

# Making Masking Security Proofs Concrete

## Or How to Evaluate the Security of any Leaking Device

–Extended Version –

Alexandre Duc<sup>1</sup>, Sebastian Faust<sup>1,2</sup>, François-Xavier Standaert<sup>3</sup>

<sup>1</sup> EPFL, Lausanne, Switzerland

<sup>2</sup> Horst Görtz Institute, Ruhr-University Bochum, Germany

<sup>3</sup> ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium  
e-mails: alexandre.duc@epfl.ch; sebastian.faust@gmail.com; fstandae@uclouvain.be.

**Abstract.** We investigate the relationships between theoretical studies of leaking cryptographic devices and concrete security evaluations with standard side-channel attacks. Our contributions are in four parts. First, we connect the formal analysis of the masking countermeasure proposed by Duc et al. (Eurocrypt 2014) with the Eurocrypt 2009 evaluation framework for side-channel key recovery attacks. In particular, we re-state their main proof for the masking countermeasure based on a mutual information metric, which is frequently used in concrete physical security evaluations. Second, we discuss the tightness of the Eurocrypt 2014 bounds based on experimental case studies. This allows us to conjecture a simplified link between the mutual information metric and the success rate of a side-channel adversary, ignoring technical parameters and proof artifacts. Third, we introduce heuristic (yet well-motivated) tools for the evaluation of the masking countermeasure when its independent leakage assumption is not perfectly fulfilled, as it is frequently encountered in practice. Thanks to these tools, we argue that masking with non-independent leakages may provide improved security levels in certain scenarios. Eventually, we consider the tradeoff between the measurement complexity and the key enumeration time complexity in divide-and-conquer side-channel attacks, and show that these complexities can be lower bounded based on the mutual information metric, using simple and efficient algorithms. The combination of these observations enables significant reductions of the evaluation costs for certification bodies.

## 1 Introduction

Side-channel attacks are an important concern for the security of cryptographic hardware, and masking is one of the most investigated solutions to counteract them. Its underlying principle is to randomize any sensitive data manipulated by a leaking implementation by splitting it into  $d$  shares, and to perform all the computations on these shared values only. Intuitively, such a process is expected to force the adversary to combine several leakages corresponding to the different shares in order to recover secret information. As a result, it has first been shown by Chari et al. that the measurement complexity of a specialized attack – namely a single-bit Differential Power Analysis (DPA) [44] – against

a carefully implemented masked computation (i.e. where the leakages of all the shares are independent and sufficiently noisy) increases exponentially with  $d$  [18]. Following this seminal work, a number of progresses have been made in order to state the security guarantee of masking in both general and rigorous terms. For example, Ishai, Sahai and Wagner introduced a compiler (next referred to as the ISW compiler), able to encode any circuit into an equivalent (secret-shared) one, and proved its security against so-called probing adversaries, able to read a bounded number of wires in the implementation [41]. A practical counterpart to these results was published at Asiacrypt 2010, where Standaert et al. analyzed the security of several masked implementations [72], using the information theoretic framework introduced in [71]. While this analysis was specialized to a few concrete case studies, it allowed confirming the exponential security increase provided by masking against actual leakages, typically made of a noisy but arbitrary function of the target device’s state. Following, Faust et al. attempted to analyze the ISW compiler against more realistic leakage functions, and succeeded to prove its security against computationally bounded (yet still unrealistic) ones, e.g. in the  $AC^0$  complexity class [31]. Prouff and Rivain then made a complementary step towards bridging the gap between the theory and practice of masking schemes, by providing a formal information theoretic analysis of a wide (and realistic) class of so-called noisy leakage functions [60]. Eventually, Duc et al. turned this analysis into a simulation-based security proof, under standard conditions (i.e. chosen-message rather than random-message attacks, without leak-free components, and with reduced noise requirements) [27]. The central and fundamental ingredient of this last work was a reduction from the noisy leakage model of Prouff and Rivain to the probing model of Ishai et al.

**Our contribution.** In view of this state-of-the-art, one of the main remaining question regarding the security of the masking countermeasure is whether its proofs can be helpful in the security evaluation of concrete devices. That is, can we state theorems for masking so that the hypotheses can be easily fulfilled by hardware designers, and the resulting guarantee is reflective of the actual security level of the target implementation. For this purpose, we first observe that the proofs in [27, 60] express their hypothesis for the amount of noise in the shares’ leakages based on a statistical distance. This is in contrast with the large body of published work where the mutual information metric introduced in [71] is estimated for various implementations (e.g. [4, 16, 33, 36, 39, 52, 61, 62, 67, 74, 78]). Since the latter metric generally carries more intuition (see, e.g. [3] in the context of linear cryptanalysis), and benefits from recent advances in leakage certification, allowing to make sure that its estimation is accurate and based on sound assumptions [28, 29], we first provide a useful link between the statistical distance and mutual information, and also connect them with easy-to-interpret (but more specialized) tools such as the Signal-to-Noise Ratio (SNR). We then re-state the theorems of Duc et al. based on the mutual information metric in two relevant scenarios. Namely, we consider both the security of an idealized implementation with a “leak-free refreshing” of the shares, and the one of a standard ISW-like encoding (i.e. capturing any type of leaking computation).

Interestingly, the implementation with leak-free refreshing corresponds to the frequently investigated (practical) context where a side-channel attack aims at key recovery, and only targets the  $d$  shares' leakage of a so-called sensitive intermediate variable (i.e. that depends on the plaintext and key) [22]. So despite being less interesting from a theoretical point of view, this scenario allows us to compare the theorem bounds with concrete attacks. Taking advantage of this comparison, we discuss the bounds' tightness and separate parameters that are physically motivated from more "technical ones" (that most likely result from proof artifacts). As a result, we conjecture a simplified link between the mutual information metric and the success rate of a side-channel adversary, which allows accurate approximations of the attacks' measurement complexity at minimum (evaluation) cost. We further illustrate that the noise condition for masking has a simple and intuitive interpretation when stated in terms of SNR.

Next, we note that the published results about masking (including the previously mentioned theorems and conjecture) assume independence between the leakages corresponding to different shares in an implementation. Yet, concrete experiments have shown that small (or even large) deviations from this assumption frequently occur in practice (see, e.g. [5, 21, 49, 64]). Hence, we complete our discussion by providing sound heuristics to analyze the impact of "non-independent leakages" which allow, for the first time, to evaluate and predict the security level of a masked implementation in such imperfect conditions.

Eventually, we consider the tradeoff between measurement complexity and time complexity in the important context of divide-and-conquer attacks. Previously known approaches for this purpose were based on launching key enumeration and/or rank estimation algorithms for multiple attacks, and to average results to obtain a success rate [75, 76]. We provide an alternative solution, where success rates (possibly obtained from estimations of the mutual information metric) are estimated/bounded for all the target key bytes of the divide-and-conquer attack first, and the impact of enumeration is evaluated only once afterwards.

Summarizing, the combination of these observations highlights that the security evaluation of a masked implementation boils down to the estimation of the mutual information between its shares and their corresponding leakages. Incidentally, the tools introduced in this paper apply identically to unprotected implementations, or implementations protected with other countermeasures, as long as one can estimate the same mutual information metric for the target intermediate values. Therefore, our results clarify the long standing open question whether the (informal) link between information theoretic and security metrics in the Eurocrypt 2009 evaluation framework [71] can be proved formally. They also have important consequences for certification bodies, since they translate the (worst-case) side-channel evaluation problem into the well-defined challenge of estimating a single metric, leading to significantly reduced evaluation costs.

**Notations.** We next use capital letters for random variables, small caps for their realizations and hats for estimations. Vectors will be denoted with bold notations, functions with sans serif fonts, and sets with calligraphic ones.

