This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2022.3144871, IEEE Transactions on Information Forensics and Security

Efficient Profiled Side-Channel Analysis of Masked Implementations, Extended

Olivier Bronchain[1], François Durvaux[1,2], Loïc Masure[1], François-Xavier Standaert[1]
[1] ICTEAM Institute, Crypto Group, Université catholique de Louvain, Louvain-la-Neuve, Belgium.
[2] Silex Insight, Louvain-la-Neuve, Belgium.

Abstract—We extend the study of efficient profiled attacks on masking schemes initiated by Lerman and Markowitch (TIFS, 2019) in different directions. First, we study both the profiling complexity and the online attack complexity of different profiled distinguishers. Second, we extend the range of the noise levels of their experiments, in order to cover (higher-noise) contexts where masking is effective. Third, we further contextualize the investigated distinguishers (e.g., in terms of adversarial capabilities and a priori assumptions on the leakage probability density function). Finally, we complete the list of distinguishers considered in this previous work and add expectation-maximization, soft analytical side-channel attacks and multi-layer perceptrons in our comparisons.
Our results allow shedding an interesting new light on the respective strengths and weaknesses of these different statistical tools, both in the context of a side-channel security evaluation and for concrete attacks. In particular, they confirm the experimental relevance of evaluation shortcuts leveraging the masking randomness during profiling, in order to speed up the evaluation process.

Index Terms—Side-Channel Analysis, Masking Countermeasure, Profiled Attacks, Machine Learning, Security Evaluations

1 Introduction

1.1 State-of-the-art & research problem

Side-Channel Attacks (SCAs) are among the most important threats against the security of cryptographic embedded devices. Such attacks exploit the unintended leakages of an implementation in order to recover sensitive information such as secret keys [1], [2]. As a result of their publication in the late 1990s, a vast body of work has investigated solutions to mitigate them, ranging from heuristic solutions (trying to hide the leakages in noise) to more formal ones [3]. Among the more formal solutions, *masking* has emerged over the last decade as an essential building block of side-channel resistant implementations. Its underlying principle is to split any sensitive variable of an implementation into d independent shares, and to perform all the operations on those shares only. As a result, the adversary is forced to "combine" the leakage of all the shares in order to recover sensitive information. Intuitively, if the leakages of these shares are sufficiently noisy and independent, exploiting this combination implies the estimation of a dth-order statistical moment. Formally, it has been shown that (under the noise and independence assumptions) masking ensures a security level that increases exponentially in the number of shares while requiring a quadratic performance overhead [4], [5], [6], [7], [8], [9]. Yet, ensuring these noise and independence conditions in practice turns out to be non-trivial and various types of imperfections can break them. Some problems already appear (and can be prevented) at the algorithmic level. For example, a lack of randomness can break the independence condition when composing gadgets in the abstract probing model (and be repaired in the same model) [10], [11], [12]. Some problems such as glitches are of more physical nature but can be prevented at the algorithmic level [13], [14]. Eventually, couplings [15], [16], [17] and a lack of noise in the leakages [18], [19] require lower-level abstractions for their analysis and can only be mitigated thanks to implementation tweaks.

In view of this state of the art, an equally large body of work has investigated tools in order to evaluate masked implementations. At a high-level, the first question is to determine the goal of the evaluation, which can range to qualitative detection to quantitative attacks or proofs.

Roughly speaking, leakage detection aims to answer the question "*does my device leak information?*", independently of whether this information can be exploited in an efficient attack. This can lead to fast conformance-based testing at the cost of a usually harder interpretation, especially in cases where no detection occurs [20], [21], [22], [23], [24], [25], [26]. Alternatively, attack-based evaluations aim at quantifying the amount of measurements (and time) needed to perform a full key recovery. They can take advantage of a wide variety of distinguishers introduced in the literature, such as Template Attacks (TAs) [6], [27], [28], [29], Correlation Power Analysis (CPA) [30], [31], [32], Linear Regression (LR)-based attacks [33], [34], [35] or Mutual Information Analysis (MIA) [36], [37], [38], [39]. Proof-based evaluations finally aim at bounding the security level of an implementation based on the aforementioned formal security guarantees, and typically work by "extrapolating" security evaluations obtained for low number of shares to larger number of shares. To a large extent, these approaches are complementary and correspond to a tradeoff between the time and expertise needed to perform the evaluations and the confidence that they provide. In this paper, we are concerned with attack-based evaluations which are at the core of any quantitative side-channel security evaluation.

Worst-case side-channel attacks essentially require to have a perfect understanding of the target device, usually reflected in the literature as a *leakage model*. Indeed, in case an exact leakage model is available, it is possible to perform an attack that maximizes the likelihood of the target key given the observed leakages [40], [41], [42], [43]. In this context, one recurrent source of discussion in the literature relates to the definition of relevant adversarial capabilities. To give one prominent example, side-channel attacks can be *profiled* — in which case the adversary can use a device she controls in order to estimate an accurate leakage model from collected measurements, or non profiled – in which case the model is based on engineering intuition. Typically, TAs and LR-based attacks are profiled, while CPA and MIA are non-profiled. Whenever trying to estimate the worst-case security level, the profiled setting is preferable, since there is no generic attack strategy that can succeed against any device without profiling [44]. This has led Lerman and Markowitch to initiate a study of efficient profiled attacks against masking schemes as a central ingredient for the security evaluation of cryptographic implementations, and to propose a systematic comparison of parametric and non-parametric distinguishers for this purpose [45].

1.2 Contributions

In this paper, we observe that despite initiating a necessary systematization effort, the work by Lerman and Markowitch is still limited in a few directions. We therefore extend their results with the following contributions:

- *Easier to interpret metrics.* Lerman *et al.*'s analyzes compare profiling methods by first fixing a number of profiling and attack measurements followed by an evaluation of the attacks' probability of success [45]. While this is a natural first step, it does not give an easily exploitable intuition about the profiling complexity of distinguishers (since this complexity is fixed rather than parametrized). To solve this limitation, we use an information theoretic approach in order to efficiently assess the profiling complexity and the attack complexity of different tools [46], [47]. In particular, we use the Perceived Information (PI) metric introduced by Renauld et al. [48] and further formalized in [40], [43] as a natural indicator of the *profiling complexity* (i.e., the number of measurements needed to reach a positive PI) and the *attack efficiency*, reflected by the asymptotic PI value – which is inversely proportional to the number of traces needed to perform a key recovery [9], [49]. As a result, and compared to the success rate centric methodology used by Lerman *et al.*, the methodology of this work allows omitting the attack data complexity to estimate the metrics used to compare the distinguishers.[1].
- *Larger noise levels.* Lerman and Markowitch consider very low to low noise levels, typically in a range limiting the effectiveness of the masking countermeasure. We consider noise levels ranging from low – corresponding to a Signal-to-Noise Ratio (SNR) of 10 – to medium (SNR=1) and large (SNR=0.1), reflecting both insecure and secure masked implementations.
- *Improved contextualization.* Lerman and Markowitch consider parametric and non-parametric distinguishers. But a discussion of what is the impact of the adversarial capabilities, for example regarding the control of the masking randomness during profiling, is still lacking. The latter limits the usability of the results in helping evaluators to determine which tool to use in which situation. We further contextualize the attack settings based on adversarial capabilities and consider profiled attacks with known or unknown masking randomness.
- *Additional distinguishers* The authors of [45] evaluate Gaussian TAs, Kernel Density Estimation (KDE) and Random Forests (RFs), but some natural candidates for the efficient profiling of masked implementations are still missing, for example the Expectation-Maximization (EM) algorithm to profile Gaussian mixtures [28], [50], Soft Analytical Side-Channel Attacks (SASCA) [51] and Multi-Layer Perceptrons (MLP) [52], [53]. We include these distinguishers in our experiments.[2]

1. Lerman *et al.* study the estimation/assumption errors of profiled models with respect to the mean square error, leveraging the bias-variance decomposition paradigm for this metric. We pursue a similar goal but use PI instead, which is a natural candidate for this task [40]

2 Background

2.1 Notation

Random variables are denoted with capital letters X and their realizations with small caps x. Random vectors, such as leakage traces $\mathbf{L}$, are denoted in bold, leading to realisations $\mathbf{l}$. Probability for a given realization of X is written as $\Pr[\mathrm{X} = x]$ and if clear from the context as $\mathsf{p}(x)$. The continuous distribution of a random variable conditioned to another one is denoted as $\mathsf{f}(\mathbf{l} \,|\, x)$. Estimations are denoted with a hat. The Boolean (bitwise) addition is denoted with $\oplus$ while addition over the reals is denoted with $+$.

