley. Good is not good enough - deriving optimal distinguishers from communication theory. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 55–74. Springer, 2014.
14. Kerstin Lemke-Rust and Christof Paar. Analyzing side channel leakage of masked implementations with stochastic methods. In Joachim Biskup and Javier Lopez, editors, *Computer Security - ESORICS 2007, 12th European Symposium On Research In Computer Security, Dresden, Germany, September 24-26, 2007, Proceedings*, volume 4734 of *Lecture Notes in Computer Science*, pages 454–468. Springer, 2007.
15. Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.
16. Luke Mather, Elisabeth Oswald, Joe Bandenburg, and Marcin Wójcik. Does my device leak information? an a priori statistical power analysis of leakage detection tests. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Con-*

ference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I, volume 8269 of *Lecture Notes in Computer Science*, pages 486–505. Springer, 2013.

17. Luke Mather, Elisabeth Oswald, and Carolyn Whitnall. Multi-target DPA attacks: Pushing DPA beyond the limits of a desktop computer. In Palash Sarkar and Tetsu Iwata, editors, *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*, volume 8873 of *Lecture Notes in Computer Science*, pages 243–261. Springer, 2014.
18. Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. *IACR Cryptology ePrint Archive*, 2014:409, 2014.
19. Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-channel resistant crypto for less than 2, 300 GE. *J. Cryptology*, 24(2):322–345, 2011.
20. Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.
21. Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 109–128. Springer, 2011.
22. Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.
23. Tobias Schneider and Amir Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.
24. Tobias Schneider, Amir Moradi, François-Xavier Standaert, and Tim Güneysu. Bridging the gap: Advanced tools for side-channel leakage estimation beyond gaussian templates and histograms. *IACR Cryptology ePrint Archive*, 2016:719, 2016.
25. Aris Spanos. *Probability theory and statistical inference: econometric modeling with observational data*. Cambridge University Press, 1999.
26. François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.
27. François-Xavier Standaert, Eric Peeters, Gaël Rouvroy, and Jean-Jacques Quisquater. An overview of power analysis attacks against field programmable gate arrays. *Proceedings of the IEEE*, 94(2):383–394, 2006.
28. François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.
29. Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, volume 7707 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2012.
30. Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Security evaluations beyond computing power. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 126–141. Springer, 2013.
31. Carolyn Whitnall and Elisabeth Oswald. A fair evaluation framework for comparing side-channel distinguishers. *J. Cryptographic Engineering*, 1(2):145–160, 2011.

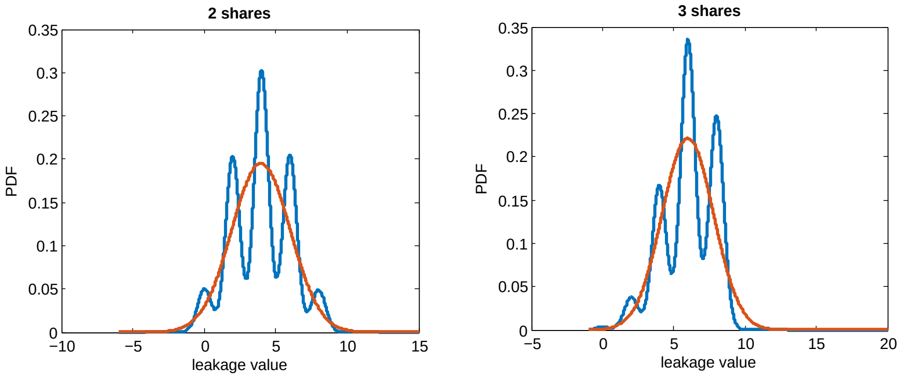

Fig. 13: Exemplary leakage distributions for masked simulations.

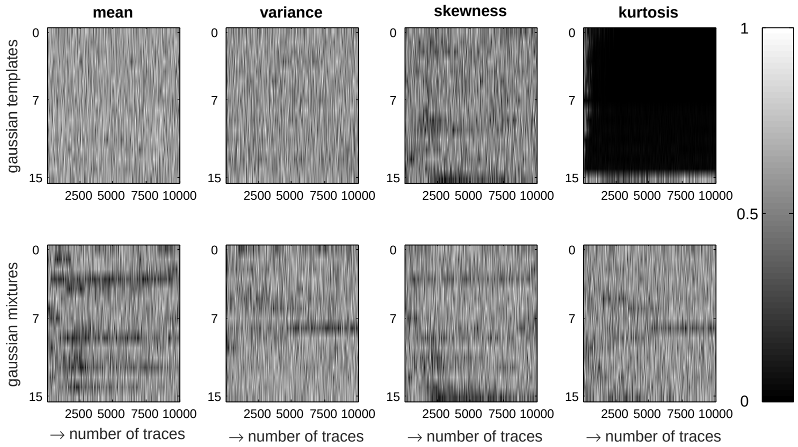

Fig. 14: Results of the new leakage certification test with masked simulations (2 shares).