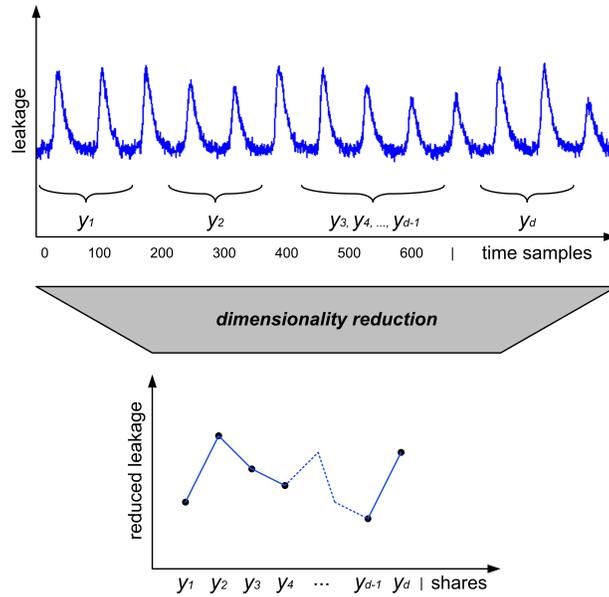
## 2 Background

### 2.1 Leakage traces and assumptions

Let  $y$  be a  $n$ -bit sensitive value manipulated by a leaking device. Typically, it could be the output of an S-box computation such that  $y = S(x \oplus k)$  with  $n$ -bit plaintext/key words  $x$  and  $k$ . Let  $y_1, y_2, \dots, y_d$  be the  $d$  shares representing  $y$  in a Boolean masking scheme (i.e.  $y = y_1 \oplus y_2 \oplus \dots \oplus y_d$ ). In a side-channel attack, the adversary is provided with some information (aka leakage) on each share. Typically, this leakage takes the form of a random variable  $L_{y_i}$  that is the output of a leakage function  $L$  with  $y_i$  and a noise variable  $R_i$  as arguments:

$$L_{y_i} = L(y_i, R_i) . \quad (1)$$

The top of Figure 1 represents a leakage trace corresponding to the manipulation of  $d$  shares. Concretely, each subtrace  $L_{y_i}$  is a vector of which the elements represent time samples. Whenever accessing a single time sample  $t$ , we use the notation  $L_{y_i}^t = L^t(y_i, R_i^t)$ . From this general setup, a number of assumptions are frequently used in the literature. We will consider the following three.



**Fig. 1.** Leakage trace & reduced leakage trace of a  $d$ -shared secret.

**a. Selection of points-of-interest / dimensionality reduction.** For convenience, a number of attacks start with a pre-processing in order to reduce each leakage subtrace  $L_{y_i}$  to a scalar random variable  $L_{y_i}$ . Such a pre-processing is motivated both by popular side-channel distinguishers such as Correlation Power Analysis (CPA) [14], which can only deal with univariate data, and by

the easier representation of small dimensional data spaces. In this respect, even distinguishers that naturally extend towards multivariate data (such as Template attacks (TA) [19], Linear Regression (LR) [69] or Mutual Information Analysis (MIA) [34]) generally benefit from some dimensionality reduction. The latter step can be achieved heuristically, for example by detecting leakage samples where one distinguisher works best, or more systematically using state-of-the-art tools such as Principal Component Analysis (PCA) [2], Linear Discriminant Analysis (LDA) [70] or Kernel Discriminant Analysis (KDA) [15]. An example of reduced leakage trace is represented at the bottom of Figure 1.

**b. Additive noise.** A standard assumption in the literature is to consider leakage functions made of a deterministic part  $G(y_i)$  and additive noise  $N_i$ :

$$\mathbf{L}_{y_i} = \mathbf{L}(y_i, \mathbf{R}_i) \approx G(y_i) + \mathbf{N}_i . \quad (2)$$

For example, a typical setting is to assume reduced leakages to be approximately generated as the combination of a Hamming weight function (or some other simple function of the shares’ bits) with additive Gaussian noise [48, 69].

**c. Independence condition.** A secure (serial) implementation of the masking countermeasure requires that each leakage vector  $\mathbf{L}_{y_i}$  depends only on a single share (parallel implementations were recently discussed in [7]). If respected, this condition ensures that a  $d$ -share masking will lead to a  $(d - 1)^{\text{th}}$ -order secure implementation as defined in [22]. That is, it guarantees that every  $(d - 1)$ -tuple of leakage vectors is independent of any sensitive variable. This means that any adversary targeting the implementation will have to “combine” the information of at least  $d$  shares, and that extracting information from these  $d$  shares will require to estimate a  $d^{\text{th}}$ -order moment of the leakage distribution (conditioned on a sensitive variable) – a task that becomes exponentially hard in  $d$  if the noise is sufficient. In software implementations, independence typically requires avoiding transition-based leakages (i.e. leakages that depend on the distance between shares rather than directly on the shares) [5, 21]. In hardware implementations, various physical defaults can also invalidate the independence assumption (e.g. glitches)[49], which motivates research efforts to mitigate this risk, both at the hardware level (e.g. [53]) and algorithmic level (e.g. [56]).

Note that only this last (independence) assumption is strictly needed for the following proofs of Section 3 to hold. By contrast, the previous assumptions (a) and (b) will be useful to provide practical intuition in Section 4. Furthermore, it is worth noting that slight deviations from this independence assumption (i.e. slight dependencies between the shares’ leakages) may still lead to concrete security improvements, despite falling outside the proofs’ formal guarantees. Such (practically meaningful) contexts will be further analyzed in Section 4.2.

## 2.2 Evaluation metrics

Following [71], one generally considers two types of evaluation metrics for leaking cryptographic devices. First, information theoretic metrics aim to capture the amount of information available in a side-channel, independent of the adversary

exploiting it. Second, security metrics aim to quantify how this information can be exploited by some concrete adversary. As will be clear next, the two types of metrics are related. For example, in the context of standard DPA attacks [49], they both measure the prediction of the (true) leakage function with some model, the latter usually expressed as an estimation of the leakage Probability Density Function (PDF). Yet they differ since information theoretic metrics only depend on the leakage function and model, while security metrics also depend on the adversary’s computational power. For example, the capacity to enumerate key candidates may improve security metrics, but has no impact on information theoretic ones. Our goal in the following is to draw a formal connection between information theoretic and security metrics, i.e. between the amount of leakage provided by an implementation and its (worst-case) security level.

In the case of masking, proofs informally state that “*given that the leakage of each share is independent of each other and sufficiently noisy, the security of the implementation increases exponentially in the number of shares*”. So we need the two types of metrics to quantify the noise condition and security level.

**a. Metrics to quantify the noise condition.** In general (i.e. without assumptions on the leakage distribution), the noise condition on the shares can be expressed with an information theoretic metric. The Mutual Information (MI) advocated in [71] is the most frequently used candidate for this purpose:

$$\text{MI}(Y_i; \mathbf{L}_{Y_i}) = H[Y_i] + \sum_{y_i \in \mathcal{Y}} \Pr[y_i] \cdot \sum_{\mathbf{l}_{y_i} \in \mathcal{L}} \Pr[\mathbf{l}_{y_i} | y_i] \cdot \log_2 \Pr[y_i | \mathbf{l}_{y_i}], \quad (3)$$

where we use the notation  $\Pr[Y_i = y_i] =: \Pr[y_i]$  when clear from the context. Note that whenever trying to compute this quantity from an actual implementation, evaluators face the problem that the leakage PDF is unknown and can only be sampled and estimated. As a result, one then computes the Perceived Information (PI), which is the evaluator’s best estimate of the MI [64]:

$$\hat{\text{PI}}(Y_i; \mathbf{L}_{Y_i}) = H[Y_i] + \sum_{y_i \in \mathcal{Y}} \Pr[y_i] \cdot \sum_{\mathbf{l}_{y_i} \in \mathcal{L}} \Pr_{\text{chip}}[\mathbf{l}_{y_i} | y_i] \cdot \log_2 \hat{\Pr}_{\text{model}}[y_i | \mathbf{l}_{y_i}], \quad (4)$$

with  $\Pr_{\text{chip}}$  the true chip distribution that can only be sampled and  $\hat{\Pr}_{\text{model}}$  the adversary’s estimated model. For simplicity, we will ignore this issue and use the MI in our discussions (conclusions would be identical with the PI [40]).