2.2 Information theoretic metrics

Next, we introduce the information theoretic (IT) metrics we use with their interest for side-channel evaluations.

Mutual Information. In general, determining the minimum number $N^{\star}_{\mathsf{kr}}$ of measurements that an adversary must observe to recover a secret (e.g., key) is an important part of a side-channel security evaluation. For the best possible (i.e., worst-case) attack, this quantity can be linked to the MI:

$$N^{\star}_{\mathsf{kr}} \geq \frac{c(sr, \kappa)}{\mathsf{MI}(\mathbf{L}; \mathrm{X})},$$

where $c(sr, \kappa)$ is a small constant that depends on the success rate sr and the bitsize of the target secret κ [9], [49]. The MI is then defined as:

$$\mathsf{MI}(\mathbf{L}; \mathrm{X}) = \mathsf{H}(\mathrm{X}) + \sum_{x \in \mathrm{X}} \mathsf{p}(x) \cdot \sum_{\mathbf{l} \in \mathrm{L}} \mathsf{p}(\mathbf{l} \,|\, x) \cdot \log_2 \mathsf{p}(x \,|\, \mathbf{l}),$$

where $\mathsf{H}(\cdot)$ denotes Shannon's entropy and $\mathsf{p}(\cdot \,|\, \cdot)$ is the Probability Mass Function (PMF) derived by applying Bayes's rule to $\mathsf{f}(\cdot \,|\, \cdot)$. Despite its theoretical interest, directly

2. We also considered other non-parametric tools such as Gaussian Processes [54], but due to the large size of our profiling sets, it did not improve over other tools so we do not include these results in our comparison and leave the optimization/specialization of such powerful machine learning tools as an interesting scope for further research.

computing this quantity is not possible because it requires knowing the leakage distribution, which adversaries and evaluators do not know a priori.[3] Practical evaluations therefore estimate this leakage distribution. The qualily of this estimation can then be quantified with the PI.

Perceived Information. The PI quantifies the amount of information that can be extracted with an estimated model $\widehat{\mathsf{f}}(\mathbf{l} \,|\, x)$. It can be evaluated "by sampling" as:

$$\widehat{\mathsf{PI}}_{n_t}(\mathbf{L}; \mathsf{X}) = \mathsf{H}(\mathsf{X}) + \sum_{x \in \mathsf{X}} \mathsf{p}(x) \cdot \frac{1}{n_t(x)} \sum_{i=1}^{n_t(x)} \log_2 \widehat{\mathsf{p}}(x \,|\, \mathbf{l}_i^x),$$

where $n_t(x)$ is the number of "test" leakage vectors $\mathbf{l}^x$ corresponding to a realisation $\mathsf{X} = x$ and the true PMF is replaced by its corresponding estimation $\widehat{\mathsf{p}}(x \,|\, \mathbf{l}_i^x)$. For the estimation to be unbiased, it is important that the test leakage vectors $\mathbf{l}_i^x$ are fresh validation traces that have not used to build the PDF estimate.[4] As shown by Bronchain et al., the PI is upper-bounded by MI so that $\mathsf{MI} \geq \mathsf{PI}$ [43]. Informally, the PI metric reflects the quality of the model: if the model is perfect, then the equality is met and the attack is worst-case; otherwise the attack is suboptimal.

2.3 Boolean masking

General principle. Boolean masking is a popular countermeasure against side-channel attacks. It consists in representing all the sensitive variables x of a leaking implementation as encodings $\{x\}$ which are collections of d shares x^j such that $x = \sum_{j=0}^{d-1} x^j$. The minimum target property of most masking schemes is called *probing security*: it ensures that an adversary who can observe $d-1$ noise-free intermediate values (i.e., probes) in an entire implementation cannot learn any information about x. For encodings, it is easily obtained since regardless the value of x, every x^j is uniformly distributed, so any subset of at most $d-1$ shares remains independent of the x. Computing on encodings while preserving probing security is more challenging but not necessary for our following investigations.

Under the assumption that the actual shares' leakages are sufficiently noisy and independent, probing security ensures that for any side-channel adversary having access to physical observations for all the intermediate variables, the complexity of the attack grows exponentially in d [7], [8], [9]. More precisely, the impact of masking on the MI is reflected by the following inequality:

$$N_{\mathsf{kr}}^{\star} \geq \frac{c}{\prod_{j=0}^{d-1} \mathsf{MI}(\mathbf{L}; \mathsf{X}^j)}, \tag{1}$$

where $\mathsf{MI}(\mathbf{L}; \mathsf{X}^j)$ is the information on a single share [9].

Impact on leakage PDF. Since masking involves randomness, the true leakage PDF $\mathsf{f}(\mathbf{l} \,|\, x)$ becomes a mixture with the following expression:

$$\mathsf{f}(\mathbf{l} \,|\, x) = \sum_{\{x\}} \mathsf{p}(\{x\}) \cdot \mathsf{f}'(\boldsymbol{l} \,|\, \{x\}). \tag{2}$$

It is therefore a weighed sum of $|\mathsf{X}|^d$ terms, each of them being the leakage PDF conditioned to an encoding $\{x\}$. The weights are the probabilities $\mathsf{p}(\{x\})$ that the processed encoding of x is $\{x\}$. In our context, where the randomness is (assumed) ideal, this probability is uniform.

2.4 Profiled distinguishers

We now detail various methods to estimate a model PMF $\widehat{\mathsf{p}}(x \,|\, \boldsymbol{l})$ or PDF $\widehat{\mathsf{f}}(\mathbf{l} \,|\, x)$ from profiling samples $(\boldsymbol{l}, \{x\})$.[5]

2.4.1 Gaussian Mixture Template Attack (GMTA).

A first solution to build a model is simply to estimate all the PDFs in Equation 2. The complete estimated conditional PDF of x is then expressed as:

$$\widehat{\mathsf{f}}(\boldsymbol{l} \,|\, x) = \sum_{\{x\}} \mathsf{p}(\{x\}) \cdot \hat{\mathsf{f}}'(\boldsymbol{l} \,|\, \{x\}), \tag{3}$$

where every $\hat{\mathsf{f}}'(\cdot|\cdot)$ is the estimated conditional PDF for one component of the mixture. In this work, we propose to use the usually considered Gaussian approximation for this PDF and next call the resulting distinguisher GMTA.

We note that this approach is computationally intensive. During the profiling phase, a template must be built for every possible encoding $\{x\}$. This increases the total number of templates to $|X|^d$ where $|X|$ is the number of possible values for x. During the attack phase, in order to recover $\widehat{\mathsf{f}}(\boldsymbol{l} \,|\, x)$, the mixture must be explicitly computed and therefore requires $\mathcal{O}(|X|^d)$ operations per attack trace.

2.4.2 Multi-Layer Perceptron (MLP).

A Multi-Layer Perceptron aims at directly approximating the discriminative model $\widehat{\mathsf{p}}(x \,|\, \boldsymbol{l})$ – seen as a multivariate function of $\mathbf{l}$ – thanks to a composition of elementary functions (a.k.a. *layers*), alternating between *linear* layers (denoted as λ), non-linear element-wise *activation functions* (denoted as σ), and a final *softmax* normalization layer (denoted as s). More precisely, in this study we use MLPs with only one intermediate layer, as follows:

$$\widehat{\mathsf{p}}(x \,|\, \mathbf{l}) = \mathrm{s} \circ \lambda_{|\mathsf{X}|} \circ \sigma \circ \lambda_C(\mathbf{l}) \ , \tag{4}$$

where C denotes the number of neurons in the hidden layer, i.e., the output dimension of the first linear layer.

Each layer is fully described by inner parameters whose values are tuned during the profiling phase. This step is done by maximizing the likelihood of the model outputs given the profiling data providing the ground truth.

Such models are known to be very expressive, since the approximation error can be made arbitrarily small by increasing C, provided that λ is not a polynomial [55]. Thus, MLPs represent a powerful tool when no further hypothesis can be made on the true leakage model to approximate. As a drawback, they may require more profiling data to tune their parameters. In our experiments, we chose *ReLU* as an activation function, and $C = 1,000$, similarly to the settings used in the previous work of Masure *et al.* [53].

