Masking Proofs Are Tight and How to Exploit it in Security Evaluations

Vincent Grosso[1(✉)] and François-Xavier Standaert[2]

[1] Digital Security Group, Radboud University Nijmegen, Nijmegen, The Netherlands
v.grosso@cs.ru.nl

[2] ICTEAM - Crypto Group, Université catholique de Louvain, Louvain-la-Neuve, Belgium

Abstract. Evaluating the security level of a leaking implementation against side-channel attacks is a challenging task. This is especially true when countermeasures such as masking are implemented since in this case: (*i*) the amount of measurements to perform a key recovery may become prohibitive for certification laboratories, and (*ii*) applying optimal (multivariate) attacks may be computationally intensive and technically challenging. In this paper, we show that by taking advantage of the tightness of masking security proofs, we can significantly simplify this evaluation task in a very general manner. More precisely, we show that the evaluation of a masked implementation can essentially be reduced to the one of an unprotected implementation. In addition, we show that despite optimal attacks against masking schemes are computationally intensive for large number of shares, heuristic (soft analytical side-channel) attacks can approach optimality efficiently. As part of this second contribution, we also improve over the recent multivariate (aka horizontal) side-channel attacks proposed at CHES 2016 by Battistello et al.

1 Introduction

Say you design a new block cipher and want to argue about its resistance against linear cryptanalysis [44]. One naive approach for this purpose would be to launch many experimental attacks. Yet, such a naive approach rapidly turns out to be unsuccessful if the goal is to argue about security levels beyond the computational power of the designer (e.g., 80-bit or 128-bit security for current standards). Hence, symmetric cryptographers have developed a variety of tools allowing them to bound the security of a block cipher against linear cryptanalysis, under sound and well-defined assumptions. As a typical example of these tools, one can cite the wide-trail strategy that has been used in the design of the AES Rijndael [17]. Its main idea is to minimize the bias (i.e., the informativeness) of the best linear characteristics within the cipher, which can be estimated under some independence assumptions thanks to the piling-up lemma.

Interestingly, the last years have shown a similar trend in the field of side-channel security evaluations. That is, while certification practices are still heavily dominated by "attack-based evaluations", solutions have emerged in order

J. B. Nielsen and V. Rijmen (Eds.): EUROCRYPT 2018, LNCS 10821, pp. 385–412, 2018.
https://doi.org/10.1007/978-3-319-78375-8_13

to both extend the guarantees and reduce the cost of these evaluations. More precisely, current certification practices focus either on the automatic verification of some minimum (non-quantitative) properties based on so-called leakage detection tools (e.g., [13,22,30,42,55]), or on the exhibition of concrete attack paths exploiting the detected leakages (typically taking advantages of standard distinguishers such as [9,11,28,54]). But they are anyway unable to claim security levels beyond the measurement efforts of the evaluation laboratory. In order to mitigate this limitation, one first intuitive line of papers proposed tools allowing to easily predict the success rate of some specialized distinguishers, based on parameters such as the noise level of the implementation [18,27,38,51]. In parallel, and following a more standard cryptographic approach trying to be independent of the adversarial strategy, significant progresses have been made in the mathematical treatment of physical security. In particular the masking countermeasure, which is one of the most common methods to improve the security of leaking cryptographic implementations, has been analyzed in several more or less formal models [10,19,34,50,58]. These works suggest that physical security via masking has strong analogies with the case of linear cryptanalysis. Namely, security against linear cryptanalysis is obtained by ensuring that the XOR of many (local) linear approximations has low bias. Similarly, masking ensures that every sensitive variable within an implementation is split (e.g., XORed) into several shares that the adversary has to recombine. So intuitively, masking security proofs can be viewed as a noisy version of the piling-up lemma.

Following these advances, the integration of masking proofs as a part of concrete security evaluation practices, undertaken in [20], appears as a necessary next step. And this is especially true when envisioning future cryptographic implementations with high (e.g., >80-bit) security levels, for which an attack-based certification process is unlikely to bring any meaningful conclusion. So the main objective of this paper is to follow such an approach and to show how masking security proofs can be used to gradually simplify side-channel security evaluations, at the cost of some conservative assumptions, but also some more critical ones (e.g., related to the independence of the shares' leakages).

More precisely, we start from the observation that a so far under-discussed issue in physical security evaluations is the case of attacks taking advantage of multiple leaking intermediate variables (e.g., see [33,43,59] for recent references).[1] As put forward in a recent CHES 2016 paper, this issue gains relevance in the context of masked implementations, in view of the (quadratic) cost overheads those implementations generally imply [7]. In this respect, our first contribution is to extend the analysis of masking security proofs from [20] and to show that these proofs remain essentially tight also for multi-target attacks.

Next, and since we aim to discuss the cost of side-channel security evaluations, we propose a simple metric for the evaluation complexity, and use it to extensively discuss the tradeoff between the time needed for a (worst-case) security evaluation and the risks related to the (e.g., independence) assumptions

[1] Which is an orthogonal concern to the more studied one of exploiting multiple leakage samples per intermediate variable (e.g., see [1] and follow up works).

it exploits. As our investigations suggest that the time complexity of optimal side-channel attacks can become a bottleneck when the security levels of masked implementations increase, we also study efficient (heuristic) multi-target attacks against masked implementations. Our best attack significantly improves the multivariate (aka horizontal) iterative attack proposed at CHES 2016 by Battistello et al., that we re-frame as a Soft Analytical Side-Channel Attack [33,59]. Note that our results provide a complementary view to those of Battistello et al., since they typically fix the masking noise parameter and look for the number of masking shares such that their attack is feasible, while we rather fix the number of shares and estimate the resulting security level in function of the noise.

Eventually, we show that the security evaluation of a leaking implementation against worst-case attacks taking advantage of all the target intermediate variables that can be enumerated by an adversary (so still limited to the first/last cipher rounds) boils down to the information theoretic analysis of a couple of its samples, for which good tools exist to guarantee a sound treatment [23,24]. By combining information theoretic evaluations with metric-based bounds for the complexity of key enumeration [48], we can obtain security graphs for optimal attacks, plotting the success rate in function of the measurement and time complexity, within seconds of computation on a computer. We argue that such tools become increasingly necessary for emerging high-security implementations.

2 Cautionary Remarks

Admittedly, the more efficient evaluations we discuss next are based on a number of simplifying assumptions. In this respect, we first recall that secure masking depends on two conditions: sufficient noise and independent leakages. This paper is about the first condition only. That is, we assume that the independence condition is fulfilled (to a sufficient extent), and study how exploiting all the leakage samples in an implementation allows reducing its noise.

We note that tools to ensure (or at least test empirically) the independence condition are already widely discussed in the literature. Concretely, there are two main issues that can break this assumption. First, imperfect refreshing schemes can cause d'-tuples of leakage samples to be key-dependent with d' lower than the number of shares used in the masking scheme d. For example, such an issue was put forward in [16]. It can be provably avoided by using "composable" (e.g., SNI [4]) gadgets or testing the security of the masking description code (i.e., the instructions defining an algorithm) thanks to formal methods [3].

Second, and more critically, different case studies have shown that actual leakage functions can break the independence assumption and recombine (a part of) the shares, e.g., because of transitions in software implementations [14] or glitches in hardware implementations [40]. Nevertheless, in practice such (partial) recombinations typically reduce the (statistical) "security order" of the implementations, captured by the lowest statistical moment of the leakage distribution that is key-dependent (minus one) [5], to some value d'' below the optimal $(d - 1)$, while leaving security margins (i.e., $d'' > 1$). As a result, by increasing the number of

shares d, one can generally mitigate these physical defaults to a good extent [2,46]. Furthermore, simple leakage detection tools such as [13,22,30,42,55] can be used to (empirically) assess the security order of an implementation, and these non-independence issues can be reflected in information theoretic evaluations (see [20], Sect. 4.2). So overall, ensuring the independence of the shares' leakages in a masked implementation is an orthogonal concern to ours. While non-independence issues may indeed increase the information leakage of the tuples of samples exploited in an high-order side-channel attack, it does not affect the importance/relevance of taking all the exploitable tuples into account in a (worst-case) security evaluation, which is our main concern.

Eventually, we insist that this work is prospective in the sense that our typical targets are masked implementations with (very) large number of shares, aimed at (very) high security levels (e.g., no key recovery with less than 2^{40} measurements). In this respect, we refer to two recently accepted papers (to Eurocrypt 2017) as an excellent motivation for our purposes [5,31]. In particular, [31] describes AES implementations masked with 5 to 10 shares, for which the security evaluation was left as an open problem by the authors and that are typical targets for which attack-based evaluations are unlikely to bring meaningful conclusions. Our following discussions describe theoretical tools allowing one to state sound security claims for such implementations. The important message they carry is that even when the independent shares' leakage assumption is guaranteed, one also needs to pay attention to noise. Simple univariate tests are not enough for this purpose. Performing highly multivariate attacks is (very) expensive. We introduce an intermediate path that allows principled reasoning and to assess the risks of overstated security based on well identified parameters. Quite naturally, this intermediate path also comes with limitations. Namely, since we focus on (very) high security levels, the bounds we provide are also less accurate, and reported as log-scaled plots for convenience (i.e., we typically ignore the impact of small constants as a first step). We will conclude the paper by referring to a recent CHES 2017 work that demonstrated to applicability of our tools based on a 32-share masked AES implementation in an ARM Cortex M4 [35].

3 Background

3.1 S-box Implementations

Our investigations consider both the unprotected and the masked implementation of an m-bit S-box S taking place in the first round of a block cipher.