Interestingly, when additionally considering reduced leakages with additive Gaussian noise, and restricting the evaluation to so-called “first-order information” (i.e. information lying in the first-order statistical moments of the leakage PDF, which is typically the case for the leakage of each share), simpler metrics can be considered [48]. For example, the SNR defined by Mangard at CT-RSA 2004 in [46] is of particular interest for our following discussions:

$$\text{SNR} = \frac{\hat{\text{var}}_{Y_i} \left( \hat{\mathbb{E}}_{n_i}(L_{Y_i}) \right)}{\hat{\mathbb{E}}_{Y_i} \left( \hat{\text{var}}_{n_i}(L_{Y_i}) \right)}, \quad (5)$$

where  $\hat{\mathbb{E}}$  is the sample mean operator and  $\hat{\text{var}}$  is the sample variance. Summarizing, stating the noise condition based on the MI metric is more general (as it can capture any leakage PDF). By contrast, the SNR provides a simpler and more intuitive condition in a more specific but practically relevant context.

Eventually, the previous works of Prouff–Rivain and Duc et al. [27, 60] consider the following Statistical Distance (SD) to state their noise condition:

$$\text{SD}(Y_i; Y_i | \mathbf{L}_{Y_i}) = \sum_{\mathbf{l}_{y_i} \in \mathcal{L}} \Pr[\mathbf{l}_{y_i}] \cdot \mathbf{d}(Y_i; Y_i | \mathbf{l}_{y_i}), \quad (6)$$

with  $\mathbf{d}$  the Euclidean norm in [60] and  $\mathbf{d}(X_1, X_2) = \frac{1}{2} \sum_{x \in \mathcal{X}} |\Pr[X_1 = x] - \Pr[X_2 = x]|$  in [27]. In their terminology, a leakage function  $\mathbf{L}$  is then called “ $\delta$ -noisy” if  $\delta = \text{SD}(Y_i; Y_i | \mathbf{L}_{Y_i})$ , which was useful to connect different leakage models.

As previously mentioned, some of these metrics can be related under certain conditions. For example, in the context of univariate Gaussian random variables, the MI can be approximated from Pearson’s correlation coefficient [48], which was also connected to the SNR by Mangard [46]. The combination of those links corresponds to the classical MI bound in Cover and Thomas [24]:

$$\text{MI}(Y_i; \mathbf{L}_{Y_i}) \approx -\frac{1}{2} \log \left( 1 - \left( \frac{1}{\sqrt{1 + \frac{1}{\text{SNR}}}} \right)^2 \right) \leq \frac{1}{2} \log (1 + \text{SNR}). \quad (7)$$

In Section 3.1, we show that the MI and SD metrics can be connected as well.

**b. Metrics to quantify the security result.** Quantifying security requires defining the adversary’s goal. Current side-channel attacks published in the literature mostly focus on key recovery. In this context, one can easily evaluate the exploitation of the leakages with a success rate, as suggested in [71], defined for any sensitive variable and adapted to our masking context as follows:

**Definition 1 (Key recovery success rate).** *Let  $y_1, y_2, \dots, y_d$  be  $d$  leaking shares of a sensitive value  $y$ , with corresponding leakages  $\mathbf{l}_{y_1}, \mathbf{l}_{y_2}, \dots, \mathbf{l}_{y_d}$ , and  $m$  a number of measurements. We define the key recovery success rate  $\text{SR}^{\text{kr}}$  as the probability that an adversary  $\mathcal{A}$  recovers the sensitive value  $y$  given  $m \times d$  leakages subtraces, with the leakages generated from random (uniform) plaintexts.*

Key recovery is a weak security notion from a cryptographic point of view. As a result, rigorous proofs for masking such as the one of Duc et al. in [27] rather define security using the standard real world / ideal world paradigm, which consider two settings: the ideal world where the adversary attacks the algorithm of a cryptographic scheme in a black-box way, and the real world where he additionally obtains leakages. In this context, a scheme is said to be secure if for any adversary in the real world there exists an adversary in the ideal world. In other words: any attack that can be carried out given the leakages can also be carried out in a black-box manner. A proof of security usually involves

constructing an efficient simulator that is able to simulate the leakages just giving black-box access to the attacked cryptographic scheme. Whenever considering this (standard) indistinguishability-based security notion, we will denote the adversary’s success probability of distinguishing the two worlds with  $\text{SR}^{\text{dist}}$ .

### 3 Making proofs concrete: theory

In this section, we discuss theoretical tweaks allowing to improve the concreteness of masking proofs. For this purpose, we recall three important leakage models that are relevant for our work. First, the  $t$ -probing and  $\epsilon$ -probing (aka random probing) models were introduced in [41]. In the former one, the adversary obtains  $t$  intermediate values of the computation (e.g. can probe  $t$  wires if we compute in binary fields). In the latter one, he obtains each of these intermediate values with probability  $\epsilon$ , and gets  $\perp$  with probability  $1 - \epsilon$  (where  $\perp$  means no knowledge). Using a Chernoff-bound it is easy to show that security in the  $t$ -probing model reduces to security in the  $\epsilon$ -probing model for certain values of  $\epsilon$ . Second, the noisy leakage model describes many realistic side-channel attacks and allows an adversary to obtain each intermediate value perturbed with a  $\delta$ -noisy leakage function  $L$  [60]. As mentioned in the previous section, a leakage function  $L$  is called  $\delta$ -noisy if for a uniformly random variable  $Y$  (over the field  $\mathbb{F}$ ) we have  $\text{SD}(Y; Y|L_Y) \leq \delta$ . In contrast with the conceptually simpler  $\epsilon$ -probing model, the adversary obtains noisy leakages on each intermediate variable. For example, in the context of masking, he obtains  $L(Y_i, \mathbf{R}_i)$  for all the shares  $Y_i$ , which is more reflective of actual implementations where the adversary can potentially observe the leakage of all these shares, since they are all present in leakage traces such as in Figure 1. Recently, Duc et al. showed that security against probing attacks implies security against noisy leakages (up to a factor  $|\mathbb{F}|$ , where  $\mathbb{F}$  is the underlying field in which the operations are carried out) [27]. In the rest of this section, we first connect the statistical distance SD with the mutual information metric MI, which shows that both can be used to quantify the noise condition required for masking. Next, we provide alternative forms for the theorems of Duc et al. and show (i) the security of the encoding used in (e.g. Boolean) masking and (ii) the security of a complete circuit based on the ISW compiler.

#### 3.1 From statistical distance to MI

The results from Duc et al. require to have a bound on the SD between the shares and the shares given the leakage. For different reasons, expressing this distance based on the MI metric may be more convenient in practice (as witnessed by the numerous works where this metric has been computed, for various types of devices, countermeasures and technologies – see the list in introduction). For example, and as previously mentioned, the MI metric is useful to determine whether the leakage model used in a standard DPA is sound [28]. Very concretely, Equations (3) and (4) are also expressed in a way that requires summing over the intermediate values first and on the leakages afterwards, which corresponds

to the way security evaluations are performed (i.e. fix the target device’s state, and then perform measurements). Thus, we now show how to express the SD in function of the MI. We use a previous result from Dodis [26] for this purpose, which proofs follows [12] that we rephrase with our notations.