3. Excepted in the case of simulated attacks, which therefore come in handy for comparing distinguishers (as we will do next), but do not help for the analysis of concrete implementations.

4. Note that the same sampling strategy can be used in order to estimated the MI, which can be useful in case its direct computation becomes intensive (e.g., due to large dimensionalities).

5. Recall that the PMF can be derived from the PDF using Bayes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2022.3144871, IEEE Transactions on Information Forensics and Security

2.4.3 Soft Analytical Side-Channel Attack (ESASCA).

In order to limit the computational complexity of GMTAs, an alternative solution is to express every $\hat{\mathsf{f}}'(\cdot|\cdot)$ from the (independent) leakage on each of the shares in $\{x\}$, leading to the following expression:

$$\widehat{\mathsf{f}}(\boldsymbol{l} \,|\, x) = \sum_{\{x\}} \mathsf{p}(\{x\}) \cdot \prod_{j=0}^{d-1} \hat{\mathsf{f}}'(\boldsymbol{l} \,|\, x^j), \tag{5}$$

where $\hat{\mathsf{f}}'(\boldsymbol{l}, x_i)$ is the PDF estimated for the share x_i. The latter can be viewed as an application of Soft Analytical Side-Channel Attacks (SASCA) limited to an encoding rather than an entire circuit [51]. This strategy has demonstrated its interest in the context of masked software implementations [56]. We next denote it as Encoding-only SASCA (ESASCA). ESASCA is a SASCA for which the factor graph has a tree structure with the root being the secret and the leaves begin the shares. In the case where $\hat{\mathsf{f}}'(\boldsymbol{l}, x_i)$ is estimated with a Gaussian distribution, we call it G-ESASCA; in the case MLPs are used, we call it MLP-ESASCA.

The main advantage of this strategy is that during profiling, the PDFs do not need to be estimated on the entire encoding but only on the shares independently. As a result, the number of templates is reduced down to $d \cdot |X|$.

Its main drawback is that by making such an independent estimation of the leakage PDF for each component, ESASCA is unable to detect flaws due to physical defaults (like glitches or couplings) that could reduce the statistical security order of the implementation (defined as the highest statistical moment of the leakage distribution that is independent of the secret). We call such a strategy *order-preserving* to reflect the fact that it targets the maximum statistical security order of the masking scheme. We mention that a similar approach (with same advantages and drawbacks) can be applied to other masking schemes [57].

2.4.4 Expectation-Maximization (EM).

The next approach we study is based on the EM algorithm [28], [50]. It is an iterative procedure that allows estimating the parameters of a mixture of Gaussian PDFs and can therefore be used to model Equation 2. The parameters that the EM algorithm has to estimate are the means and the covariances of each of the components $\widehat{\mathsf{f}}(\boldsymbol{l} \,|\, \{x\})$, the weight of each component being known and equal to $\{\mathsf{x}\}$. During the profiling phase, one mixture must be estimated per possible x meaning that EM must be executed $|X|$ times. Each EM execution has to model a mixture with $|X|^{d-1}$ components. In our experiments, the EM algorithm is ran for a maximum of 200 iterations. During the attack phase, the mixture is explicitly computed as in a GMTA, leading to a computational complexity of $\mathcal{O}(|X|^d)$. The adversary / evaluator could model a smaller number of components leading to more (computationally) efficient heuristic attacks.

2.4.5 Kernel Density Estimator (KDE).

For completeness, we briefly recall hereafter the principles of some other estimators considered by Lerman and Markowitch, that we use in this paper for comparison. The first one is a non-parametric method called KDE. It estimates the leakage pdf thanks to a *kernel* function, computed based on the empirical leakage distribution of the profiling traces. In our experiments, we used a Gaussian kernel with a bandwidth of 1. We refer the reader to Lerman and Markowitch's paper for more technical details [45, Sec. II.B.3]. The main advantage of such estimators is that they do not rely on strong assumptions regarding the leakage – *e.g.*, it may be assumed not to be Gaussian, contrary to parametric methods. Unfortunately, this comes with a major drawback, since they suffer from the so-called *curse of dimensionality*. This phenomenon informally states that the estimation error of KDE scales with $\mathcal{O}\left(n^{-\frac{p}{p+|X|}}\right)$, where p denotes a smoothness parameter [58], [59]. Accordingly, unless assuming that the leakage belong to a class of very smooth functions, the estimation error increases exponentially with the leakage dimensionality – and thereby the masking order.

2.4.6 Random Forest (RF).

The last type of estimator we consider in this study is also considered by Lerman and Markowitch in their initial work. Random Forests (RF) is a particular case of ensemble method, where a collection of *weak* estimators – *i.e.* performing slightly better than randomness – are combined together, *e.g.* with a majority rule, to provide a meta-estimator. In the case of RF, the weak estimators are some decision trees. Provided that the latter ones are not fully correlated, the meta-estimator can reach higher levels of accuracy [60]. In our experiments, we use the same setting as Lerman and Markowitch, namely with 100 trees, each of depth equal to 10. Since the output scores are computed by *pro rata* of the outcomes of each decision tree, the number of decision trees must be set depending on the required resolution in the output scores. Intuitively, the more trees in the random forest, the higher the score resolution. A higher resolution may be needed if the bit size κ or the noise level increases.

3 Methodology

In this section, we detail the methodology we used in order to compare profiled side-channel distinguishers against masked implementations. We first describe our simulated attack framework and follow with a description about how we use the PI metric to compare distinguishers.

3.1 Simulation framework

Our experimental analyzes are based on simulated leakages. The rationale behind this choice is similar to the one of Lerman and Markowitch [45]. Since we aim to compare distinguishers, it allows us to evaluate them in different well-controlled scenarii (e.g., low-noise, high-noise), which considerably simplifies the interpretation of the results. We note however that all the distinguishers we consider have been applied to real traces in previous works (e.g., [61] for GMTA, [53] for MLP, [56] for ESASCA and [34] for EM).

More precisely, we first consider a simulated masked (flawless) implementation as graphically represented in Figure 1. To simulate the leakage of a secret κ-bit variable x, $d-1$ first shares are drawn at random from the set of all κ-bit strings $\{0,1\}^\kappa$. The last share is set with $x^{d-1} = x \oplus \sum_{j=0}^{d-2} x^j$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2022.3144871, IEEE Transactions on Information Forensics and Security

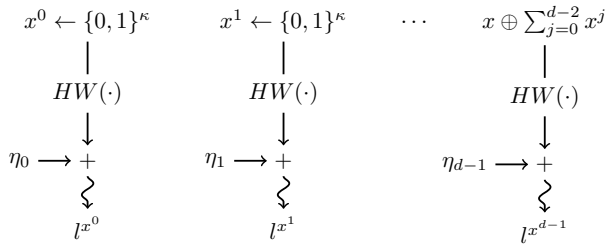


Figure 1: Simulation of masked a implementation.

in order to ensure that $x = \sum_{j=0}^{d-1} x^j$. The leakage vector $\boldsymbol{l}$ is then defined as the concatenation of all the leakage $l^{x^j} = HW(x^j) + \eta_j$ where $HW(\cdot)$ is the Hamming weight function and η_j is a Gaussian noise with variance σ^2.

Since some of the distinguishers we consider are order-preserving, we additionally consider the case of a flawed masked implementation, in order to determine the security gaps that may appear when targeting the maximum security order despite physical defaults. We use the setting of [9] for this purpose, and assume a 2-share implementation with a first-order flaw. In this case, the simulated leakages additionally contain a value $\mathrm{l}^x = f \cdot HW(x) + \eta$. The noise η is again a Gaussian random variable with variance σ^2. The parameter f captures the amplitude of the flaw (or its SNR). If $f = 0$, the independence assumption is met. As f increases, it is expected that the first-order leakage will be more and more dominating over the second-order one.

Overall, the proposed simulations take three parameters. The number of shares d, the noise variance σ^2 and the first order-flaw magnitude f. When relying on simulations, these parameters are known and so is the true leakage PDF $\mathsf{f}(\boldsymbol{l} \,|\, x)$ that is defined by Equation 2. As discussed in subsection 2.2, knowing this PDF allows us to compute the MI which is representative of the best-possible attack. In the next sections, it will enable us to compare the PI of the proposed distinguishers to this optimal MI value.