For the unprotected case, we denote the input plaintext with x and the secret key with k. We define $y_a = x \oplus k$ as the result of a key addition between x and k, and $y_b = \mathsf{S}(y_a)$ as the S-box output. The vector of the target intermediate variables is further denoted with $\boldsymbol{y} = [y_a, y_b]$ and the leakage vector corresponding to these variables with $\boldsymbol{L} = [L_a, L_b] + \boldsymbol{N}$, where $\boldsymbol{N}$ is a bivariate random variable representing an additive Gaussian noise. We make the usual assumption that the noise covariance matrix is diagonal and each sample L_i has a similar noise

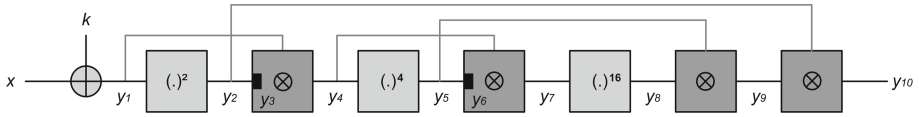

Fig. 1. Multiplication chain for the inversion in GF(2^8) from [52].

variance σ_n^2.[2] Eventually, the deterministic part of the leakage samples are the output of a leakage function L such that $L_i = \mathsf{L}_i(y_i)$, $\forall i \in \{a, b\}$. For simplicity, our experiments will consider L_i to be the Hamming weight function $\forall i$'s. We detail in Sect. 3.2 why this choice does not affect our conclusions.

For the masked case, we will focus on the secure inversion in GF(2^8) proposed in [52], which is the core of the AES S-box and illustrated in Fig. 1. More precisely, we use a slightly modified version of the algorithms of [52], with a secure refreshing R (e.g., from [4,34]) represented by black rectangles on the figure, in order to avoid the attack put forward in [16].[3] Next, we define the notations $y_1 = y_a = x \oplus k$, $y_2 = (y_1)^2$, $y_3 = \mathsf{R}(y_2)$, $y_4 = y_1 \otimes y_3 = (y_1)^3$, $y_5 = (y_4)^4 = (y_1)^{12}$, $y_6 = \mathsf{R}(y_5)$, $y_7 = y_4 \otimes y_6 = (y_1)^{15}$, $y_8 = (y_7)^{16} = (y_1)^{240}$, $y_9 = y_5 \otimes y_8 = (y_1)^{252}$, $y_{10} = y_2 \otimes y_9 = (y_1)^{254}$, with $\otimes$ the field multiplication. This leads to a vector of target intermediate variables $\boldsymbol{y} = [y_1, y_2, \ldots, y_{10}]$. For an implementation masked with d shares, we additionally have a vector of shares $\bar{\boldsymbol{y}} = [\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{10}]$ such that $\bar{y}_i = [y_i(1), y_i(2), \ldots, y_i(d)]$ $\forall i \in \{1, 2, \ldots, 10\}$. It leads to a leakage vector $\bar{\boldsymbol{L}} = [\bar{L}_1, \bar{L}_2, \ldots, \bar{L}_{10}] + \boldsymbol{N}$, where each leakage d-tuple is denoted as $\bar{L}_i = [L_i(1), L_i(2), \ldots, L_i(d)]$ and made of d samples, the multivariate noise variable is defined as in the unprotected case (with more dimensions) and $L_i(j) = \mathsf{L}_{i,j}(y_i(j))$ $\forall i \in \{1, 2, \ldots, 10\}, j \in \{1, 2, \ldots, d\}$. Such a masking scheme has security order $(d-1)$, meaning that any $(d-1)$-tuple of leakage samples is independent of k, *given that each leakage sample depends on a single share*. We call this assumption the Independent Shares' Leakage (ISL) assumption.

Concretely, the multiplication chain of Fig. 1 is made of squarings, that are GF(2)-linear, and multiplications. In order to evaluate them securely, we use Algorithms 1 and 2 given in Appendix A. For the squarings, the operations are applied to each share independently and therefore can be tabulized. For the multiplications, the different shares need to interact and the algorithm has quadratic overheads that correspond to the computation of all the partial products and their refreshing. For example, for $x = x(1) \oplus \cdots \oplus x(d)$ and $y = y(1) \oplus \cdots \oplus y(d)$, producing the shares of $x \otimes y$ requires to compute (for $d = 3$):

$$\begin{pmatrix} x(1) \otimes y(1) & x(1) \otimes y(2) & x(1) \otimes y(3) \\ x(2) \otimes y(1) & x(2) \otimes y(2) & x(2) \otimes y(3) \\ x(3) \otimes y(1) & x(3) \otimes y(2) & x(3) \otimes y(3) \end{pmatrix} \oplus \begin{pmatrix} 0 & r_{1,2} & r_{1,3} \\ -r_{1,2} & 0 & r_{2,3} \\ -r_{1,3} & -r_{2,3} & 0 \end{pmatrix}. \quad (1)$$

[2] The impact of this noise assumption is specifically discussed in Sect. 5.3.

[3] Note that more efficient solutions for this secure inversion exist, such as [32]. We kept the chain of Rivain and Prouff because for its simpler description.

This directly implies that whenever such a multiplication is targeted by the adversary, we need to add d leakage d-tuples to the leakage vector $\bar{\boldsymbol{L}}$ he is provided with, that we next denote as $[\bar{L}_i^1, \bar{L}_i^2, \ldots, \bar{L}_i^d]$, with $i \in \{4, 7, 9, 10\}$.

Eventually, the GF(2^8) field multiplication is usually implemented using log/alog tables, as described in Appendix A, Algorithm 3. In case the adversary additionally targets these operations, another set of d leakage d-tuples must be added to $\bar{\boldsymbol{L}}$, next denoted as $[\bar{L}_i^{d+1}, \bar{L}_i^{d+2}, \ldots, \bar{L}_i^{2d}]$, with $i \in \{4, 7, 9, 10\}$.

In the following, we will consider different (more or less powerful) attack cases:

C1. The adversary targets only a single d-tuple (e.g., the S-box output one).
C2. The adversary exploits the ten d-tuples of the multiplication chain.
C3. The adversary additionally exploits the leakage of the four secure multiplications (i.e., Algorithm 2), leading to a total of 10 d-tuples and 4 d^2-tuples.
C4. The adversary additionally exploits the leakage of the field multiplications (i.e., Algorithm 3), leading to a total of 10 d-tuples and 8 d^2-tuples.

Furthermore, since a number of these d-tuples contain fresh randomness (e.g., the ones corresponding to multiplications algorithms) while other ones are deterministically related to each other, we will denote with $\delta = \lambda + \ell$ the number of d-tuples exploited, such that we have λ fresh ones and ℓ deterministic ones.

Note that our notations describe serial implementations where the adversary can observe the noisy leakage of each share in his d-tuples separately. This is a relevant choice since serial implementations are typically very expensive to analyze due to their large number of dimensions/leakage samples to consider. Yet, as recently discussed in [5], side-channel security for a serial implementation generally implies side-channel security for its parallel counterpart (as long as the ISL assumption remains fulfilled). So our conclusions apply in this case too.

3.2 Mutual Information Metric

In order to evaluate the worst-case security level of our different (unprotected and masked) simulated implementations, we will use the mutual information metric first put forward in [57]. The motivation of this choice is twofold. First, it was shown recently that this metric can be linked to the measurement complexity of the corresponding (worst-case) Bayesian adversary [20]. Second, it is significantly faster to estimate than the success rate, which is specially important/relevant in our context where we aim to minimize the evaluator's workload. We illustrate this fact with a simple example. Say an evaluator has 1000,000 measurements to estimate the security of an implementation with a worst-case Bayesian attack that is roughly successful after the collection of 1000 traces. In this case, it means that he can repeat 1000 independent experiments to estimate the success rate with 1000 traces (with good confidence). But say now that the implementation to evaluate can only be broken after (roughly) 1000,000 traces. Then it means that from his set of traces, the evaluator can only estimate the success rate based on a single experiment (which will not lead to any statistical confidence). By contrast, as discussed in [24], cross-validation allows him to exploit most of

his 1000,000 evaluation traces to estimate the mutual information metric, which will then be correlated with the success rate (for any number of traces).[4]

Concretely, computing the mutual information for an unprotected implementation simply requires to estimate the following sum of log probabilities:

$$\mathrm{MI}(K; X, \boldsymbol{L}) = \mathrm{H}[K] + \sum_{k \in \mathcal{K}} \Pr[k] \cdot \sum_{x \in \mathcal{X}} \Pr[x] \cdot \underbrace{\sum_{\boldsymbol{l} \in \mathcal{L}^{\delta}} \Pr[\boldsymbol{l}|k, x] \cdot \log_2 \Pr[k|x, \boldsymbol{l}]}_{\delta-\text{dimension integral}}, \quad (2)$$

where the conditional probability $\Pr[k|x, \boldsymbol{l}]$ is computed from the Probability Density Function (PDF) $\mathsf{f}[\boldsymbol{l}|x, k]$ thanks to Bayes' theorem as: $\frac{\mathsf{f}[\boldsymbol{l}|x,k]}{\sum_{k^*} \mathsf{f}[\boldsymbol{l}|x,k^*]}$. This corresponds to performing δ-dimensional integrals over the leakage samples, for each combination of the key k and plaintext x, or each bitwise XOR between k and x if taking advantage of the Equivalence under Independent Subkeys (EIS) assumption formalized in [54]. There are numerous publications where this metric has been computed, via numerical integrals or sampling (e.g., [20] provides an open source code for it), so we do not detail its derivation further.