**Lemma 1 ([26], Lemma 6).** *Let  $Y_i$  and  $\mathbf{L}_{Y_i}$  be two random variables. Then:*

$$\frac{1}{2} \left( \sum_{(y \in \mathcal{Y}, \ell \in \mathcal{L})} |\Pr[Y_i = y, \mathbf{L}_{Y_i} = \ell] - \Pr[Y_i = y] \Pr[\mathbf{L}_{Y_i} = \ell]| \right)^2 \leq \text{MI}(Y_i; \mathbf{L}_{Y_i}) .$$

Using this lemma, we can now express the SD in function of the MI as follows.

**Theorem 1.** *Let  $Y_i$  and  $\mathbf{L}_{Y_i}$  be two random variables. Then:*

$$2 \cdot \text{SD}(Y_i; Y_i \mid \mathbf{L}_{Y_i})^2 \leq \text{MI}(Y_i; \mathbf{L}_{Y_i}) .$$

*Proof.* The proof follows the proof of [11], Lemma 4.4. We have:

$$\begin{aligned} & \sum_{(y \in \mathcal{Y}, \ell \in \mathcal{L})} |\Pr[Y_i = y, \mathbf{L}_{Y_i} = \ell] - \Pr[Y_i = y] \Pr[\mathbf{L}_{Y_i} = \ell]| , \\ &= \sum_{\ell \in \mathcal{L}} \Pr[\mathbf{L}_{Y_i} = \ell] \sum_{y \in \mathcal{Y}} |\Pr[Y_i = y \mid \mathbf{L}_{Y_i} = \ell] - \Pr[Y_i = y]| , \\ &= 2 \cdot \text{SD}(Y_i; Y_i \mid \mathbf{L}_{Y_i}) . \end{aligned}$$

The final result directly derives from Lemma 1. □

### 3.2 Security of the encoding

In this section, we analyze the security of an encoding when  $m$  measurements are performed and the encoding is refreshed between each measurement using a leak-free gate. More precisely, we assume that a secret  $y$  is secret-shared into  $d$  shares  $y_1, \dots, y_d$ , using an additive masking scheme over a finite field  $\mathbb{F}$ . Between each measurement, we assume that we take fresh  $y_1, \dots, y_d$  values such that  $y = y_1 + \dots + y_d$  (e.g. it could be the Boolean encoding of Section 2.1). We also assume that this refreshing process does not leak and first recall a previous result from [27] that relates the random probing model to the noisy model. For conciseness, we call an adversary in the random-probing model a “random-probing adversary”, an adversary in the  $\delta$ -noisy model a “ $\delta$ -noisy adversary”, and an adversary having access to leakages such that  $\text{MI}(Y; \mathbf{L}_Y) \leq \delta$  a “ $\delta$ -MI-adversary”. However, note that the physical noise (and its quantification with the MI) is a property of the implementation rather than of the adversary.

**Lemma 2 ([27], Lemma 3).** *Let  $\mathcal{A}$  be a  $\delta$ -noisy adversary targeting values in  $\mathbb{F}^d$ . Then, there is a  $\delta \cdot |\mathbb{F}|$ -random-probing adversary  $\mathcal{S}$  on  $\mathbb{F}^d$  such that for every  $(y_1, \dots, y_d)$ ,  $\mathcal{A}$  and  $\mathcal{S}$  produce the same view when applied on  $(y_1, \dots, y_d)$ .*

This result enables us to work directly in the random-probing model instead of the noisy leakage model. Next, we study the security of the encoding. As mentioned in introduction, the adversary’s goal in this case is to recover the encoded value, which is equivalent to key recovery if this value is a key. In order to make it completely comparable with actual attacks, we also add the number of measurements  $m$  used by the adversary as a parameter in our bounds.

**Theorem 2.** *Let  $d$  be the number of shares used for a key encoding,  $m$  be the number of measurements, and  $\text{MI}(Y_i, \mathbf{L}_{Y_i}) \leq t$  for some  $t \leq 2/|\mathbb{F}|^2$ . Then, if we refresh the encoding in a leak-free manner between each measurement, the probability of success of a key recovery adversary under independent leakage is:*

$$\text{SR}^{\text{kr}} \leq 1 - \left(1 - \left(|\mathbb{F}| \sqrt{t/2}\right)^d\right)^m. \quad (8)$$

*Proof.* In the random probing model with parameter  $\epsilon$ , an adversary learns nothing about the secret if there is at least one share that did not leak. Since all the measurements are independent and we use leak-free refreshing gates, we have:

$$\text{SR}^{\text{kr}} \leq 1 - (1 - \epsilon^d)^m. \quad (9)$$

Let  $\mathbf{A}$  be a  $t$ -MI-adversary on  $\mathbb{F}^d$ . From Theorem 1, we know that  $\mathbf{A}$  implies a  $\sqrt{t/2}$ -noisy-adversary on  $\mathbb{F}^d$  and, by Lemma 2, we obtain a  $|\mathbb{F}| \sqrt{t/2}$ -random-probing adversary on  $\mathbb{F}^d$ . Letting  $\epsilon := |\mathbb{F}| \sqrt{t/2}$  in (9) gives us the result.  $\square$

Note that Equation (9) focuses on the impact of the adversary’s measurement complexity  $m$  on the success rate, which is usually the dominating factor in concrete side-channel analyses. However, the impact of time complexity when considering key enumeration, which is the standard way to exploit computational power in side-channel analysis [75, 76], will be discussed in Section 4.3. Besides, and for readability, we ignore the additional terms corresponding to mathematical cryptanalysis (e.g. exhaustive search, linear cryptanalysis, ...) that should be added for completeness. In order to allow us comparing this result with the case where we study the security of a complete circuit encoded with the ISW compiler, we also write our result according to the following corollary.

**Corollary 1.** *Let  $d$  be the number of shares used for a key encoding and  $m$  the number of measurements. Then, if we refresh the encoding in leak-free manner between each measurement and for any  $\alpha > 0$ , the probability of success of a key recovery adversary under independent leakage is:*

$$\text{SR}^{\text{kr}} \leq m \cdot \exp(-\alpha d), \quad (10)$$

*if we have:*

$$\text{MI}(Y_i; \mathbf{L}_{Y_i}) \leq 2 \left( \frac{1}{e^\alpha |\mathbb{F}|} \right)^2. \quad (11)$$

*Proof.* We have:

$$1 - (1 - \epsilon^d)^m \leq m \epsilon^{\log(\epsilon)d}.$$

We want  $\log(\epsilon) = -\alpha$ . Hence, from Theorem 2, we get our result.  $\square$

### 3.3 Security of the whole circuit

In this section, we restate the theorems from Duc et al. when securing a whole circuit with the seminal ISW compiler. The main theorem from [27] bounds the probability of success of a distinguishing adversary in the noisy leakage model. We provide an alternative version of their theorem and, as in the previous section, we relate it to the mutual information instead of the statistical distance.

**Theorem 3.** *Suppose that we have a circuit of size  $|\Gamma|$  (i.e., with  $|\Gamma|$  gates) protected with the ISW compiler with  $d$  shares. Then, the probability of success of a distinguishing adversary under independent leakage is:*

$$\text{SR}^{\text{dist}} \leq |\Gamma| \cdot \exp\left(-\frac{d}{12}\right) = |\Gamma| \cdot 2^{\left(-\frac{d \cdot \log_2(e)}{12}\right)} \leq |\Gamma| \cdot 2^{-d/9}, \quad (12)$$

if we have:

$$\text{MI}(Y_i; \mathbf{L}_{Y_i}) \leq 2 \cdot \left(\frac{1}{|F| \cdot (28d + 16)}\right)^2. \quad (13)$$

The relation between the circuit size and concrete multivariate (aka horizontal) side-channel attacks such as [9, 38] is discussed in Section 4.1, Paragraph (c), together with its impact regarding composability issues [23, 6]. Similarly to what we did in the previous section, we also write the following corollary.