3.2 How to compare two distinguishers?

In this work, we make use of information theoretic metrics in order to compare profiled distinguishers. For this purpose, we denote as $\mathsf{PI}^{\hat{\mathsf{f}}}_n(\mathrm{X}, \mathbf{L})$ the PI of the model for which n profiling measurements are used to estimate the PDF $\hat{\mathsf{f}}(\mathbf{l} \,|\, x)$. We use a similar notation for the PMF $\hat{\mathsf{p}}(x \,|\, \mathbf{l})$. We then consider the following two criteria:

Profiling complexity. The first way to compare distinguishers is to evaluate their performances by fixing their profiling data complexity (i.e., the number of traces acquired during the profiling phase) to n. Comparing the PI values in this case answers the question: *"For a given data complexity n, what is the best model for the adversary?"*. That is, since having a larger PI implies an online attack with lower number of traces, we can deduce that if:

$$\mathsf{PI}^{\hat{\mathsf{f}}_1}_n \leq \mathsf{PI}^{\hat{\mathsf{f}}_2}_n, \tag{6}$$

then the best strategy for the adversary is to use $\hat{\mathsf{f}}_2(\cdot \,|\, \cdot)$. We stress that this comparison is conditional to n: a tool may be better for some a small n and not for a larger one. A natural way to compare profiling complexity is to extract the n values needed to reach a positive PI (i.e., a model that is sufficiently accurate to allow key recoveries).

Online attack complexity. The second way to compare distinguishers is to analyse the asymptotic PI value that is reach by a model. For this purpose, we denote as $\mathsf{PI}^{\hat{\mathsf{f}}}_\infty(X, \boldsymbol{L})$ the PI of a model for which n is sufficiently large for perfectly estimating all its parameters, which we assume to be the maximum reachable one. This value then indicates the complexity of the best online attack that can be performed with this model. Applied to a pair of models, it typically leads to inequalities like:

$$\mathsf{PI}^{\hat{\mathsf{f}}_1}_\infty \leq \mathsf{PI}^{\hat{\mathsf{f}}_2}_\infty \leq \mathsf{MI}. \tag{7}$$

From an evaluator's viewpoint, it answers the question: *"What is the best attack that can be performed if my profiling phase is sufficient for my model to converge?"*.

4 Distinguisher Classifications

Before performing quantitative (experimental) comparisons in the next section, we propose a systematic contextualization of the distinguishers we study. We consider both the tools studied by Lerman and Markowitch in [45] and the new ones we list in subsection 2.4. The resulting classification is sumarized in Table 1. We next develop it by first highlighting the importance of (masking) randomness knowledge during profiling and then discussing assumptions on the PDF that may reduce the profiling complexity, possibly at the cost of less efficient online attacks.

	Known masks	Gaussian distri.	Gaussian comps.	Order-preserving
MLP	✗	✗	✗	✗
RF	✗	✗	✗	✗
KDE	✗	✗	✗	✗
EM	✗	✗	✓	✗
GMTA	✓	✗	✓	✗
G-ESASCA	✓	✗	✓	✓
MLP-ESASCA	✓	✗	✗	✓
HO-GTA	✗	✓	✗	NA

Table 1: Classification of distinguishers according to the *a priori* assumptions made on the leakage distribution. ✓ means that the given algorithm relies on the given assumption to work, whereas ✗ means that the given assumption is not mandatory to run the given algorithm.

4.1 Randomness knowledge

Allowing an adversary / evaluator to know the randomness used during the profiling stage of a side-channel attack generally allows estimating a model with a smaller profiling data complexity n. Yet, an important question is to determine whether such an assumption is only a useful shortcut for evaluators (as promoted in [62]) or if it also creates a complexity gap in the online attacks. In other words, does this randomness knowledge only speed up evaluations, leading to online attacks that could also be reached by determined adversaries without this shortcut (with more profiling), or are there realistic examples where

profiling without masks knowledge cannot lead to the same online attacks as profiling with masks knowledge?

In order to contribute to this question, we evaluate two types of distinguishers. The first type does not require the knowledge of the randomness during profiling, and directly builds an estimate $\widehat{\mathsf{f}}(\cdot \mid \cdot)$ for the full PDF $\mathsf{f}(\cdot \mid \cdot)$, or $\widehat{\mathsf{p}}(\cdot \mid \cdot)$ for the full PMF $\mathsf{p}(\cdot \mid \cdot)$. The MLP, EM, Random Forest (RF) and Kernel Density Estimation (KDE) distinguishers typically fall in this category.[6] The same holds for Higher-Order Gaussian Template Attacks (HO-GTA) such as [31], [32]. The second type of methods does require the knowledge of $(\mathbf{l}, \{x\})$ during profiling, in order to decompose $\mathsf{f}(\cdot \mid \cdot)$ in multiple simpler PDFs $\mathsf{f}'(\cdot|\cdot)$. Indeed estimating $\mathsf{f}'(\cdot|\cdot)$ for each of the modes in the mixture requires to know which mode is the one corresponding to the profiling trace $\mathbf{l}$. The GMTA, G-ESASCA and MLP-ESASCA distinguishers are representatives candidates of this second category.

4.2 A priori PDF assumptions

Another natural way to speed up the profiling of a model (and possibly to make the online distinguishers more efficient) is to rely on good a priori assumptions on the leakage PDF $\mathsf{f}(\cdot \mid \cdot)$. As a counterpart, if these assumptions are not correct, the estimated distribution may not converge to to true PDF so that $\mathsf{PI}_\infty < \mathsf{MI}$, indicating that an online attack using this model will be suboptimal. We next detail two assumptions that we will consider in our experiments.

4.2.1 Gaussian components

We first note that assuming a Gaussian distribution is common when targeting unprotected devices [27]. However, when the masking countermeasure is implemented, it is expected that this assumption is not valid anymore, since the masking randomness turns the distribution into a mixture. A natural extension is to assume that the distribution of each component in the mixture is Gaussian so that:

$$\mathsf{f}'(\boldsymbol{l}|\{x\}) \approx \mathcal{N}(\boldsymbol{\mu}_{\{x\}}, \boldsymbol{\Sigma}_{\{x\}}). \tag{8}$$

As in the unprotected case, the interest of this assumption is that it reduces the parameters that must be estimated during profiling to only a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. Estimating this covariance has a cost that is quadratic in the size of $\boldsymbol{L}$, which is significantly cheaper than non-parametric estimators based on histograms or Kernels (of which the cost grows exponentially in this size). The EM, GMTA and G-ESASCA attacks take advantage of this assumption.

4.2.2 Independent shares' leakages

Eventually, another hypothesis about the leakage PDF that speeds up the profiling of a masked implementation is to assume that all the shares leak independently. As a result, each component of the PDF $\mathsf{f}(\cdot \mid \cdot)$ is approximated with:

$$\mathsf{f}'(\boldsymbol{l} \mid \{x\}) \approx \prod_{j=0}^{d-1} \hat{\mathsf{f}}'(\boldsymbol{l} \mid x^j), \tag{9}$$

where one single PDF $\hat{\mathsf{f}}'(\boldsymbol{l} \mid x^j)$ must be estimated per share. The main interest of this assumption is to scale gently with the number of shares. Namely, and as detailed in subsubsection 2.4.3, the number of templates to estimate grows linearly with the number of shares, while it grows exponentially for methods like GMTA (which also makes its computation during the online attack computationally intensive as the number of shares increases) [19].

It is important to note that the independent leakage assumption is usually considered as a *design assumption* that engineers implementing masking have to fulfill [9], [13], [14]. Failing to meet this assumption may lead the worst-case security level of an implementation to be lower than expected. By contrast, we here consider it as a *distinguisher assumption*. So analyzing an implementation under this assumption when it is not fulfilled may lead to a false sense of security (i.e., to the PI extracted with a model exploiting this assumption being lower than the MI). In particular, if the shares of a masked implementation do not leak (sufficiently) independently due to a physical default like glitches or couplings, an adversary using this assumption will not be able to detect and exploit this flaw. For this reason, we use the terminology "order-preserving" for the distinguishers assuming independent shares' leakages in Table 1. In this work, only the ESASCA-based attacks (i.e., G-ESASCA and MLP-ESASCA) are doing such an hypothesis.