When moving to masked implementations, the computation of the metric remains essentially similar. The only difference is that we need to sum over the randomness vector $\bar{\boldsymbol{y}}$ (which may become computationally intensive as the number of shares increases, as discussed in the next sections):

$$\mathrm{MI}(K; X, \bar{\boldsymbol{L}}) = \mathrm{H}[K] + \sum_{k \in \mathcal{K}} \Pr[k] \cdot \sum_{x \in \mathcal{X}} \Pr[x] \cdot \sum_{\bar{\boldsymbol{y}} \in \mathcal{Y}^{(d-1)\cdot\lambda}} \Pr[\bar{\boldsymbol{y}}] \cdot \underbrace{\sum_{\bar{\boldsymbol{l}} \in \mathcal{L}^{d\cdot\delta}} \Pr[\bar{\boldsymbol{l}}|k, x, \bar{\boldsymbol{y}}] \cdot \log_2 \Pr[k|x, \bar{\boldsymbol{l}}]}_{\delta-\text{dimension integral}}. \quad (3)$$

The computation of the conditional probability $\Pr[k|x, \boldsymbol{l}]$ follows similar guidelines as in the unprotected case, where the PDF of masked implementations becomes a mixture that can be written as $\mathsf{f}[\boldsymbol{l}|x, k] = \sum_{\bar{\boldsymbol{y}}} \mathsf{f}[\boldsymbol{l}|x, k, \bar{\boldsymbol{y}}]$ [36,58].

Remark. In our experiments where the (simulated) noise is Gaussian, we use a Gaussian PDF in the unprotected case, and a Gaussian mixture PDF in the masked case. Since we know the PDF exactly in these cases, we can compute the MI metric exactly and perform worst-case security evaluations. However, we insist that our discussions relate to the *complexity* of side-channel security evaluations, not their *optimality*. More precisely, our goal is to show that we can significantly simplify the evaluation of a highly protected implementation. These efficiency gains and our methodological conclusions are independent of the

[4] Note that the mutual information metric is not the only one allowing to simplify the estimation of a security level for a leaking cryptographic implementation. However, it is the most generic one since it does not require assumptions on the leakage distribution, nor on the choice of concrete distinguisher chosen by the adversary. More specialized (and sometimes more efficient) solutions include [18,27,38,51].

leakage function and model used by a concrete adversary (which however impacts the numerical results obtained). The main difference, if a concrete adversarial model was used in place of the perfect one, is that the log probabilities in Eqs. 2 and 3 would be evaluated based on it. This implies that less information would be extracted in case of model estimation or assumption errors, which is again an orthogonal concern to ours. Leakage certification could then be used to test whether estimation and assumption errors are small enough [23,24].

4 Unprotected Implementations

Evaluation complexity. Since our goal is to make side-channel security evaluations more efficient, a first question is to specify how we will evaluate complexity. Eventually we are interested in the measurement complexity of the attacks, which masking is expected to increase exponentially (in the number of shares). But of course, we also want to be able to evaluate the security of implementations of which the security is beyond what we can actually measure as evaluators. As just mentioned, computing the mutual information metric is an interesting tool for this purpose. Yet, it means that we still have to compute Eqs. 2 and 3, which are essentially made of a sum of δ-dimension integrals. Concretely, the (time) complexity for computing such a sum is highly dependent on the choice of PDF estimation tool chosen by the adversary/evaluator. In our case where we focus on attacks based on the exhaustive estimation of a mixture model, the number of integrals to perform is a natural candidate for the complexity of a (worst-case) side-channel evaluation, which we will next denote with $\mathsf{E}_{\$}$.[5]

In the case of an unprotected S-box implementation in GF(2^m), this leads to $\mathsf{E}_{\$} = 2^{2m}$ in general (since we sum over 2^m key bytes and 2^m plaintext bytes). This complexity is reduced to $\mathsf{E}_{\$} = 2^m$ if we take advantage of the EIS assumption. Since the latter assumption is generally correct in the "standard DPA" attack context we consider in this paper [39], we will always consider the complexity of evaluations taking advantage of EIS in the following (ignoring this simplification implies an additional 2^m factor in the evaluation complexities).

Practical evaluation results. As suggested by the previous formula, evaluating the security of an unprotected (8-bit) S-box is cheap. We now report on some exemplary results which we use to introduce an important assumption regarding our following simplifications. We consider different attack cases:

[5] Note that the only message this metric supports is that the evaluation complexity of an optimal side-channel attack can be reduced from unrealistic to easy by exploiting various assumptions. The integral count provides an intuitive solution for this purpose, but other metrics could be considered equivalently. Note also that heuristic attacks may approach worst-case ones more efficiently (we address this issue in Sect. 5.4). So we mostly use this metric to motivate the need of new tools for evaluating and attacking masked cryptographic implementations: for evaluations, it justifies why shortcut approaches are useful; for attacks, it justifies why heuristic appraoches such as outlined in Sect. 5.4 become necessary for large d's.

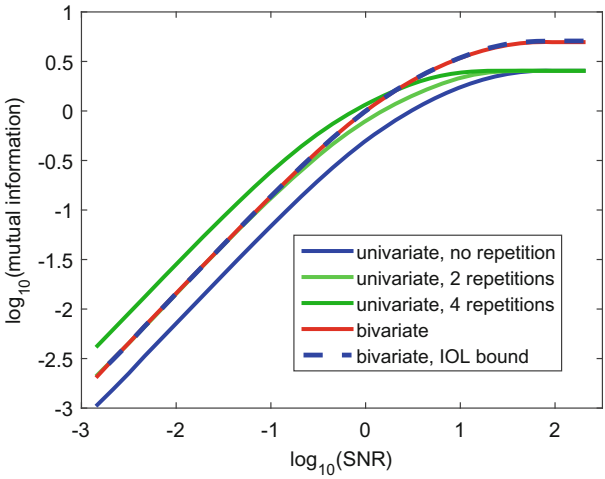

Fig. 2. Unprotected AES S-box evaluation results.

- Univariate, no repetition: the adversary observes the S-box output leakage.
- Univariate, with repetitions: the adversary observes the S-box output leakage several times with independent noise samples (e.g., 2 times, 4 times).
- Bivariate: the adversary observes the S-box input and output leakage.

Additionally, we consider a "bivariate attack bound" which is just the sum of two "univariate, no repetition" curves. In order to allow an easier interpretation of the results, we use the Signal-to-Noise Ratio (SNR) as X axis, defined as the variance of the noise-free traces (i.e., $m/4$ for Hamming weight leakages) divided by the variance of the noise. It better reflects the fact that the impact of the noise depends on the scaling of the signal. The results of these information evaluations are given in Fig. 2 from which two main conclusions can be extracted.

First, there is a difference between the impact of repeated observations, which just reduce the noise and therefore translate the information curves on the right, and bivariate attacks which (may) add information and shift these curves vertically. Interestingly, the latter observation is dependent on the S-boxes [49]: an identity S-box would lead to a repetition without information gain; a truly random one would lead to independent information for the two observations.

Second, the bivariate attack bound is tight in this case. This suggests that the AES S-box leads to quite independent information for the leakage samples L_a and L_b of our case study, which is consistent with the conclusions in [49]. Formally, we will say that this bound is tight if the Independent Operations' Leakages (IOL) assumption holds, which considers that the inputs/outputs of an operation (i.e., the AES S-box in our case study) are independent.

Note that as for the ISL assumption, the latter does not mean that the noise of the leakage samples has to be independent (which will be discussed in Sect. 5.2). Note also that the impact of a deviation from this IOL assumption is

very different than a deviation from the ISL assumption. Namely, if the share's leakages are not independent, then the formal security guarantees of masking vanish. By contrast, if the operation leakages are not independent, this will lead to less information and therefore less effective attacks. So the IOL assumption is not critical for the conclusion of a security evaluation, and overstating IOL may only lead to less tight (i.e., more conservative) security bounds.

5 Masked Implementations

We now move to the context of masked implementations which is the main contribution of this paper. We start by arguing that an exhaustive security evaluation is rapidly unreachable as the number of shares in masking increases. We then gradually simplify the evaluations, first without critical assumptions on the leakage distributions, second by exploiting the ISL assumption.

5.1 Exhaustive Approach

By visual inspection of Eq. 3, we directly find that the evaluation complexity $\mathsf{E}_{\$} = 2^{dm\lambda} + \ell \cdot 2^{dm}$, where we recall that λ is the number of fresh dimensions and ℓ the number of deterministic ones. For the case C1 in Sect. 3.1 with $d = 2$ shares, where the adversary targets only one 2-tuple of leakage samples corresponding to the masked S-box output y_8 in Fig. 1, this means a reachable 2^{2m} integrals. But as soon as we move to a (slightly) more powerful adversary, the complexity explodes. For example, the adversary of case C2 (who is still not optimal) with $m = 8$, $d = 2$, $\lambda = 6$ (due to the fresh intermediate values in Fig. 1) and $\ell = 4$ (due to the key addition and squarings), already leads to $\mathsf{E}_{\$} = 2^{96}$ integrals which is by far too expensive for evaluation laboratories.

5.2 Reducing Dimensionality with the IOL Assumption

The first cause in the complexity explosion of the exhaustive approach is the number of fresh dimensions. In this respect, a natural simplification is to exploit the IOL assumption. Indeed, by considering the operations in the multiplication chain of Fig. 1 as independent, the evaluation complexity of the previous (C2) adversary can be reduced to $\mathsf{E}_{\$} = \delta \cdot (2^{dm}) = 10 \cdot 2^{16}$ integrals. This is an interesting simplification since it corresponds to the strategy of an adversary willing to perform a multivariate attack against such a leaking masked implementation. Namely, he will identify the ten d-tuples of interest and combine their results via a maximum likelihood approach. We report the result of an information theoretic evaluation of this C2 adversary in Fig. 3, where we also plot the IOL bound provided by multiplying the information theoretic curve of the C1 adversary by ten. As for the case of unprotected implementations, the bound is tight.

Nevertheless, this simplification also implies two important technical questions. First, and since we assume the leakage of independent operations to be independent, what would be the impact of a dependent noise? Second, how to generalize this simplification to the adversaries C3 and C4 which imply the need of considering d^2-tuples jointly (rather than d-tuples jointly in the C2 case)?