**Corollary 2.** *Suppose that we have a circuit of size  $|\Gamma|$  protected with the ISW compiler with  $d$  shares. Then, if  $\text{MI}(Y_i, \mathbf{L}_{Y_i}) \leq t$ , a distinguisher adversary under independent leakage needs:*

$$d \geq \frac{1 - 16|F|\sqrt{\frac{1}{2}t}}{28|F|\sqrt{\frac{1}{2}t}} \quad (14)$$

shares in order to obtain:

$$\text{SR}^{\text{dist}} \leq |\Gamma| \cdot \exp\left(-\frac{d}{12}\right) \leq |\Gamma| \cdot \exp\left(-\frac{1 - 16|F|\sqrt{\frac{1}{2}t}}{336|F|\sqrt{\frac{1}{2}t}}\right). \quad (15)$$

Note that the ISW compiler can actually be used to efficiently compute any circuit. For example, the work of Rivain and Prouff at CHES 2010 showed how to adapt the compiler to  $|F| = 256$  which leads to efficient masked implementations of the AES [66] (see also various following works such as [17, 37, 67]).

## 4 Making proofs concrete: practice

In this section, we complement the previous theoretical results with an experimental analysis. Our contributions are threefold. First, we provide an empirical evaluation of the encoding scheme in Section 3.2, which allows us to discuss the noise condition and tightness of the bounds in our proofs. We use this discussion to conjecture a simple connection between the mutual information metric and the success rate of a (worst-case) side-channel adversary, and argue that it can lead

to quite accurate approximations of the attacks’ measurement complexity. Next, we discuss possible deviations from the independent leakage assumption and provide tools allowing one to approximate the security level of concrete devices in such cases. Eventually, we consider the tradeoff between measurement complexity and time complexity in the context of divide-and-conquer side-channel attacks. We show how one can build a side-channel security graph (i.e. a plot of the adversary’s success probability bounds in function of both parameters [76]), based only on the estimation of the MI metric for each share of a masking scheme. Along these lines, we additionally provide a formal justification for the physical security evaluation framework proposed at Eurocrypt 2009 [71].

#### 4.1 Experimental validation

In order to discuss the relevance of the proofs in the previous section, we take the (usual) context of standard DPA attacks defined in [48]. More precisely, we consider the simple case where an adversary targets a single S-box from a block cipher (e.g. the AES) as specified in Section 2.1, and obtains leakage variables  $\mathbf{L}_{y_i} = \mathbf{L}(y_i, \mathbf{R}_i)$  for  $1 \leq i \leq d$  (the case of multiple S-boxes will be studied in Section 4.3). For convenience, we mainly consider the context of mathematically-generated Gaussian Hamming weight leakages, where  $\mathbf{L}_{y_i} = \text{HW}(y_i) + N_i$ , with  $\text{HW}$  the Hamming weight function and  $N_i$  a Gaussian-distributed noise, with variance  $\sigma^2$ . In this respect, we note that we did not mount concrete attacks since we would have had to measure hundreds of different implementations to observe useful trends in practice. Our experiments indeed correspond to hundreds of different noise levels. Yet, we note that devices that exhibit close to Hamming weight leakages are frequently encountered in practice [47]. Furthermore, such a simulated setting is a well established tool to analyze masking schemes (see, e.g. [67] for polynomial masking, [4] for inner product masking and [16] for leakage squeezing). Besides, we also consider random Gaussian leakage functions, of which the deterministic part corresponds to random functions over  $\mathcal{Y}$ , to confirm that all the trends we put forward are also observed with leakage functions that radically differ from the usual Hamming weight one.

**a. Computing the MI metric.** In this DPA setting, we aim to compute the MI between the key and the plaintext and leakages. For conciseness, we use the notations  $\bar{Y} = [Y_1, \dots, Y_d]$  and  $\bar{\mathbf{L}}_Y = [\mathbf{L}_{Y_1}, \dots, \mathbf{L}_{Y_d}]$  for vectors containing the  $d$  shares and their corresponding leakages. Then we compute:

$$\begin{aligned} \text{MI}(K; X, \bar{\mathbf{L}}_Y) &= \text{H}[K] + \sum_{k \in \mathcal{K}} \text{Pr}[k] \cdot \\ &\sum_{x \in \mathcal{X}, \bar{y} \in \mathcal{Y}^d} \text{Pr}[x, \bar{y}] \cdot \sum_{\bar{\mathbf{l}}_y \in \mathcal{L}^d} \text{Pr}[\bar{\mathbf{l}}_y | k, x, \bar{y}] \cdot \log_2 \text{Pr}[k | x, \bar{\mathbf{l}}_y]. \end{aligned} \quad (16)$$

While this expression may look quite involved, we note that it is actually simple to estimate in practice, by sampling the target implementation. Evaluators just have to set keys  $k$  in their device and generate leakage traces corresponding to

(known) plaintexts  $x$  and (unknown) shares  $\bar{y}$ . Say there are  $|\mathcal{K}| = n_k$  key candidates and we generate  $n_t$  leakage traces  $\bar{l}_i$ , then, one just assigns probabilities  $\hat{p}_i^j$  to each key candidate  $k_j^*$ , for each measured trace, as in Table 1. This is typically done using TA or LR. Following, if the correct key candidate is  $k$ , the second line of (16) can be computed as  $\bar{E}_i \log_2(\hat{p}_i^k)$ . Note that whenever considering the standard DPA setting where the target operations follow a key addition, it is not even necessary to sum over the keys since  $\text{MI}(K = k; X, \bar{L}_Y)$  is identical for all  $k$ 's, thanks to the key equivalence property put forward in [48].

**Table 1.** Computing key candidate probabilities for MI metric estimation.

State & leakage	Key candidates			
	$k_1^*$	$k_2^*$	$\dots$	$k_{N_k}^*$
$(k, x_1) \rightsquigarrow \bar{l}_1$	$\hat{p}_1^1$	$\hat{p}_1^2$	$\dots$	$\hat{p}_1^{n_k}$
$(k, x_2) \rightsquigarrow \bar{l}_2$	$\hat{p}_2^1$	$\hat{p}_2^2$	$\dots$	$\hat{p}_2^{n_k}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$(k, x_{n_t}) \rightsquigarrow \bar{l}_{n_t}$	$\hat{p}_{n_t}^1$	$\hat{p}_{n_t}^2$	$\dots$	$\hat{p}_{n_t}^{n_k}$

Intuitively,  $\text{MI}(K; X, \bar{L}_Y)$  measures the amount of information leaked on the key variable  $K$ . The framework in [71] additionally defines a Mutual Information Matrix (MIM) that captures the correlation between any key  $k$  and the key candidates  $k^*$ . Using our sampling notations, the elements of this matrix corresponds to  $\text{MIM}_{k,k^*} = H[K] + \mathbf{E}_i \log_2(\hat{p}_i^{k^*})$ . More formally:

**Definition 2 (Mutual Information Matrix (MIM)).** *For a random variable  $K$ , we define the  $|\mathcal{K}| \times |\mathcal{K}|$  mutual information matrix (MIM) such that to each key  $k$  and key candidate  $k^*$ , we associate the value:*

$$\text{MIM}_{k,k^*} = H[K] + \sum_{x \in \mathcal{X}, \bar{y} \in \mathcal{Y}^d} \Pr[x, \bar{y}] \cdot \sum_{\bar{l}_y \in \mathcal{L}^d} \Pr[\bar{l}_y | k, x, \bar{y}] \cdot \log \Pr[k^* | x, \bar{l}_y]. \quad (17)$$