5 Experimental (simulated) results

We now move to the presentation of our simulated experiments. We structure the section in three main parts. First, we estimate IT metrics to compare different profiled distinguishers in a setting corresponding to a "properly implemented" masked implementation (i.e., without flaw) with 2 and 3 shares. We use this experiment to discuss the profiling complexity and online attack complexity of attacks against the masking countermeasure, in function of the implementation context and the assumptions used by the distinguishers. Second, we validate our findings by running simulated attacks with these distinguishers and report the corresponding Guessing Entropy (GE) [46]. Eventually, we evaluate the extent to which the presence of a flaw in the masking can lead the order-preserving distinguishers to overstate the security level of an implementation in subsection 5.3. For all the following results, the sensible variables are 4-bit wide such that $\kappa = 4$. The meta-parameters for the distinguishers are additionally discussed in Appendix C.

For these purposes, we selected representative distinguishers from the categories of Table 1, namely MLP, EM, GMTA, G-ESASCA and MLP-ESASCA. We excluded the KDE and RF distinguishers from the IT analysis: the first one because it consistently led to significantly more computationally intensive attacks than the MLP-based one (see Appendix B) while not providing better results in terms of data complexity; the second one because it does not provide the probabilities needed in order to estimate our information theoretic metrics. Yet, both tools are considered in the GE estimations of subsection 5.2. We also did not consider HO-GTAs in our comparisons since they are based on the estimation of statistical moments rather than full

6. For the last two ones, we refer to [45].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2022.3144871, IEEE Transactions on Information Forensics and Security

distributions and we see their motivation as the assessment of a statistical security order rather than efficient attacks.[7]

5.1 Flawless masked implementation

The convergence of the PI metric estimated for our investigated distinguishers is given in Figure 2 for a 2-share implementation, with SNRs of 10, 1 and 0.1 and in Figure 3 for a 3-share implementation with SNRs of 10 and 1.[8] It leads to the following three main observations:

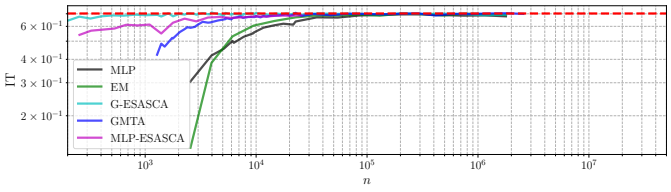
1) The value of the asymptotic PI reaches the value of the MI for all the investigated distinguishers. This implies that all these distinguishers can lead to worst-case attacks if profiled with a sufficient amount of traces. This observation is naturally explained by the fact that the assumptions exploited by some of these distinguishers are all fulfilled in this first simulated setting.
2) By contrast, the speed of convergence of the different distinguishers, and therefore their profiling complexity, significantly varies. As expected, the distinguishers that have the lowest profiling (data) complexity are also the ones that take advantage of more assumptions. Concretely, we observe that the G-ESASCA is the fastest, followed by MLP-ESASCA, GMTA, MLP and EM.
3) The ordering of the distinguishers in terms of profiling complexity is independent of the SNR and number of shares, but the quantitative gap between them increases as the SNR decreases and the number of shares increases (i.e., for better protected implementations).

The first point is particularly important regarding the question raised in subsection 4.1. Namely, it shows that for the flawless masked implementation simulated in this section, a determined adversary profiling without masks knowledge can reach the same attack efficiency as an evaluator leveraging this knowledge. So it confirms the masks knowledge as a useful shortcut for evaluations which does not affect the final security claims in terms of online attack complexity.

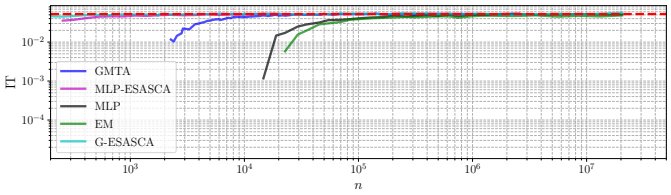
5.2 Validation with Guessing Entropy

We confirm the previous observations by running simulated attacks and by reporting the GE, which is the average position of the correct subkey k in the list of subkey candidates provided by an attack [46]. Precisely, in Figure 4 and Figure 5, we report the GE depending on the number of measurements used by the adversary. As expected, the GE decreases as the number of measurements N_a increases. These plots include the RF distinguisher, as well as the KDE one in settings whenever it was tractable (i.e., for attacks with low profiling complexity), which allows comparison with the results of Lerman and Markowitch [45].

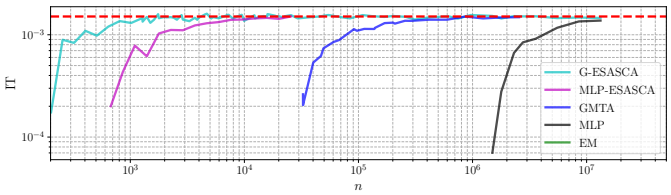
The attacks of in Figure 4 are performed for two shares and with $\mathsf{SNR} = 1$, which corresponds to the same simulated implementation as the PI curves of Figure 2b. The different plots correspond to different profiling (data) complexities n. As expected from the PI curves, $n = 10^3$ is



(a) $\mathsf{SNR} = 10$

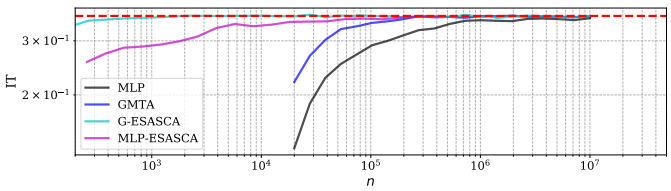


(b) $\mathsf{SNR} = 1$

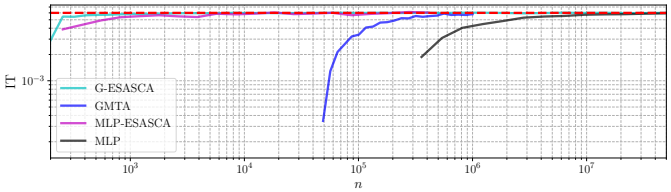


(c) $\mathsf{SNR} = 0.1$

Figure 2: Flawless 2-share implementation profiling.



(a) $\mathsf{SNR} = 10$



(b) $\mathsf{SNR} = 1$

Figure 3: Flawless 3-share implementation profiling.

sufficient for GT-ESASCA to converge. As n increases, all the distinguishers converge towards the results obtained with GT-ESASCA. The results of subsection 5.1 are therefore confirmed: distinguishers with additional hypotheses about the leakage distribution require a smaller profiling complexity when these hypotheses are fulfilled. A similar trend is observed for larger noise variances (e.g., $\mathsf{SNR} = 0.1$) as reported in Figure 5. Eventually we report the running time for each of the distinguishers in Appendix B.

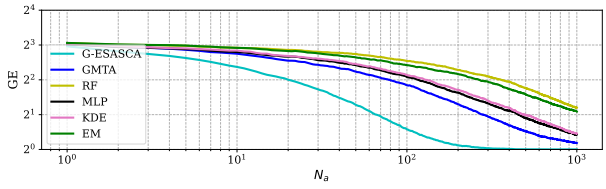
5.3 Flawed masked implementation

We complement the previous analysis with the case of a flawed masked implementation, in order to determine the extent to which the order-preserving distinguishers can overstate security. We focus on the online attack complexity reflected by the asymptotic PI value in this case, since a

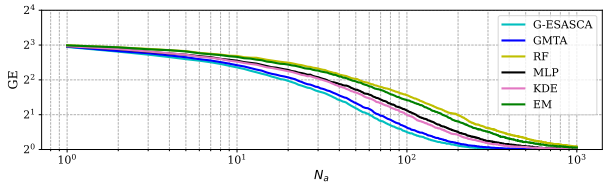
7. It is for example shown in [24] that they can require significantly more data to attack than a distribution-based distinguisher in low noise conditions, while it is shown in [6] that they reach the same data complexity as distribution-based distinguishers in high-noise conditions.

8. The y-axis called IT reports the MI and the PI for each of the distinguishers. The x-axis n reports the profiling data complexity.

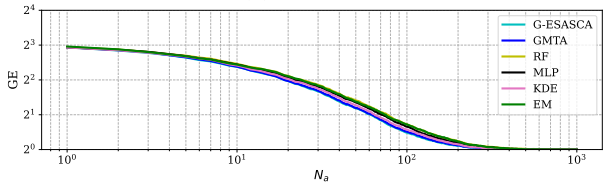
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2022.3144871, IEEE Transactions on Information Forensics and Security



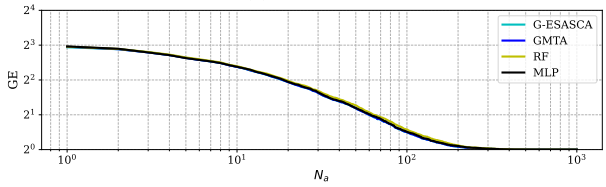
(a) $n = 10^3$.