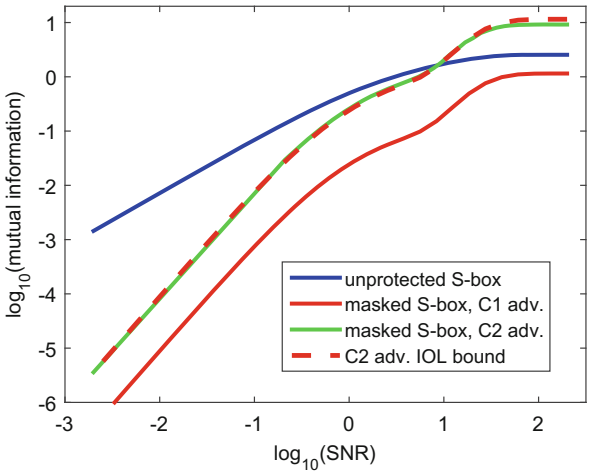

Fig. 3. Masked AES S-box evaluation results: cases C1 & C2 ($d = 2$).

5.3 The Dependent Noise Issue

To the best of our knowledge, this noise dependency issue has not been specifically discussed in the literature on masking, although the existence of correlated noise has been put forward in other contexts (e.g., see the discussion in [12], Chap. 6). We therefore launched an information theoretic evaluation of our masked S-box (case C1) with $d = 2$ and the covariance matrix such that the correlation between the noise samples of the two shares equals 0, 0.25, 0.5 and 0.75. The results of these evaluations are in Fig. 4. As expected, a correlated noise does not impact the security order of the countermeasure, defined as the lowest key-dependent moment in the leakage distribution $\Pr[k|x, \bar{\boldsymbol{l}}]$ minus one, and reflected by the slope of the information theoretic curves in the high-noise region (i.e., where the curves are linear) minus one. By contrast, correlated noise implies a shift of the curves by a factor that can be significant (e.g., $\times 2$ for correlation 0.5 and $\times 8$ for correlation 0.75). Such large correlations typically vanish after a couple of clock cycles. Yet, our results highlight that estimating the non-diagonal elements of the noise covariance matrices in masked implementations is an important sanity check that could be part of a certification process.

5.4 Secure Multiplication Leakages

When also considering the leakages of the d^2 cross products involved in a secure multiplication (such as the ones of Eq. 1 in Sect. 3.1 for $d = 3$), an additional problem is that computing an integral of d^2 dimensions rapidly becomes computationally intensive. This is particularly true if one considers an optimal Gaussian mixture model for the PDF since in this case the computation of the integral requires summing over the randomness vector. In fact, already for small field

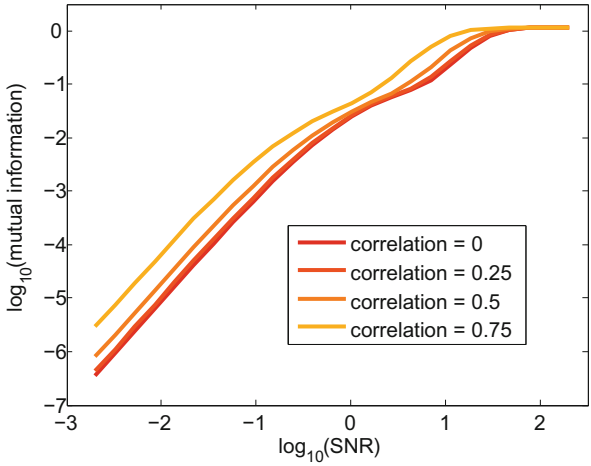

Fig. 4. Masked AES S-box evaluation results: case C1 with correlated noise ($d = 2$).

sizes and number of shares, the problem is hard. For example, for $d = 3$ and $m = 8$, the multiplication between two dependent values such as required in the multiplication chain of Fig. 1 requires performing 2^{24} integrals (corresponding to 8 bits of secret and twice 8 bits of randomness) of a 9-dimensional PDF.

In order to deal with this limitation, a solution is to look at masking proofs. In particular, Theorem 3 in [50] and Theorem 2 in [19] both provide bounds on the amount of information leaked by the multiplication of two secrets shared with Boolean masking, roughly corresponding to $(1.72d + 2.72)$ and $(28d + 16)$ times the information leakage of a single d-tuple. In this respect, there are again two important questions. First, are these bounds (and in particular the first one) tight? Second, given that the evaluation with an optimal attack becomes computationally intensive for large d values as just argued, does it mean that these bounds are unreachable by adversaries with realistic computing power?

We answer these questions in two steps. First, we investigate a simplified context with small d and m values such that the optimal attack is applicable. Second, we discuss heuristic attacks which approach the optimal attack efficiently.

Simplified case study. Figure 5 shows the information theoretic evaluation of a secure multiplication with $d = 3$ and $m = 2$.[6] We can clearly observe the larger leakage of optimal attack exploiting the $\delta = 9$ dimensions of the multiplication jointly, compared to the information provided by the encoding (i.e., the C1 adversary). As for the bounds, we first note that a simple (intuitive) bound is to assume that given two dependent values that are multiplied together, one

[6] Due to the large number of dimensions, the integrals were computed via sampling in this case, which also explains the lower noise variances that we could reach. However, we note that these lower noise levels were sufficient to reach the asymptotic (i.e., linear) regions of the information theoretic curves supporting our conclusions.

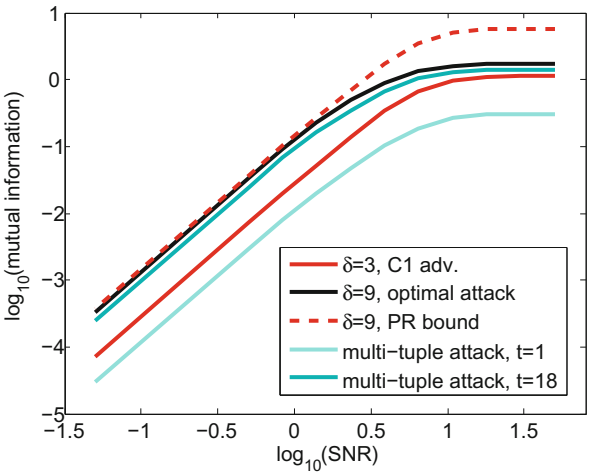

Fig. 5. Secure multiplication evaluation results ($d = 3$, $m = 2$).

leaks d horizontal d-tuples corresponding to one value (assuming the other to be known) and another d vertical d-tuples corresponding to the other value (under the same assumption). This leads to an estimation of the multiplication matrix leakage as $2d$ times the one of a single d-tuple, which is close to the $1.72d$ factor given by Prouff and Rivain in [50]. Hence, we added the latter bound on the figure (under the name PR bound). Concretely, it simply consists in multiplying the information of the encoding by $1.72d$ and turns out to be remarkably tight as soon as a sufficient amount of noise affects the measurements.[7]

Heuristic attacks. As the optimal attack in the previous paragraph becomes computationally intensive for large d and m values, we now consider alternatives that allow an adversary to exploit the information leakage of the multiplication matrix without summing over all the randomness and considering all the dimensions jointly. A first candidate is the recursive attack proposed by Battistello et al. at CHES 2016 [7]. In the following, we revisit and improve this attack by framing it as a Soft Analytical Side-Channel Attack (SASCA) [33,59].[8] In a SASCA, the adversary essentially describes all the leaking operations in his target implementation as a "factor graph" and then decodes the leakage information by exploiting the Belief Propagation (BP) algorithm. The main interest of this approach is that it allows combining the information of multiple leaking instructions (e.g., the cross products in a secure multiplication) locally, without the need to consider them jointly. Its time complexity depends on the diameter of the factor graph (which is constant when all target intermediate variables are directly connected as in the secure multiplication), the cost of the probabilities'

[7] Note that a parallel implementation would lead to a slightly better bound of $\approx d$ since reducing the amount of observable leakage samples by a factor d [5].

[8] Details about SASCA are provided in supplementary material for completeness.

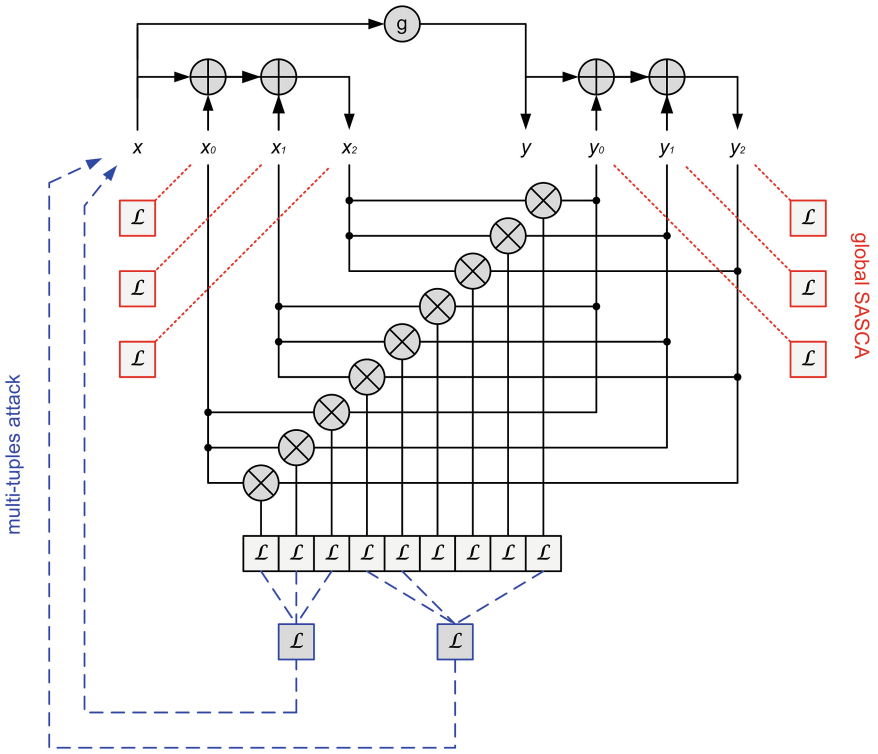

Fig. 6. Factor graph of a secure multiplication ($d = 3$).

updates (which is constant and depends on the bit size of the operations considered) and the number of these updates (which depends on the size of the factor graph and grows quadratically in d). The factor graph of a secure multiplication with $d = 3$ shares is pictured in Fig. 6. Its only specificity is that for the BP algorithm to succeed, we need to initialize the leakage on the shares x_0, x_1, x_2 and y_0, y_1, y_2, which means that a SASCA must consider the target operations more globally. In our experiments we added the leakage of these shares for this purpose, which can be obtained, e.g., when loading them into a register.