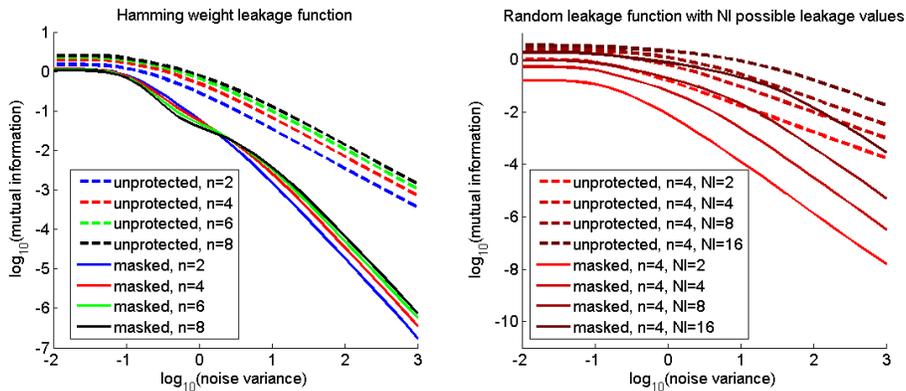
This definition directly leads to the equality:  $\text{MI}(K; X, \bar{L}_Y) = \mathbf{E}_k(\text{MIM}_{k,k})$ , i.e. the mutual information is the average value of the diagonal elements of MIM.

**b. Intuition behind the noise condition.** Theorems 2 and 3 both require that the MI between the shares and their corresponding leakage is sufficiently small. In other words, they require the noise to be sufficiently large. In this section, we compute the MI metric for both an unprotected implementation (i.e.  $d = 1$ ) and a masked one (i.e.  $d = 2$ ) in function of different parameters.<sup>1</sup> In order to illustrate the computation of this metric, we provide a simple open source code that evaluates the MI between a sensitive variable  $Y$  and its Hamming weights, for different noise levels, both via numerical integration (that is only possible for mathematically-generated leakages) and sampling (that is more reflective of the evaluation of an actual device) [1]. In the latter case, an evaluator additionally

<sup>1</sup> For the masked case, we consider univariate leakages corresponding to the parallel setting in [10], for which computing the MI is slightly faster than in the serial one.

has to make sure that his estimations are accurate enough. Tools for ensuring this condition are discussed in [28]. In the following, this sufficient sampling is informally confirmed by the smooth shape of our experimental curves.

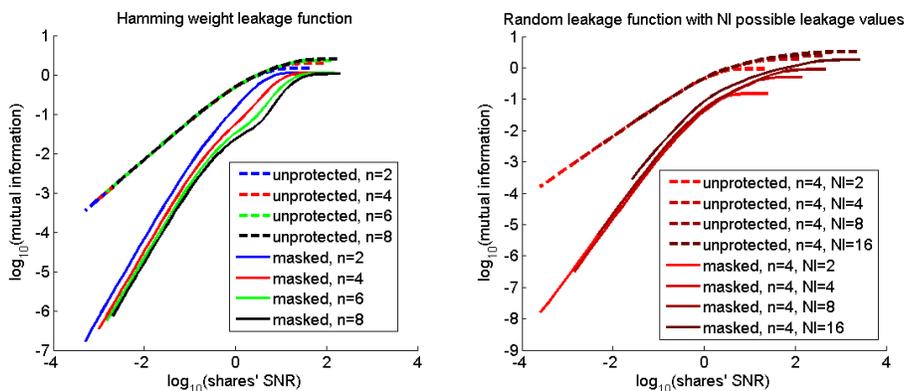
We start with the simplest possible plot, where the MI metric is computed in function of the noise variance  $\sigma^2$ . Figure 2 shows these quantities, both for Hamming weight leakage functions and for random ones with output range  $N_l$  (in the latter context, the functions for different  $N_l$ 's were randomly picked up prior to the experiments, and stable across experiments). We also considered different bit sizes ( $n = 2, 4, 6, 8$ ). Positively, we see that in all cases, the curves reach a linear behavior, where the slope corresponds to the number of shares  $d$ . Since the independent leakage condition is fulfilled in these experiments, this  $d$  corresponds to the smallest key-dependent moment in the leakage distribution. And since the measurement (aka sampling) cost for estimating such moments is proportional to  $(\sigma^2)^d$ , we observe that the MI decreases exponentially in  $d$  for large enough noises. Note that this behavior is plotted for  $d = 1, 2$ , but was experimented for  $d$ 's up to 4 in [72], and in fact holds for any  $d$ , since it exactly corresponds to Theorem 2 in a context where its assumptions are fulfilled.



**Fig. 2.** MI metric in function of  $\sigma^2$ . HW (left) and random (right) leakages.

Negatively, we also see that the noise level that can be considered as high enough depends on the leakage functions. For example, the random leakage functions in the right part of the figure have signals that vary from approximately  $\frac{2}{4}$  for  $N_l = 2$  to  $\frac{16}{4}$  for  $N_l = 16$ . It implies that the linearly decreasing part of the curves is reached for larger noises in the latter case. Yet, this observation in fact nicely captures the intuition behind the noise condition. That is, the noise should be high enough for hiding the signal. Therefore, a very convenient way to express it is to plot the MI metric in function of shares' SNR, as in Figure 3. Here, we clearly see that as soon as the SNR is below a certain constant ( $10^{-1}$ , typically), the shape of the MI curves gets close to linear. This corroborates the condition in Theorem 2 that masking requires  $\text{MI}(K_i; X, \mathbf{L}_{Y_i})$  to be smaller than

a given constant. Our experiments with different bit sizes also suggest that the  $|\mathbb{F}|$  factor in this noise condition is a proof artifact. This is now formally proven by Dziembowski, Faust and Skorski in [30]. Of course, and as mentioned in Section 2.2, the SNR metric is only applicable under certain conditions (univariate Gaussian leakages). So concretely, an evaluator may choose between computing it after dimensionality reduction (leading to a heuristic but intuitive condition), or to directly state the condition in function of the MI. For completeness, we also plot the MI metric for an unprotected and masked implementation in function of the share’s MI in Appendix, Figure 11. It clearly exhibits that as the share’s MI decreases, this reduction is amplified by masking (exponentially in  $d$ ).



**Fig. 3.** MI metric in fct. of the shares’ SNR. HW (left) and random (right) leakages.

**c. Tightness of the bounds.** Given that the noise is high enough (as just discussed), Theorems 2 and 3 guarantee that the success rate of a side-channel adversary can be bounded based on the value of the share’s leakage, measured with  $\text{MI}(K_i; X, \mathbf{L}_{Y_i})$ . This directly leads to useful bounds on the measurement complexity to reach a given success rate, e.g. from (8) we can compute:

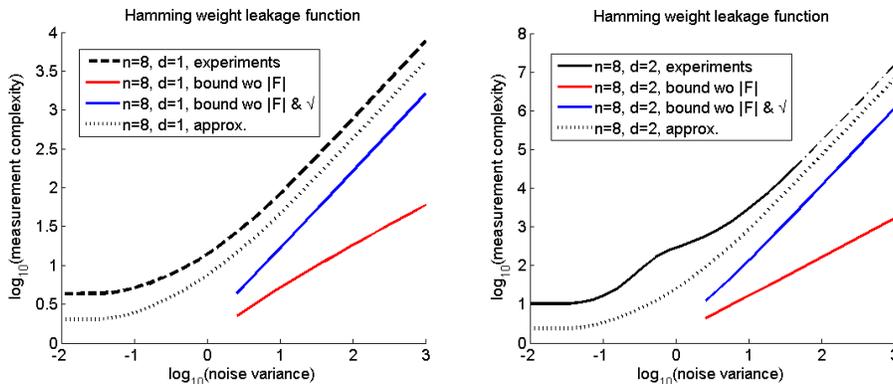
$$m \geq \frac{\log(1 - \text{SR}^{\text{kr}})}{\log\left(1 - \left(|\mathbb{F}| \sqrt{\frac{\text{MI}(K_i; X, \mathbf{L}_{Y_i})}{2}}\right)^d\right)}. \quad (18)$$