(b) $n = 10^4$.

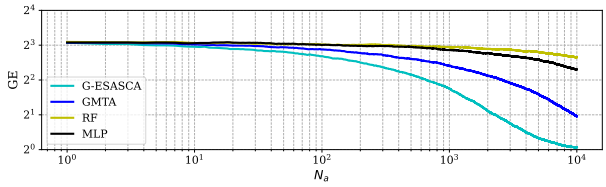


(c) $n = 10^5$.

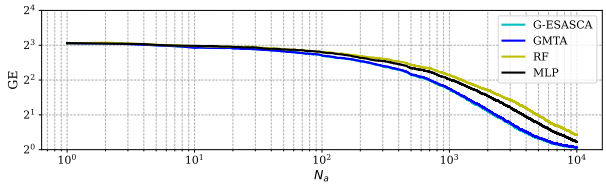


(d) $n = 10^6$.

Figure 4: GE according to the number of attack traces N_a, for $d = 2$ and SNR $= 1$ (flawless implementation).



(a) $n = 10^4$.

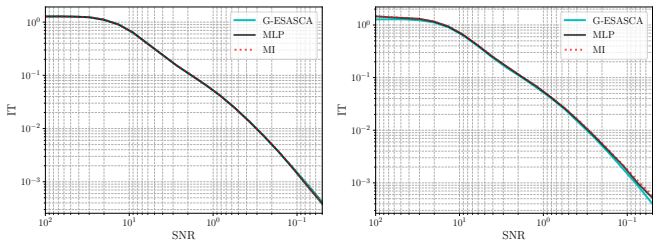


(b) $n = 10^6$.

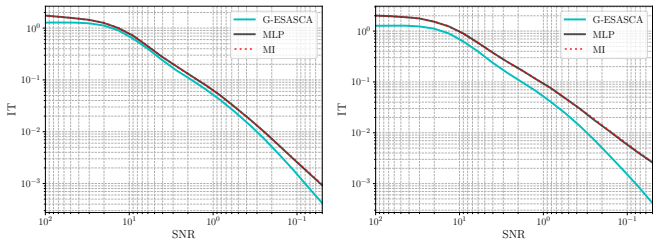
Figure 5: GE according to the number of attack traces N_a, for $d = 2$ and SNR $= 0.1$ (flawless implementation).

flaw in a masked implementation is not expected to affect our conclusions regarding the profiling complexity of the distinguishers (as confirmed in Appendix A, Figure 8).

We first illustrate the possibility of a security overstatement in Figure 6. It shows that while the order-preserving G-ESASCA and the MLP distinguishers both have an asymptotic PI equal to the MI independent of the SNR when there is no flaw in the masked implementation (in the left part of the figure), the presence of a flaw makes the PI of the G-ESASCA distinguisher significantly lower than the MI in the presence of a flaw (in the right part of the figure).



(a) $f = 0.0$ (b) $f = 0.0625$



(c) $f = 0.125$ (d) $f = 0.25$

Figure 6: Flawed 2-share implementation: asymptotic PI.

We then systematize this investigation in Figure 7, which shows how the security overstatement of the G-ESASCA distinguisher increases with the amplitude of the flaw (i.e., the f parameter), while the MLP distinguisher preserves an optimal PI value independent of the f parameter. As expected from Table 1, we observe a similar trend from MLP-ESASCA which is also a order-preserving distinguisher.

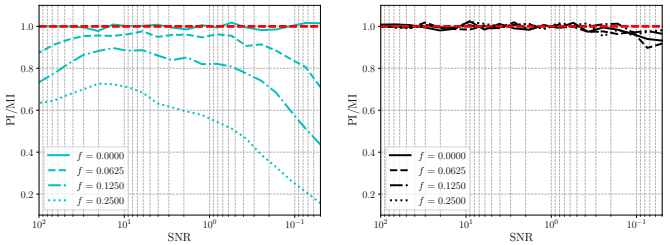
Overall, the results in this section confirm that while using the masking randomness in an evaluation is sound for the investigated case studies, the order-preserving assumption can lead to security overstatements. Whenever used in the evaluation of a leaking implementation, it is therefore important to confirm in parallel that it is sufficiently fulfilled (i.e., that the f parameter is low enough), for example using moment-based detections or attacks [21], [32] or, more formally, by relying on leakage certification [43].

Eventually, the above experiments were performed for a fixed set of meta-parameters. We refer to Appendix C for a discussion about their influence on our conclusions. We also provide an source code enabling to reproduce our results and to change these meta-parameters: https://github.com/uclcrypto/efficient_profiled_attacks_extended.

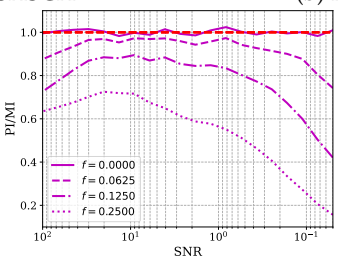
6 Conclusions

One of the long-standing open problems in the evaluation of cryptographic implementations against side-channel attacks

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2022.3144871, IEEE Transactions on Information Forensics and Security



(a) G-ESASCA (b) MLP



(c) MLP-ESASCA

Figure 7: Flawed 2-share implementation: asymptotic PI/MI.

is to determine what are the adversarial capabilities to consider. In the context of masked implementations, this question is typically reflected by a dilemma regarding the knowledge of the shares during profiling. On the one hand, knowing this randomness can significantly speed up the evaluations. On the other hand, and to the best of our knowledge, the question whether it leads to a gap between online attacks that can be performed in an evaluation context and more concrete attacks profiled without access to this randomness remains open (see [63] for a recent discussion). In this paper, we first contribute to this issue and show that for implementations with large enough noise (so that the masking countermeasure is effective), determined adversaries can reach the same (worst-case) attack complexities as evaluators, with only a penalty in profiling complexity.

We complement this main conclusion with a systematization effort and evaluate the impact of other solutions that may simplify the evaluation problem, by positing sound assumptions on the leakage distribution. We show that such assumptions are in general useful from the profiling complexity viewpoint but (as theoretically known), can lead to a false "sense of security" in case the actual leakages significantly deviate from the evaluator's assumptions. We therefore propose a classification of profiled distinguishers in function of the assumptions they make, which we hope can help evaluators selecting the appropriate statistical tools in function of the implementations to analyze.

Acknowledgments. François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS). This work has been funded in parts by the European Union and the Walloon Region through the ERC project SWORD (project 724725) and the FEDER project USERMedia (convention number 501907-379156).

REFERENCES

[1] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *CRYPTO*, ser. Lecture Notes in Computer Science, vol. 1666. Springer, 1999, pp. 388–397.

[2] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *CHES*, ser. Lecture Notes in Computer Science, vol. 2162, no. Generators. Springer, 2001, pp. 251–261.

[3] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[4] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *CRYPTO*, ser. Lecture Notes in Computer Science, vol. 1666. Springer, 1999, pp. 398–412.

[5] Y. Ishai, A. Sahai, and D. A. Wagner, "Private circuits: Securing hardware against probing attacks," in *CRYPTO*, ser. Lecture Notes in Computer Science, vol. 2729. Springer, 2003, pp. 463–481.

[6] F. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, and S. Mangard, "The world is not enough: Another look on second-order DPA," in *ASIACRYPT*, ser. Lecture Notes in Computer Science, vol. 6477. Springer, 2010, pp. 112–129.

[7] E. Prouff and M. Rivain, "Masking against side-channel attacks: A formal security proof," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, vol. 7881. Springer, 2013, pp. 142–159.

[8] A. Duc, S. Dziembowski, and S. Faust, "Unifying leakage models: From probing attacks to noisy leakage," *J. Cryptology*, vol. 32, no. 1, pp. 151–177, 2019.

[9] A. Duc, S. Faust, and F. Standaert, "Making masking security proofs concrete (or how to evaluate the security of any leaking device), extended version," *J. Cryptology*, vol. 32, no. 4, pp. 1263–1297, 2019.