An alternative (and conceptually simple) approach allowing to get rid of the need of initialization is to always target d-tuples of informative leakage samples jointly. Such a "multi-tuple attack" can be viewed as an intermediate between the optimal attack targeting d^2 samples jointly and the previous SASCA targeting samples one by one, as illustrated in Fig. 6. More precisely, the optimal attack outlined in Sect. 3.2 exploits a leakage PDF $\Pr[\bar{\boldsymbol{l}}_{d^2}|k, x, \bar{\boldsymbol{y}}_{d^2}]$, where the d^2 subscripts of the vectors $\bar{\boldsymbol{l}}_{d^2}$ and $\bar{\boldsymbol{y}}_{d^2}$ now highlight their number of dimensions. In a multi-tuple attack, we simply select a number of d-tuples of which the combination depends on the target secret and approximate:

$$\Pr\left[\bar{\boldsymbol{l}}_{d^2}|k, x, \bar{\boldsymbol{y}}_{d^2}\right] \approx \Pr\left[\bar{\boldsymbol{l}}_{d}^{1}|k, x, \bar{\boldsymbol{y}}_{d}^{1}\right] \cdot \Pr\left[\bar{\boldsymbol{l}}_{d}^{2}|k, x, \bar{\boldsymbol{y}}_{d}^{2}\right] \cdot \ldots \cdot \Pr\left[\bar{\boldsymbol{l}}_{d}^{t}|k, x, \bar{\boldsymbol{y}}_{d}^{t}\right],$$

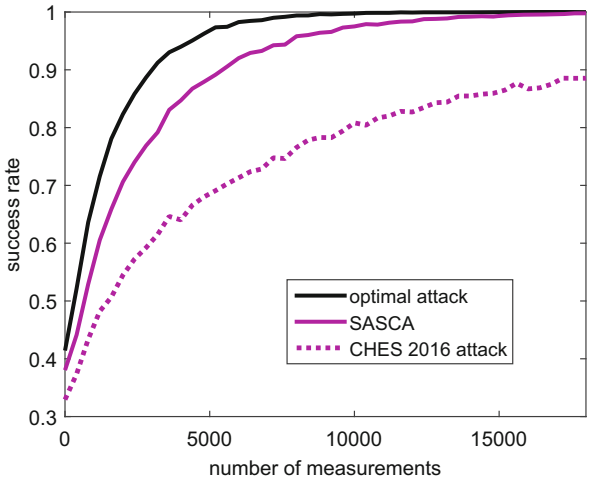

Fig. 7. Optimal attack vs. efficient heuristics ($d = 3$, $m = 2$, SNR $= \frac{2}{10}$).

where t is the number of tuples exploited.[9] As illustrated in Fig. 5, an attack using a single d-tuple (e.g., here a matrix line) only leads to little exploitable information, which is consistent with the observations in [7]. By contrast, increasing t rapidly allows reaching a close-to-optimal attack.

Note that the multi-tuples attack still does not scale well since the total number of informative d-tuples in the matrix multiplications grows following a binomial rule. So the most appealing attacks to target the secure multiplication algorithm are the CHES 2016 iterative one and the SASCA. Unfortunately, in these cases we face the problem that the heuristic nature of the decoding algorithms (which both propagate information locally without formal guarantees of convergence) does not formally lead them to output probabilities. Typically, by iterating the CHES 2016 and BP algorithms more, it is possible to artificially crush the probabilities of the variable nodes in the factor graph. So formally, we cannot evaluate the mutual information metric in this case. As a result, and for this part of our experiments only, we directly evaluated the success rate of an optimal attack, a SASCA and the CHES 2016 iterative attack (using exactly the same leaking operations as the SASCA) for various noise levels.

For example, Fig. 7 contains the result of such an experiment (for $\sigma_n^2 = 10$ meaning SNR $= \frac{2}{10}$) where we observe that (i) the SASCA outperforms the CHES 2016 iterative attack, and (ii) the SASCA leads to attack efficiencies that approach the optimal one. The first observation is easily explained since the CHES 2016 iterative attack can in fact be viewed as a modified version of SASCA. Namely, the main difference between the SASCA and the CHES 2016

[9] Note that whenever an imperfect model is used by the adversary/evaluator, the estimation of Eqs. 2 and 3 does not strictly converge towards the mutual information, but only to the so-called perceived information discussed in [24].

iterative attack is the fact that we take advantage of the relation between the two secrets that are multiplied (i.e., the g function in Fig. 6), which allows the BP algorithm to extract more information (while the factor graph of the CHES 2016 iterative attack ignores this connection).[10] As for the second observation, we launched attacks for different values of the SNR in order to verify whether the distance between the optimal attack and the SASCA is noise-dependent. For example, Fig. 12 in Appendix B shows the result of an experiment similar to the one of Fig. 7 but with better SNR ($\sigma_n^2 = 1$ meaning SNR = 2), leading to a tighter approximation. It suggests that the heuristic use of the BP algorithm in a SASCA against the multiplication of Fig. 6 tends to perform worse as the noise increases. In this respect, we note that our experiments consider the (more efficient) variation of SASCA where the factor graph is decoded for each measurement independently and probabilities are re-combined afterwards. It is an interesting open problem to investigate the (more expensive) version where the factor graph is extended for each new measurement (since it was shown in [59] that the gain such attacks provide over a standard DPA attack is independent of the noise in the context of an unprotected AES implementation).

Remark. As previously mentioned, extending these experiments to larger d and m values is not possible because the optimal attack becomes too expensive (computationally). By contrast, we could check that the success rate curves of the SASCA consistently outperform the ones of the CHES 2016 iterative attack by an approximate factor > 2 in measurement complexity, for larger m values. For example, we report the result of such a comparison for the relevant $m = 8$-bit case corresponding to the AES S-box in Appendix B, Fig. 13.

Overall, we conclude from this section that the IOL assumption and the PR bound for secure multiplications give rise to quite tight estimations of the information leakage of a masked implementation (at least for the leakage functions and noise levels considered experimentally). Furthermore, this leakage can also be exploited quite efficiently using heuristics such as the BP algorithm. We conjecture that these observations generally remain correct for most leakage functions, and when the number of shares in the masking schemes increases.

5.5 Reducing Cardinality with the ISL Assumption

Eventually, the previous experiments suggest that the evaluation of a masked implementation against multivariate attacks can boil down to the evaluation of the information leakage of a d-tuple. Yet, this still has evaluation cost proportional to 2^{dm}. Fortunately, at this stage we can use the ISL assumption and the bound discussed at Eurocrypt 2015 showing that this information can be (very efficiently) computed based on the information of a single share (essentially by raising this information to the security order), which has (now minimal) evaluation cost $\mathsf{E}_{\$} = \delta \cdot 2^m$ (or even 2^m if one assumes that the leakage function

[10] Technically, the rules used for updating the probabilities in the CHES 2016 attack are also presented slightly differently than in SASCA, where the BP algorithm is explicitly invoked with variable to factors and factors to variable message passing.

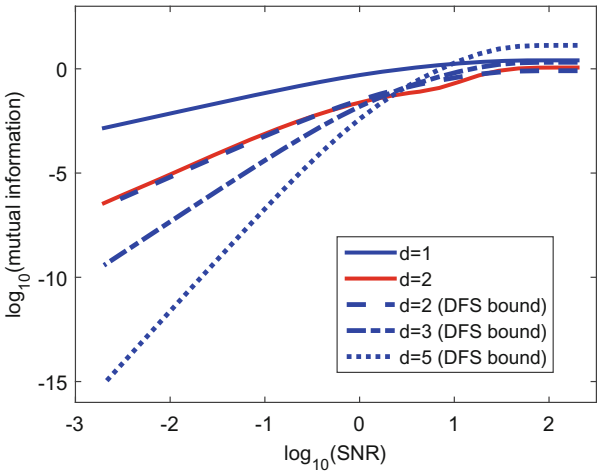

Fig. 8. Masked AES S-box evaluation results: case C1 with ISL assumption.

of the target implementation is similar for all operations, or if we bound the overall leakage based on the most informative d-tuple found) [20]. For completeness, we illustrate such a result in Fig. 8, where we compare the bound (denoted as DFS) and the true information leakage for $d = 2$, and only plot the bound for larger d's. As already mentioned, the big conceptual change at this step of our simplifications is that the ISL assumption is no longer a conservative one. If it turns out to be incorrect, then the security order of higher-order masking schemes may be less than predicted by the number of shares. Yet, as discussed in Sect. 2, this does not decrease the relevance of our method and bounds: it simply implies that applying them first requires to assess the security order of the target implementation (which we briefly discuss in Sect. 6.2).