Note that in our standard DPA experiments where we consider bijective S-boxes<sup>2</sup>, we have that  $\text{MI}(K_i; X, \mathbf{L}_{Y_i})$  simplifies into  $\text{MI}(Y_i; \mathbf{L}_{Y_i})$ , i.e. the security only depends on the leakage of the target intermediate variables  $Y_i$ ’s. We now want to investigate how tight this bound is. For this purpose, we compared it with the measurement complexity of concrete key recovery TA (using a perfect

<sup>2</sup> Our attacks exploit the leakages of an S-box output, as specified in Section 2.1. We took the PRESENT S-box for  $n = 4$ , the AES one for  $n = 8$ , and picked up two random S-boxes for  $n = 2, 6$ , as we did for the random leakage functions.

leakage model). As previously mentioned, the  $|\mathbb{F}|$  factor in this equation can be seen as a proof artifact related to the reduction in our theorems – so we tested a bound excluding this (possibly significant) factor (e.g. it would typically be equal to 256 in the AES case). For similar reasons, we also tested a bound additionally excluding the square root loss in the reductions (coming from Theorem 1).

As illustrated in Figure 4, the measurement complexity of the attacks is indeed bounded by Equation (18), and removing the square root loss allows the experimental and theoretical curves to have similar slopes. The latter observation fits with the upper bound  $\text{MI}(Y_i; \mathbf{L}_{Y_i}) \leq \frac{|\mathbb{F}|}{\ln(2)} \cdot \text{SD}(Y_i; Y_i | \mathbf{L}_{Y_i})$  given in [60] that becomes tight as the noise increases.<sup>3</sup> As expected, the bounds become meaningless for too low noise levels (or too large SNRs, see Appendix, Figure 12). Intuitively, this is because we reach success rates that are stuck to one when we deviate from this condition. For completeness, we added approximations obtained by normalizing the shares’ MI by  $H[K]$  to the figure<sup>4</sup>, which provide hints about the behavior of a leaking device when the noise is too low.



**Fig. 4.** Measurement complexity and bounds/approximations for concrete TA.

Interestingly, these results also allow us to reach a comprehensive view of the parameters in Theorem 3, where the security of a complete circuit encoded according to the ISW compiler is proven. That is, in this case as well we expect the  $|\mathbb{F}|$  and  $1/9$  factors in Equation (12) to be due to proof technicalities. By contrast, the  $|T|$  factor is physically motivated, since it corresponds to the size of the circuit and fits the intuition that more computations inevitably means more exploitable leakage. The  $d$  factor appearing in the noise condition of Equation (13) can also be explained, since it directly relates to the fact that in the ISW compiler, any multiplication will require to manipulate each share  $d$  times.

<sup>3</sup> Since their inequality comes from a  $\log_2(1+x) \leq \frac{x}{\ln(2)}$  inequality that gets close to an equality when  $x$  gets close to 0, which happens for large noise levels.

<sup>4</sup> And multiplying the measurement complexity by a constant: see Equation (21).

They typically reflect the distance between simple side-channel attacks that only target an encoding (such as analyzed in this section) and more powerful (highly multivariate) adversaries trying to exploit all the leakage samples that depend on each share’s manipulation, that we discuss at the end of this paragraph. Taking all these observations into account, we summarize the concrete security provided by any masking scheme with the following informal conjecture.

**Informal conjecture.** *Suppose that we have a circuit of size  $|\Gamma|$  masked with  $d$  shares such that the information leakage on each of these shares (using all available time samples) is bounded by  $\text{MI}(Y_i; \mathbf{L}_{Y_i})$ . Then, the probability of success of a distinguishing adversary using  $m$  measurements and targeting a single element (e.g. gate) of the circuit under independent and sufficiently noisy leakage is:*

$$\text{SR}_1^{\text{dist}} \leq 1 - (1 - \text{MI}(Y_i; \mathbf{L}_{Y_i})^d)^m, \quad (19)$$

and the probability of success targeting all  $|\Gamma|$  elements independently equals:

$$\text{SR}_{|\Gamma|}^{\text{dist}} \leq 1 - (1 - \text{SR}_1^{\text{dist}})^{|\Gamma|}. \quad (20)$$

Note that Equation (19) is backed up by the results in [3] (Theorem 6) where a similar bound is given in the context of statistical cryptanalysis. By additionally using the approximation  $\log(1 - x) \approx -x$  that holds for  $x$ ’s close to 0, Equation (19) directly leads to the following simple approximation of a standard DPA’s measurement complexity for large noise levels:

$$m \geq \frac{\log(1 - \text{SR}_1^{\text{dist}})}{\log(1 - \text{MI}(Y_i; \mathbf{L}_{Y_i})^d)} \approx \frac{cst}{\text{MI}(Y_i; \mathbf{L}_{Y_i})^d}, \quad (21)$$

where  $cst$  is a small constant that depends on the target success rate.

In this conjecture, the words “*using all the available time samples*” refer to the fact that in order to evaluate the (worst-case) information of the shares, one should exploit multivariate attacks or dimensionality reduction techniques (e.g. such as PCA, LDA, KDA), as mentioned in Section 2.1, Paragraph (a).

Besides, Equation (20) (like Theorem 3) assumes that the leakages of the  $|\Gamma|$  gates (or target intermediate values) are exploited independently. This perfectly corresponds to the probing model in which the adversary gains either full knowledge or no knowledge of such computing elements. However, side-channel attacks against concrete implementations can sometimes take advantage of different computations jointly in a more efficient manner. For example, standard DPA can optimally combine multiple operations depending on the same (enumerable) part of key [48, 51], and algebraic/analytical side-channel attacks can heuristically exploit all the information leakages of an implementation (see [63, 77] and follow ups). The latter multivariate (aka horizontal) attacks can be very powerful in the context of masking (e.g. whenever it is possible to average the shares’ leakages before the estimation of a higher-order statistical moment) [9, 38]. Eventually, we recall that the protection of large circuits additionally needs to ensure composability (e.g. as recently formalized in [6]). That is, a sufficient amount of refreshing gadgets is required to prevent attacks such as [23].

**d. Relation with the Eurocrypt 2009 evaluation framework.** The evaluation of leaking cryptographic implementations with a combination of information and security metrics was put forward by Standaert et al. at Eurocrypt 2009 [71]. In this reference, the authors showed a qualitative connection between both metrics. Namely, they proved that the model (i.e. the approximation of the leakage PDF) used by a side-channel adversary is sound (i.e. allows key recoveries) if and only if the mutual information matrix (defined in Paragraph (a) of this section) is such that its diagonal values are maximum for each line. By contrast, they left the quantitative connection between these metrics as an open problem (i.e. does more MI imply less security?). Our results provide a formal foundation for this quantitative connection. They prove that for any implementation, decreasing the MI of the target intermediate values is beneficial to security. This can be achieved by ad hoc countermeasures, in which case it is the goal of an evaluation laboratory to quantify the MI metric, or by masking, in which case we can bound security based only on the value of this metric for each share taken separately (of course assuming that the independent leakage assumption holds to a sufficient extent, as more carefully discussed in the next section).