[10] J. Coron, E. Prouff, M. Rivain, and T. Roche, "Higher-order side channel security and mask refreshing," in *FSE*, ser. Lecture Notes in Computer Science, vol. 8424. Springer, 2013, pp. 410–424.

[11] G. Barthe, S. Belaïd, F. Dupressoir, P. Fouque, B. Grégoire, and P. Strub, "Verified proofs of higher-order masking," in *EUROCRYPT (1)*, ser. Lecture Notes in Computer Science, vol. 9056. Springer, 2015, pp. 457–485.

[12] G. Barthe, S. Belaïd, F. Dupressoir, P. Fouque, B. Grégoire, P. Strub, and R. Zucchini, "Strong non-interference and type-directed higher-order masking," in *ACM Conference on Computer and Communications Security*. ACM, 2016, pp. 116–129.

[13] S. Nikova, V. Rijmen, and M. Schläffer, "Secure hardware implementation of nonlinear functions in the presence of glitches," *J. Cryptology*, vol. 24, no. 2, pp. 292–321, 2011.

[14] S. Faust, V. Grosso, S. M. D. Pozo, C. Paglialonga, and F. Standaert, "Composable masking schemes in the presence of physical defaults & the robust probing model," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 89–120, 2018.

[15] T. D. Cnudde, B. Bilgin, B. Gierlichs, V. Nikov, S. Nikova, and V. Rijmen, "Does coupling affect the security of masked implementations?" in *COSADE*, ser. Lecture Notes in Computer Science, vol. 10348. Springer, 2017, pp. 1–18.

[16] T. D. Cnudde, M. Ender, and A. Moradi, "Hardware masking, revisited," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 2, pp. 123–148, 2018.

[17] I. Levi, D. Bellizia, and F. Standaert, "Reducing a masked implementation's effective security order with setup manipulations and an explanation based on externally-amplified couplings," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 2, pp. 293–317, 2019.

[18] A. Battistello, J. Coron, E. Prouff, and R. Zeitoun, "Horizontal side-channel attacks and countermeasures on the ISW masking scheme," in *CHES*, ser. Lecture Notes in Computer Science, vol. 9813. Springer, 2016, pp. 23–39.

[19] V. Grosso and F. Standaert, "Masking proofs are tight and how to exploit it in security evaluations," in *EUROCRYPT (2)*, ser. Lecture Notes in Computer Science, vol. 10821. Springer, 2018, pp. 385–412.

[20] L. Mather, E. Oswald, J. Bandenburg, and M. Wójcik, "Does my device leak information? an a priori statistical power analysis of leakage detection tests," in *ASIACRYPT (1)*, ser. Lecture Notes in Computer Science, vol. 8269. Springer, 2013, pp. 486–505.

[21] T. Schneider and A. Moradi, "Leakage assessment methodology - extended version," *J. Cryptographic Engineering*, vol. 6, no. 2, pp. 85–99, 2016.

[22] F. Durvaux and F. Standaert, "From improved leakage detection to the detection of points of interests in leakage traces," in *EUROCRYPT (1)*, ser. Lecture Notes in Computer Science, vol. 9665. Springer, 2016, pp. 240–262.

[23] A. Moradi, B. Richter, T. Schneider, and F. Standaert, "Leakage detection with the x2-test," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 1, pp. 209–237, 2018.

[24] F. Standaert, "How (not) to use welch's t-test in side-channel security evaluations," in *CARDIS*, ser. Lecture Notes in Computer Science, vol. 11389. Springer, 2018, pp. 65–79.

[25] O. Bronchain, T. Schneider, and F. Standaert, "Multi-tuple leakage detection and the dependent signal issue," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 2, pp. 318–345, 2019.

[26] D. B. Roy, S. Bhasin, S. Guilley, A. Heuser, S. Patranabis, and D. Mukhopadhyay, "CC meets FIPS: A hybrid test methodology for first order side channel analysis," *IEEE Trans. Computers*, vol. 68, no. 3, pp. 347–361, 2019.

[27] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *CHES*, ser. Lecture Notes in Computer Science, vol. 2523. Springer, 2002, pp. 13–28.

[28] K. Lemke-Rust and C. Paar, "Gaussian mixture models for higher-order side channel analysis," in *CHES*, ser. Lecture Notes in Computer Science, vol. 4727. Springer, 2007, pp. 14–27.

[29] T. Schneider, A. Moradi, F. Standaert, and T. Güneysu, "Bridging the gap: Advanced tools for side-channel leakage estimation beyond gaussian templates and histograms," in *SAC*, ser. Lecture Notes in Computer Science, vol. 10532. Springer, 2016, pp. 58–78.

[30] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *CHES*, ser. Lecture Notes in Computer Science, vol. 3156. Springer, 2004, pp. 16–29.

[31] E. Prouff, M. Rivain, and R. Bevan, "Statistical analysis of second order differential power analysis," *IEEE Trans. Computers*, vol. 58, no. 6, pp. 799–811, 2009.

[32] A. Moradi and F. Standaert, "Moments-correlating DPA," in *TIS@CCS*. ACM, 2016, pp. 5–15.

[33] W. Schindler, K. Lemke, and C. Paar, "A stochastic model for differential side channel cryptanalysis," in *CHES*, ser. Lecture Notes in Computer Science, vol. 3659. Springer, 2005, pp. 30–46.

[34] K. Lemke-Rust and C. Paar, "Analyzing side channel leakage of masked implementations with stochastic methods," in *ESORICS*, ser. Lecture Notes in Computer Science, vol. 4734. Springer, 2007, pp. 454–468.

[35] G. Dabosville, J. Doget, and E. Prouff, "A new second-order side channel attack based on linear regression," *IEEE Trans. Computers*, vol. 62, no. 8, pp. 1629–1640, 2013.

[36] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis," in *CHES*, ser. Lecture Notes in Computer Science, vol. 5154. Springer, 2008, pp. 426–442.

[37] E. Prouff and M. Rivain, "Theoretical and practical aspects of mutual information based side channel analysis," in *ACNS*, ser. Lecture Notes in Computer Science, vol. 5536, 2009, pp. 499–518.

[38] B. Gierlichs, L. Batina, B. Preneel, and I. Verbauwhede, "Revisiting higher-order DPA attacks:," in *CT-RSA*, ser. Lecture Notes in Computer Science, vol. 5985. Springer, 2010, pp. 221–234.

[39] L. Batina, B. Gierlichs, E. Prouff, M. Rivain, F. Standaert, and N. Veyrat-Charvillon, "Mutual information analysis: a comprehensive study," *J. Cryptology*, vol. 24, no. 2, pp. 269–291, 2011.

[40] F. Durvaux, F. Standaert, and N. Veyrat-Charvillon, "How to certify the leakage of a chip?" in *EUROCRYPT*, ser. Lecture Notes in Computer Science, vol. 8441. Springer, 2014, pp. 459–476.

[41] A. Heuser, O. Rioul, and S. Guilley, "Good is not good enough - deriving optimal distinguishers from communication theory," in *CHES*, ser. Lecture Notes in Computer Science, vol. 8731. Springer, 2014, pp. 55–74.

[42] N. Bruneau, S. Guilley, A. Heuser, and O. Rioul, "Masks will fall off - higher-order optimal distinguishers," in *ASIACRYPT (2)*, ser. Lecture Notes in Computer Science, vol. 8874. Springer, 2014, pp. 344–365.

[43] O. Bronchain, J. M. Hendrickx, C. Massart, A. Olshevsky, and F. Standaert, "Leakage certification revisited: Bounding model errors in side-channel security evaluations," in *CRYPTO (1)*, ser. Lecture Notes in Computer Science, vol. 11692. Springer, 2019, pp. 713–737.

[44] C. Whitnall, E. Oswald, and F. Standaert, "The myth of generic dpa...and the magic of learning," in *CT-RSA*, ser. Lecture Notes in Computer Science, vol. 8366. Springer, 2014, pp. 183–205.

[45] L. Lerman and O. Markowitch, "Efficient profiled attacks on masking schemes," *IEEE Trans. Information Forensics and Security*, vol. 14, no. 6, pp. 1445–1454, 2019.

[46] F. Standaert, T. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, vol. 5479. Springer, 2009, pp. 443–461.

[47] F. Standaert, F. Koeune, and W. Schindler, "How to compare profiled side-channel attacks?" in *ACNS*, ser. Lecture Notes in Computer Science, vol. 5536, 2009, pp. 485–498.