Note also that as carefully discussed in [20] (Sect. 4.1, part c), the DFS bound is only conjectured and ignores a square root loss in the reduction from the mutual information to the statistical distance used in the proofs.[11] Yet, this square root loss vanishes when the noise increases, as per the upper bound in [50]. More precisely, this reference showed that the mutual information is (up to constants) lower than the statistical distance (not its square), and this inequality becomes an equality for low SNRs. In this respect, we recall that masking proofs are anyway only relevant for large enough noises (or low enough SNRs), which corresponds to the linear (left) parts of the information theoretic curves of Fig. 8 (i.e., where the DFS bound is tight). Intuitively, this is because masking can be viewed as a noise amplification mechanism. So without noise to amplify, the countermeasure does not provide any concrete benefits. In other words, either the

[11] Strictly speaking, it also ignores a small constant factor discussed in the optimal reduction given by [26], which is assumed to be a proof artifact and is at least not observed for the simple leakage functions considered in our experiments.

noise is too low and our tools do not apply but the implementation is insecure anyway, or the noise is sufficient and the bound applies. Technically, this is reflected by the hypotheses of masking security proofs, which require that the information of a single share (to be raised to the security order) is at least lower than one.[12] The DFS bound was also confirmed experimentally in [58].

6 Fast and Sound Leakage Assessment

6.1 Putting Things Together

By combining the IOL assumption, the PR bound for evaluating the leakage of a secure multiplication, the ISL assumption and the DFS bound for evaluating the leakage of an encoding with large number of shares, all evaluated and discussed in the previous section, we can now easily obtain the results of a security evaluation for the four adversaries outlined in Sect. 3.1. For example, Fig. 9 plots them for $d = 3, 5$ and 7 shares, for various noise levels. For readability, we only provide the results of the extreme attacks (C1 and C4). These curves are simply obtained by performing powers and sums of the information theoretic curve for the simplest possible case $d = 1$. In other words, we can evaluate the leakage of a masked implementation against optimal (highly multivariate) side-channel attacks at the cost of the evaluation of an unprotected implementation.

Note that the curves clearly highlight the need of a higher noise level when implementing higher-order masking schemes, in order to mitigate the noise reduction that is caused by the possibility to perform highly multivariate attacks (reflected by a shift of the curves towards the left of the figure). And quite naturally, they directly allow one to quantify the increasing impact of such attacks when the security order increases. For example, the factor between the measurement complexity of the adversary C1 (exploiting one tuple of leakage samples) and the optimal C4 ranges from 50 (for $d = 3$) to 100 (for $d = 7$).

In this respect, there is one final remark. In concrete implementations, it frequently happens that some of the target intermediate values appear several times (e.g., because they need to be reloaded for performing the cross products in a secure multiplication). In this case, the adversary can additionally average the noise for these target intermediate values, as proposed in [7]. As mentioned in Sect. 4, such an effect is also easy to integrate into our evaluations since it only corresponds to a shift of the information theoretic curves. However, it is worth emphasizing that this averaging process is applied to the shares (i.e., before their combination provides noise amplification), which implies that it is extremely damaging for the security of masking. Concretely, this means that averaging the leakage samples of a masked implementation with d shares by a factor d (because these shares are loaded d times to perform cross products) may lead to a reduction of the security level by a factor d^d. For illustration, Fig. 10

[12] Otherwise raising the information leakage of individual shares to some power may lead to larger values than the maximum m. For convenience, the following plots limit the mutual information to m when this happens (i.e., for too low noise levels).

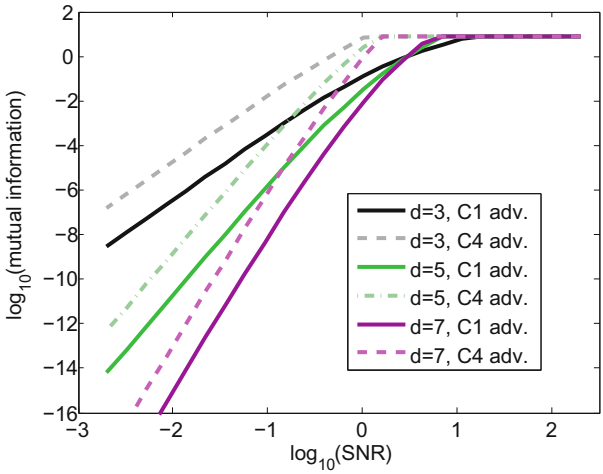

Fig. 9. Masked AES S-box evaluation results: cases C1 & C4 (with all assumptions).

shows the result of such a security evaluation in a context similar to Fig. 9, where the shares of each the masked multiplication are averaged d times, this times causing reductions of the security level by several orders of magnitude.

The difference between the multivariate attacks of Fig. 9 and the ones in Fig. 10 is easily explained by looking back at Fig. 8 where the leakage of a single d-tuple of shares is reported. First observe that in practice, any actual device has a fixed SNR: say 10^{-2} for illustration, leading to a mutual information of $\approx 10^{-13}$ for the $d = 5$ case. In case several (say ≈ 78) independent d tuples are combined in an attack (which is essentially what the C4 adversary of Fig. 9 achieves), the total amount of information available to the adversary is multiplied by a factor ≈ 78. This corresponds to a vertical shift in the information theoretic curves. Say now the adversary can average each of his leakage samples $d = 5$ times (which is essentially what the adversaries of Fig. 10 achieve). Then the SNR will be reduced 5 times, leading to an horizontal shift in the information theoretic curve, and reducing the averaged 5-tuples' information leakage by a factor $5^5 = 3125$, as per the curves' slope. This last observation suggests that each share in a masked implementation should be manipulated minimally.

Note that such an averaging is easy to perform in worst-case evaluations where the implementation is known to the adversary and the points in time where the shares are manipulated can be directly spot. But its exploitation with limited implementation knowledge may be more challenging. It is an interesting scope for further research to investigate whether some statistical tools can approach such worst-case attacks in this case (e.g., based on machine learning [37]).[13]

[13] Note also that traces averaging can be exploited constructively in the assessment of a security order. For example, in case the masks are known to the evaluator, he can average traces before evaluating the security order, leading to the efficiency gains [56].

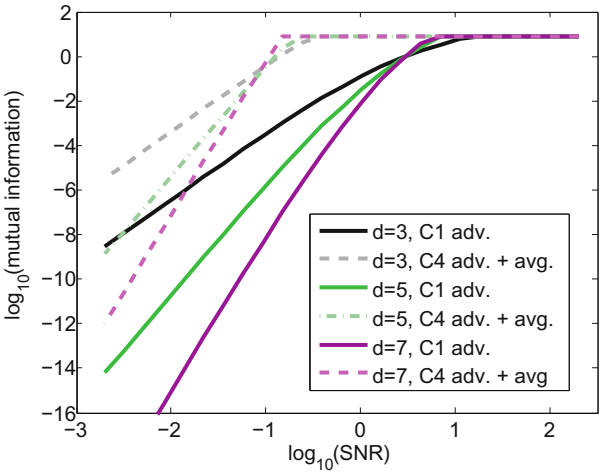

Fig. 10. Masked AES S-box evaluation results: cases C1 & C4 (with all assumptions & d-times averaging applied to the shares of the secure multiplications).

6.2 A Real-World Case Study

Our methodology has been recently applied (at CHES 2017) to a 32-bit shared implementation of the AES in an ARM Cortex M4 device [35]. As mentioned in introduction, such implementations are the typical targets for which our proof-based approach becomes necessary. This reference first assesses the security order by investigating reduced-shares versions of the algorithms (which can be viewed as an analogy to the reduced-rounds versions in block cipher cryptanalysis). Since overestimated orders are the main risk of overstated security (because they reflect the ISL assumption), the paper additionally considers a risk factor (i.e., evaluates the security for the best detected order and for its reduction by a conservative factor two). "Single tuple", "all tuples" and "all tuples + averaging" attacks are then analyzed, based on the number of leaking operations in the implementation code, and the number of shares' repetitions. It allows claiming so far unreported security levels, under well identified (and empirically falsifiable) assumptions.

6.3 Exploiting Computational Power

Eventually, and given some mutual information value extracted from the previous plots, we mention that one can easily insert this value in a metric-based bound in order to build a security graph, such as suggested in [21] and illustrated in Fig. 11. While such metric-based bounds only provide a conservative estimation of the impact of key enumeration in a side-channel attack [41,48], they are obtained withing seconds of computation on a desktop computer. We detail how to build such a graph and the heuristics we rely on in Appendix C.

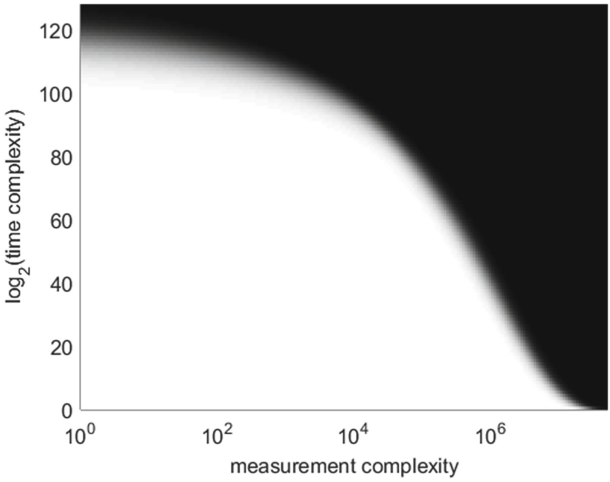

Fig. 11. Exemplary metric-based bound for a security graph (with MI $= 10^{-7}$).

6.4 Conclusions

1. On too simple evaluation methodologies. Looking at the power of multivariate (aka horizontal) side-channel attacks taking advantage of all the leaking operations in the multiplicative chain of a masked AES S-box, an important conclusion is that simple (univariate) evaluation strategies become increasingly irrelevant as the number of shares in a masked implementation increases.

2. On the need of formal methods and security order detection. As made clear in Sect. 2, the tools we provide in this paper only solve the "noise" part of the security evaluation problem for masked implementations. Hence, their combination with formal methods and security order detection techniques is an interesting scope for further research. Typically, one could extend the tools put forward in [3] in order to detect all the leaking operations in an assembly code (possibly with repetitions), then use leakage detection methods such as [13,22,30,42,55] to assess the security order of actual measurements, and finally evaluate their informativeness as we suggest in this paper, in order to obtain a fast assessment of the worst-case security level of an implementation.

3. On how to reach high security levels. Our results show that ensuring high security levels against optimal adversaries taking advantage of all the information provided by a masked implementation is challenging. It requires many shares, high noise levels and independence. In this respect, the application of our progresses to alternative multiplication chains [16,32], to the optimized algorithms in [8], to new primitives allowing more efficient masking (e.g., the proposal in [25] of which the complexity scales linearly in the number of shares and is well suited to guarantee the ISL assumption), and the combination of these ideas with parallel and/or hardware implementations (which improve security against multivariate attacks), is another interesting research direction.

4. On the "circuit size parameter" of masking security proofs. Eventually, our investigations are still limited to the evaluation of leakage samples that can be exploited via a divide-and-conquer strategy (i.e., attacks targeting independent parts of the key one by one). Yet, masking security proofs suggest that the success rate of an adversary is proportional to the target circuit size (i.e., the total amount of leakage samples) independent of whether these samples correspond to enumerable intermediate computations [20]. In this respect, analyzing the extent to which a SASCA exloiting a factor graph for a full masked implementation can confirm this fact is one more important open problem, which would first require a better formalization of such analytical attacks.

Acknowledgments. François-Xavier Standaert is a senior associate researcher of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in parts by the ERC project 724725 (acronym SWORD), the EU project REASSURE and the Brussels Region INNOVIRIS project SCAUT.

A Algorithms for the Masked S-box

Algorithm 1. Secure evaluation of a GF(2)-linear function g.

Require: Shares $x(i)$ such that $x = x(1) \oplus \cdots \oplus x(d)$.
Ensure: Shares $y(i)$ such that $\mathsf{g}(x) = y = y(1) \oplus \cdots \oplus y(d)$.
1: **for** i from 1 to d **do**
2: $\quad y(i) \leftarrow \mathsf{g}(x(i))$
3: **end for**

Algorithm 2. Multiplication of two masked secrets $\in \mathrm{GF}(2^m)$.

Require: Shares $x(i)$ and $y(i)$ such that $x = x(1) \oplus \cdots \oplus x(d)$ and $y = y(1) \oplus \cdots \oplus y(d)$.
Ensure: Shares $z(i)$ such that $x \otimes y = z = z(1) \oplus \cdots \oplus z(d)$.
1: **for** i from 1 to d **do**
2: $\quad$ **for** j from $i+1$ to d **do**
3: $\quad\quad r_{i,j} \xleftarrow{\mathrm{r}} \mathrm{GF}(2^m)$
4: $\quad\quad r_{j,i} \leftarrow (r_{i,j} \oplus x(i) \otimes y(j)) \oplus x(j) \otimes y(i)$
5: $\quad$ **end for**
6: **end for**
7: **for** i from 1 to d **do**
8: $\quad z(i) \leftarrow x(i) \otimes y(i)$
9: $\quad$ **for** j from 1 to $d, j \neq i$ **do**
10: $\quad\quad z(i) \leftarrow z(i) \oplus r_{i,j}$
11: $\quad$ **end for**
12: **end for**
13: **return** $(z(1), ..., z(d))$

Algorithm 3. Field multiplication of two elements $\in \mathrm{GF}(2^m)$.

Require: $x, y \in \mathrm{GF}(2^m)$.
Ensure: z such that $z = x \otimes y$.
1: $x' \leftarrow \mathsf{LogTab}[x]$
2: $y' \leftarrow \mathsf{LogTab}[y]$
3: $z' \leftarrow x' + y' \bmod 2^m - 1$
4: $z \leftarrow (x \neq 0 \wedge y \neq 0)\ \mathsf{aLogTab}[z']$
5: **return** z

B Additional Figures

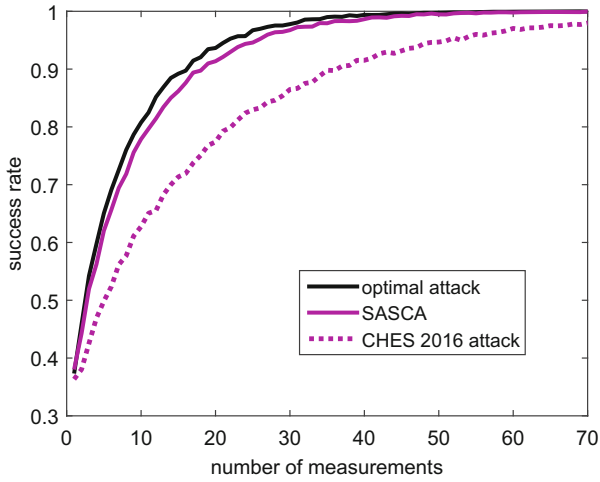

Fig. 12. Optimal attack vs. efficient heuristics ($d = 3$, $m = 2$, SNR $= 2$).

C Metric-Based Bound for the Key Rank

Very summarized, the two core ideas used in [21] to take the computational (enumeration) power of a divide-and-conquer adversary into account in a side-channel evaluation are: (i) to bound the success rate per S-box in function of the adversary's computational power thanks to the mutual information of an aggregated key variable K^c_{agg}, where c is an aggregation parameter (corresponding to the computational power), and (ii) to plug these success rate bounds into the metric-based rank-estimation algorithm of [48]. So technically, the only ingredient needed to exploit the same tools is the mutual information of the aggregated key variable (i.e., the so-called NAMI, for Normalized Aggregated Mutual Information). Unfortunately, the exact computation of the NAMI is impossible in our

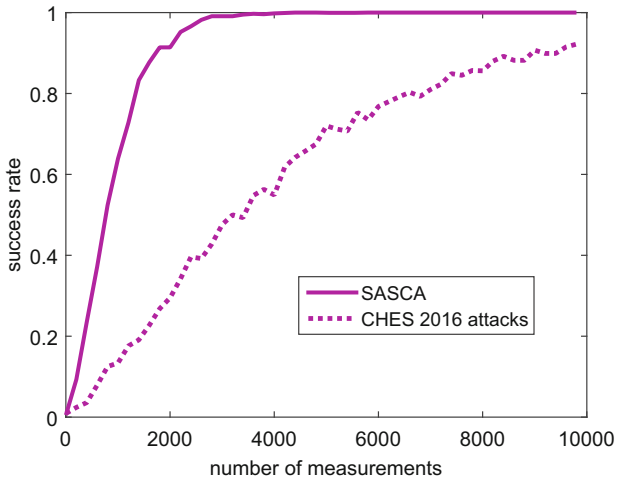

Fig. 13. Efficient heuristic attacks ($d = 3$, $m = 8$, SNR $= 2$).

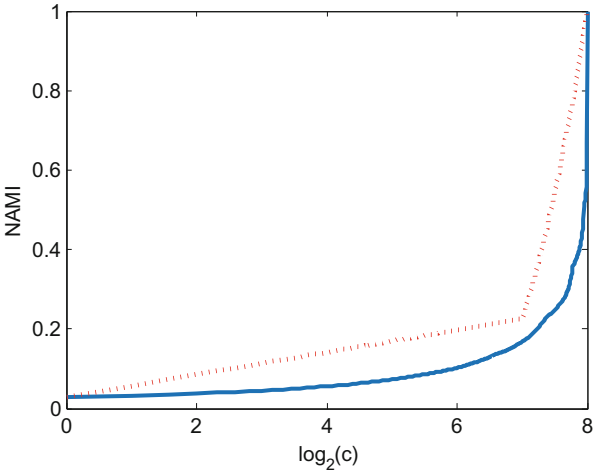

Fig. 14. Bound on the Normalized Aggregated Mutual Information.

case, since we do not have access to the probabilities of all the key candidates (that are combined during the aggregation process). So we need a way to bound the NAMI based on its first value $\text{NAMI}(c = 1) = \text{MI}(K; X, \bar{\boldsymbol{L}})$.

For this purpose, a simple observation is that for $c \leq 2^{m-1}$, aggregating $c = 2^q$ key candidates together can at most multiply the NAMI by $q+1$. The behavior of the NAMI for $c > 2^{m-1}$ is less intuitive (since in general, the definition of the NAMI is most intuitive when c is a power of two). Yet, as illustrated by the example in Fig. 14, a simple heuristic to bound it is then to connect the value of the NAMI at $c = 2^{m-1}$ and the maximum value of 1 that is reached at $c = 2^m$

by a straight line (which is obviously conservative as well, since the figure is in log-lin scale). Alternatively, when $\mathrm{MI}(K; X, \bar{\boldsymbol{L}}) < \frac{1}{2^m}$, an even simpler bound is to connect $\log(\mathrm{NAMI}(c = 1)) = \log(\mathrm{MI}(K; X, \bar{\boldsymbol{L}}))$ and $\log(\mathrm{NAMI}(c = 2^m)) = 0$ by a straight line. More accurate bounds are certainly reachable, yet not useful here since the general focus of the paper is on providing fast intuitions regarding the computational security of a key manipulated by a leaking device.