[48] M. Renauld, F. Standaert, N. Veyrat-Charvillon, D. Kamel, and D. Flandre, "A formal study of power variability issues and side-channel attacks for nanoscale devices," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, vol. 6632. Springer, 2011, pp. 109–128.

[49] E. de Chérisey, S. Guilley, O. Rioul, and P. Piantanida, "Best information is most successful mutual information and success rate in side-channel analysis," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 2, pp. 49–79, 2019.

[50] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, 1996.

[51] N. Veyrat-Charvillon, B. Gérard, and F. Standaert, "Soft analytical side-channel attacks," in *ASIACRYPT (1)*, ser. Lecture Notes in Computer Science, vol. 8873. Springer, 2014, pp. 282–296.

[52] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *SPACE*, ser. Lecture Notes in Computer Science, vol. 10076. Springer, 2016, pp. 3–26.

[53] L. Masure, C. Dumas, and E. Prouff, "A comprehensive study of deep learning for side-channel analysis," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2020, no. 1, pp. 348–375, 2020. [Online]. Available: https://doi.org/10.13154/tches.v2020.i1.348-375

[54] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, ser. Adaptive computation and machine learning. MIT Press, 2006.

[55] A. Pinkus, "Approximation theory of the MLP model in neural networks," *Acta Numerica*, vol. 8, pp. 143–195, 1999.

[56] O. Bronchain and F.-X. Standaert, "Breaking masked implementations with many shares on 32-bit software platforms or when the security order does not matter," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 3, pp. XXX–XXX, 2021.

[57] M. Ouladj, S. Guilley, P. Guillot, and F. Mokrane, "Spectral approach to process the (multivariate) high-order template attack against any masking scheme," *Journal of Cryptographic Engineering*, 2021.

[58] C. J. Stone, "Optimal Rates of Convergence for Nonparametric Estimators," *The Annals of Statistics*, vol. 8, no. 6, pp. 1348 – 1360, 1980. [Online]. Available: https://doi.org/10.1214/aos/1176345206

[59] ——, "Optimal uniform rate of convergence for nonparametric estimators of a density function or its derivatives," in *Recent Advances in Statistics*, M. H. Rizvi, J. S. Rustagi, and D. Siegmund, Eds. Academic Press, 1983, pp. 393–406. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780125893206500228

[60] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014. [Online]. Available: http://www.cambridge.org/de/academic/subjects/computer-science/pattern-recognition-and-machine-learning/understanding-machine-learning-theory-algorithms

[61] E. Peeters, F. Standaert, N. Donckers, and J. Quisquater, "Improved higher-order side-channel attacks with FPGA experiments," in *CHES*, ser. Lecture Notes in Computer Science, vol. 3659. Springer, 2005, pp. 309–323.

[62] M. Azouaoui, D. Bellizia, I. Buhan, N. Debande, S. Duval, C. Giraud, É. Jaulmes, F. Koeune, E. Oswald, F. Standaert, and C. Whitnall, "A systematic appraisal of side channel evaluation strategies," in *SSR*, ser. Lecture Notes in Computer Science, vol. 12529. Springer, 2020, pp. 46–66.

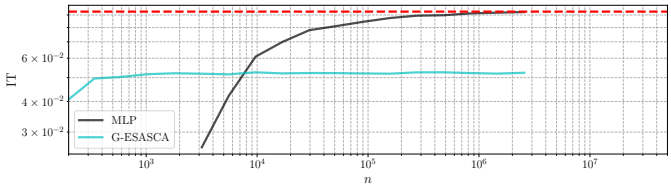
[63] O. Bronchain, G. Cassiers, and F. Standaert, "Give me 5 minutes: Attacking ascad with a single side-channel trace," *IACR Cryptology ePrint Archive*, vol. 2021, p. 817, 2021. [Online]. Available: https://eprint.iacr.org/2021/817

[64] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian, "Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks," *J. Mach. Learn. Res.*, vol. 20, pp. 63:1–63:17, 2019. [Online]. Available: http://jmlr.org/papers/v20/17-612.html

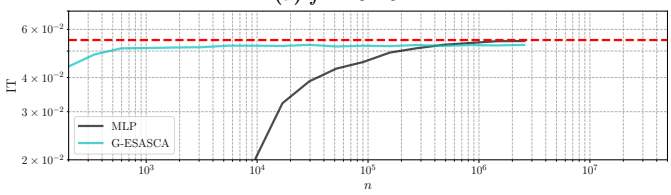
[65] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1,

pp. 5–32, 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

APPENDIX A
IT CURVES FOR FLAWED IMPLEMENTATIONS



(a) $f = 0.25$



(b) $f = 0.0625$

Figure 8: Flawed 2-share implem.: profiling with SNR $= 1$.

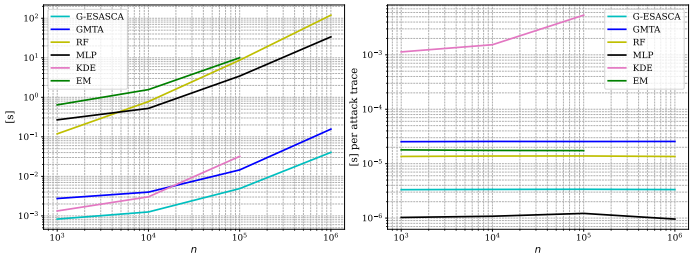
APPENDIX B
DISTINGUISHER' RUNTIMES

Figure 9 contains the running times for profiling and attack (to process a single trace) according to the number of traces available during profiling. Generally, we observe that additional hypotheses and randomness knowledges leads to more efficient profiling from the computational viewpoint as well. In particular, GMTA and G-ESASCA are the most efficient for profiling. During the attack phase, only KDE has a runtime increasing with the number of profiling traces making its interest limited for protected targets (i.e., with large n values) . We stress that the analyses presented in this paper are primarily focused on the data complexity. These time complexity values are given for completeness and could be further optimized. Their main goal is to show that the one-time profiling effort of practically-relevant distinguishers is not unrealistic computationally.

APPENDIX C
INFLUENCE OF META-PARAMETERS.

The results presented in this work are done for a given setting of hyper-parameters (see the description of each estimator in subsection 2.4). One may wonder whether the results presented may evolve depending on the choice of meta-parameters in some models, *e.g.*, MLP or RF. We argue hereafter that the trends depicted in this work remain mostly unchanged when modifying the meta-parameters. Like comparing two different models, comparing two meta-parameters may be discussed in terms of profiling complexity or online attack complexity (see subsection 3.2).

MLP. For MLP, we chose a light architecture with one hidden layer of $1,000$ neurons, and a *Rectified Linear Unit* (ReLU) activation function. The choice of the activation function affects the approximation error, as long as it is



(a) Profiling. (b) Normalized attack.

Figure 9: Time complexity of different distinguishers when profiling and attacking. The profiling values include the full profiling process and the attacking values are normalized by the number of exploited traces. The MLP distinguisher is running on a GPU. Other distinguishers are implemented with sklearn and numpy running on a single CPU.

not a polynomial [55]. This means that the online attack complexity is not sensitive to the choice of the activation function. Likewise, Masure *et al.* empirically verified that the approximation error is negligible in a similar experimental setting with higher-order masking (up to $d = 4$). This not only suggests that increasing the number of neurons shall not increase the online attack complexity, but also that decreasing the number of neurons in the case where $d < 4$ should not affect much the online attack complexity.

Regarding the profiling complexity, it is known that the convergence rate of MLP is roughly proportional to the number of parameters to fit [64]. Since even when using our light architecture, we got the slowest convergence in Figure 2 and Figure 3, repeating the same comparisons, up to a more complex architecture should not make the convergence faster. Hence, the trends observed in Figures 2 and 3 should remain essentially unchanged.

RF. For RF, increasing the number of trees may improve the online attack complexity (up to a computational overhead) [65]. Nevertheless, as suggested by Figure 5b, the asymptotic performance of RF relatively to the other estimators suggests that the number of decision trees is high enough for our experiments. Moreover, the number of decision trees should not impact much the profiling complexity – beyond the asymptotic performance –, as the amount of data used to design each decision tree does not depend on the number of trees [60]. Accordingly, decreasing the number of decision trees is not expected to significantly improve the performance. As explained in subsubsection 2.4.6, more decision trees in the random forest may however be needed for a higher noise level, or a higher bit size.