References

1. Archambeau, C., Peeters, E., Standaert, F.-X., Quisquater, J.-J.: Template attacks in principal subspaces. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 1–14. Springer, Heidelberg (2006). https://doi.org/10.1007/11894063_1
2. Balasch, J., Gierlichs, B., Grosso, V., Reparaz, O., Standaert, F.-X.: On the cost of lazy engineering for masked software implementations. In: Joye, M., Moradi, A. (eds.) CARDIS 2014. LNCS, vol. 8968, pp. 64–81. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16763-3_5
3. Barthe, G., Belaïd, S., Dupressoir, F., Fouque, P.-A., Grégoire, B., Strub, P.-Y.: Verified proofs of higher-order masking. In: Oswald and Fischlin [47], pp. 457–485
4. Barthe, G., Belaïd, S., Dupressoir, F., Fouque, P.-A., Grégoire, B., Strub, P.-Y., Zucchini, R.: Strong non-interference and type-directed higher-order masking. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S., (eds.) Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October, 2016, pp. 116–129. ACM (2016)
5. Barthe, G., Dupressoir, F., Faust, S., Grégoire, B., Standaert, F.-X., Strub, P.-Y.: Parallel implementations of masking schemes and the bounded moment leakage model. In: Coron and Nielsen [15], pp. 535–566
6. Batina, L., Robshaw, M. (eds.): CHES 2014. LNCS, vol. 8731. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44709-3
7. Battistello, A., Coron, J.-S., Prouff, E., Zeitoun, R.: Horizontal side-channel attacks and countermeasures on the ISW masking scheme. In: Gierlichs and Poschmann [29], pp. 23–39
8. Belaïd, S., Benhamouda, F., Passelègue, A., Prouff, E., Thillard, A., Vergnaud, D.: Randomness complexity of private circuits for multiplication. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 616–648. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_22
9. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28632-5_2
10. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 398–412. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_26
11. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Kaliski, B.S., Koç, K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 13–28. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36400-5_3
12. Choudary, M.O.: Efficient multivariate statistical techniques for extracting secrets from electronic devices. Ph.D. thesis, University of Cambridge (2014)

13. Cooper, J., De Mulder, E., Goodwill, G., Jaffe, J., Kenworthy, G., Rohatgi, P.: Test vector leakage assessment (TVLA) methodology in practice (extended abstract). In: ICMC 2013. http://icmc-2013.org/wp/wp-content/uploads/2013/09/goodwillkenworthtestvector.pdf
14. Coron, J.-S., Giraud, C., Prouff, E., Renner, S., Rivain, M., Vadnala, P.K.: Conversion of security proofs from one leakage model to another: a new issue. In: Schindler, W., Huss, S.A. (eds.) COSADE 2012. LNCS, vol. 7275, pp. 69–81. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29912-4_6
15. Coron, J.-S., Nielsen, J.B. (eds.): EUROCRYPT 2017. LNCS, vol. 10210. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7
16. Coron, J.-S., Prouff, E., Rivain, M., Roche, T.: Higher-order side channel security and mask refreshing. In: Moriai, S. (ed.) FSE 2013. LNCS, vol. 8424, pp. 410–424. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43933-3_21
17. Daemen, J., Rijmen, V.: The wide trail design strategy. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 222–238. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45325-3_20
18. Ding, A.A., Zhang, L., Fei, Y., Luo, P.: A statistical model for higher order DPA on masked devices. In: Batina and Robshaw [6], pp. 147–169
19. Duc, A., Dziembowski, S., Faust, S.: Unifying leakage models: from probing attacks to noisy leakage. In: Nguyen and Oswald [45], pp. 423–440
20. Duc, A., Faust, S., Standaert, F.-X.: Making masking security proofs concrete - or how to evaluate the security of any leaking device. In: Oswald and Fischlin [47], pp. 401–429
21. Duc, A., Faust, S., Standaert, F.-X.: Making masking security proofs concrete or how to evaluate the security of any leaking device (extended version). IACR Cryptology ePrint Archive 2015, 119 (2015)
22. Durvaux, F., Standaert, F.-X.: From improved leakage detection to the detection of points of interests in leakage traces. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 240–262. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_10
23. Durvaux, F., Standaert, F.-X., Del Pozo, S.M.: Towards easy leakage certification. In: Gierlichs and Poschmann [29], pp. 40–60
24. Durvaux, F., Standaert, F.-X., Veyrat-Charvillon, N.: How to certify the leakage of a chip? In: Nguyen and Oswald [45], pp. 459–476
25. Dziembowski, S., Faust, S., Herold, G., Journault, A., Masny, D., Standaert, F.-X.: Towards sound fresh re-keying with hard (physical) learning problems. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 272–301. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_10
26. Dziembowski, S., Faust, S., Skorski, M.: Noisy leakage revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 159–188. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_6
27. Fei, Y., Luo, Q., Ding, A.A.: A statistical model for dpa with novel algorithmic confusion analysis. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 233–250. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33027-8_14
28. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual information analysis. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 426–442. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85053-3_27
29. Gierlichs, B., Poschmann, A.Y. (eds.): CHES 2016. LNCS, vol. 9813. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53140-2

30. Goodwill, G., Jun, B., Jaffe, J., Rohatgi, P.: A testing methodology for side channel resistance validation. In: NIST Non-invasive Attack Testing Workshop (2011). http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf
31. Goudarzi, D., Rivain, M.: How fast can higher-order masking be in software? In: Coron and Nielsen [15], pp. 567–597
32. Grosso, V., Prouff, E., Standaert, F.-X.: Efficient masked s-boxes processing – a step forward. In: Pointcheval, D., Vergnaud, D. (eds.) AFRICACRYPT 2014. LNCS, vol. 8469, pp. 251–266. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06734-6_16
33. Grosso, V., Standaert, F.-X.: ASCA, SASCA and DPA with enumeration: which one beats the other and when? In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9453, pp. 291–312. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48800-3_12
34. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: securing hardware against probing attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_27
35. Journault, A., Standaert, F.-X.: Very high order masking: efficient implementation and security evaluation. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 623–643. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66787-4_30
36. Lemke-Rust, K., Paar, C.: Gaussian mixture models for higher-order side channel analysis. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 14–27. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74735-2_2
37. Lerman, L., Poussier, R., Bontempi, G., Markowitch, O., Standaert, F.-X.: Template attacks vs. machine learning revisited (and the curse of dimensionality in side-channel analysis). In: Mangard, S., Poschmann, A.Y. (eds.) COSADE 2014. LNCS, vol. 9064, pp. 20–33. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21476-4_2
38. Lomné, V., Prouff, E., Rivain, M., Roche, T., Thillard, A.: How to estimate the success rate of higher-order side-channel attacks. In: Batina and Robshaw [6], pp. 35–54
39. Mangard, S., Oswald, E., Standaert, F.-X.: One for all - all for one: unifying standard differential power analysis attacks. IET Inf. Secur. **5**(2), 100–110 (2011)
40. Mangard, S., Popp, T., Gammel, B.M.: Side-channel leakage of masked CMOS gates. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 351–365. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30574-3_24
41. Martin, D.P., Mather, L., Oswald, E., Stam, M.: Characterisation and estimation of the key rank distribution in the context of side channel evaluations. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10031, pp. 548–572. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_20
42. Mather, L., Oswald, E., Bandenburg, J., Wójcik, M.: Does my device leak information? An *a priori* statistical power analysis of leakage detection tests. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 486–505. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_25
43. Mather, L., Oswald, E., Whitnall, C.: Multi-target DPA attacks: Pushing DPA beyond the limits of a desktop computer. In: Sarkar and Iwata [53], pp. 243–261
44. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48285-7_33

45. Nguyen, P.Q., Oswald, E. (eds.): EUROCRYPT 2014. LNCS, vol. 8441. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5
46. Nikova, S., Rijmen, V., Schläffer, M.: Secure hardware implementation of nonlinear functions in the presence of glitches. J. Cryptol. **24**(2), 292–321 (2011)
47. Oswald, E., Fischlin, M. (eds.): EUROCRYPT 2015. LNCS, vol. 9056. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5
48. Poussier, R., Grosso, V., Standaert, F.-X.: Comparing approaches to rank estimation for side-channel security evaluations. In: Homma, N., Medwed, M. (eds.) CARDIS 2015. LNCS, vol. 9514, pp. 125–142. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31271-2_8
49. Prouff, E.: DPA attacks and s-boxes. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 424–441. Springer, Heidelberg (2005). https://doi.org/10.1007/11502760_29
50. Prouff, E., Rivain, M.: Masking against side-channel attacks: a formal security proof. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 142–159. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_9
51. Rivain, M.: On the exact success rate of side channel analysis in the Gaussian model. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 165–183. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04159-4_11
52. Rivain, M., Prouff, E.: Provably secure higher-order masking of AES. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 413–427. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15031-9_28
53. Sarkar, P., Iwata, T. (eds.): ASIACRYPT 2014. LNCS, vol. 8874. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45608-8
54. Schindler, W., Lemke, K., Paar, C.: A stochastic model for differential side channel cryptanalysis. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 30–46. Springer, Heidelberg (2005). https://doi.org/10.1007/11545262_3
55. Schneider, T., Moradi, A.: Leakage assessment methodology - extended version. J. Crypt. Eng. **6**(2), 85–99 (2016)
56. Standaert, F.-X.: How (not) to use Welch's t-test in side-channel security evaluations. IACR Cryptology ePrint Archive **2017**, 138 (2017)
57. Standaert, F.-X., Malkin, T.G., Yung, M.: A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 443–461. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_26
58. Standaert, F.-X., Veyrat-Charvillon, N., Oswald, E., Gierlichs, B., Medwed, M., Kasper, M., Mangard, S.: The world is not enough: another look on second-order DPA. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 112–129. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_7
59. Veyrat-Charvillon, N., Gérard, B., Standaert, F.-X.: Soft analytical side-channel attacks. In: Sarkar and Iwata [53], pp. 282–296