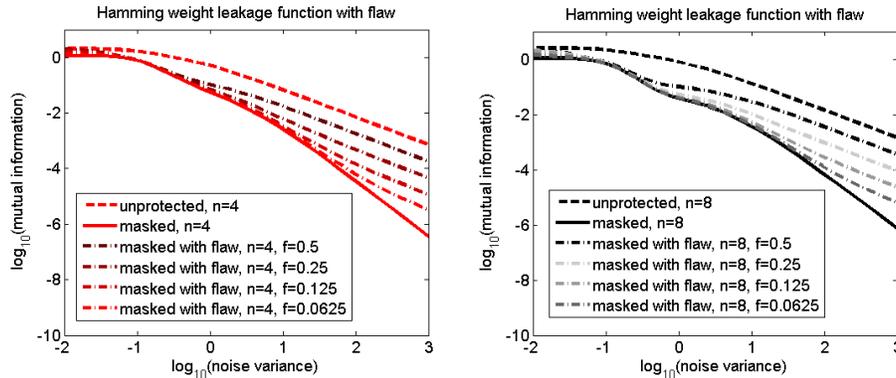
## 4.2 Beyond independent leakage

The previous section evaluated an experimental setting where the leakage of each share is independent of each other, i.e.  $L_{y_i} = G(y_i) + N_i$ . But as discussed in introduction, this condition frequently turns out to be hard to fulfill and so far, there are only limited (even informal) tools allowing to analyze the deviations from independent leakages that may be observed in practice. In order to contribute to this topic, we first launched another set of experiments (for 2-share masking), where the leakage of each share can be written as:

$$\begin{aligned} L_{y_1} &= G_1(y_1) + f \cdot G_{1,2}(y_1, y_2) + N_1 , \\ L_{y_2} &= G_2(y_2) + f \cdot G_{2,1}(y_1, y_2) + N_2 . \end{aligned}$$

Here the  $G_i$  functions manipulate the shares independently, while the  $G_{i,j}$  functions depend on both shares. We additionally used the  $f$  (for flaw) parameter in order to specify how strongly we deviate from the independent leakage assumption. As in the previous section, we considered Hamming weight and random functions for all  $G$ 's (and we used  $G_{i,j}(y_i, y_j) = G(y_i \oplus y_j)$  for illustration). Exemplary results of an information theoretic analysis in this context are given in Figure 5 for the  $n = 4$ -, and 8-bit cases (and in Appendix, Figure 13 for the  $n = 2$ - and 6-bit S-box cases). We mainly observe that as the noise increases, even small flaws are exploitable by an adversary. Indeed, breaking the independence condition makes smaller-order moments of the leakage distribution key-dependent. Consequently, for large enough noise, it is always this smaller-order moment that will be the most informative. This is empirically confirmed by the slopes of the IT curves in the figures, that gradually reach one rather than two.

Following these experiments, let us consider a chip that concretely exhibits such a flaw for a given noise level  $\sigma_{\text{exp}}^2$  (corresponding to its actual measurements). Despite falling outside the masking proofs' guarantees, an important

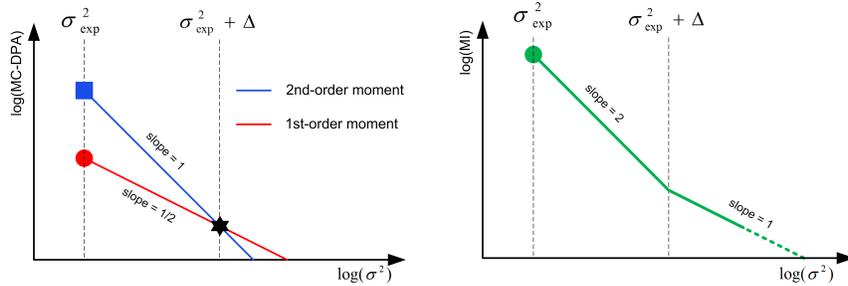


**Fig. 5.** MI metric for masked implementation with flaw ( $n = 4, 8$ ).

question is whether we can still (approximatively) predict its security level based on sound statistical tools. In this respect, a useful observation is that the MI metric cannot directly answer the question since it captures the information lying in all the statistical moments of the leakage PDF. So we need another ingredient in order to reveal the informativeness of each moment of the leakage PDF, separately. The Moments-Correlating DPA (MC-DPA) recently introduced in [54] is a natural candidate for this purpose. We now describe how it can be used to (informally) analyze the security of a flawed masked implementation.

In this context, we first need to launch MC-DPA for different statistical moments, e.g. the first- and second-order ones in our 2-share example. They are illustrated by the circle and square markers in the left part of Figure 6. For concreteness, we take the (most revealing) case where the second-order moment is more informative than the first-order one. Assuming that the noise condition in our theorems is fulfilled, the impact of increasing the noise on the value of the MC-DPA distinguisher can be predicted as indicated by the curves of the figure. That is, with a slope of  $1/2$  for the first-order moment and a slope of 1 for the second-order one.<sup>4</sup> Hence, we can directly predict the noise level  $\sigma_{\text{exp}}^2 + \Delta$  such that the first-order moment becomes more informative. Eventually, we just observe that concrete side-channel attacks always exploit the smallest key-dependent moment in priority (which motivates the definition of the security-order for masking schemes [22]). So starting from the value of the MI at  $\sigma_{\text{exp}}^2$  (represented by a circle in the right part of the figure), we can extrapolate that this MI will decrease following a curve with slope 2 until  $\sigma_{\text{exp}}^2 + \Delta$  and a curve with slope 1 afterwards. Taking advantage of the theorems in the previous sections, this directly leads to approximations of the best attacks' measurement complexity. Furthermore, extending this reasoning to more shares and higher-

<sup>4</sup> Slopes are divided by 2 when considering Pearson's correlation rather than the MI since this correlation is essentially proportional to the square root of the SNR. This is also reflected by the measurement complexity of CPA, that is proportional to the inverse of the squared correlation vs. the inverse of the MI for TA [73].



**Fig. 6.** Evaluating non-independent leakages with MC-DPA (left) and MI (right).

order statistical moments is straightforward: it just requires to add MC-DPA curves in the left part of Figure 6, and to always consider the one leading to the highest MC-DPA value to set the slope of the MI curves, in the right part of the figure. To the best of our knowledge, such figures (despite informal) provide the first concrete tools to approximate the security level in such contexts.

Note finally that the shape of the non-independent leakages (i.e. the  $G_{i,j}$  functions) observed in practice highly depends on the implementations. For example in software, the most usual issue (due to transition-based leakages, which actually corresponds to our exemplary function  $G_{i,j}(y_i, y_j) = G(y_i \oplus y_j)$ ) is easy to analyse [5]. It typically divides the order of the smallest key-dependent moment in the leakage distribution by two, which corresponds to the additional square root loss in the security bounds of Duc et al. when considering leakages that depend on two wires simultaneously (see [27], Section 5.5). By contrast in hardware, multiple shares can leak jointly in a hardly predictable manner, for example due to glitches [49] or couplings [20]. Yet, even in this case (and in general) the important conclusion of this section is that (independence) flaws due to physical defaults are always relative to noise. For example, in the latter reference about couplings, we have that for the noise level of the authors' measurement board, higher-order leakages are more informative than lower-order ones, just as analyzed in Figure 6. So in this case, and up to noise levels such that lower-order leakages become more informative, we can conclude that the (coupling) flaw does not reduce the concrete security level of their masked implementation.

### 4.3 Exploiting computational power

In this section, we finally tackle the problem of divide-and-conquer DPA attacks, where the adversary aims to combine side-channel information gathered from a number of measurements and computational power. That is, how to deal with the practically critical situation where the number of measurements available is not sufficient to exactly recover the key? As discussed in [75, 76], optimal enumeration and key ranking algorithms provide a concrete answer to this question. They allow building security graphs, where the success rate is plotted in function of a number of measurements and computing power, by repeating attacks multiple times. We next discuss more efficient and analytical strategies.

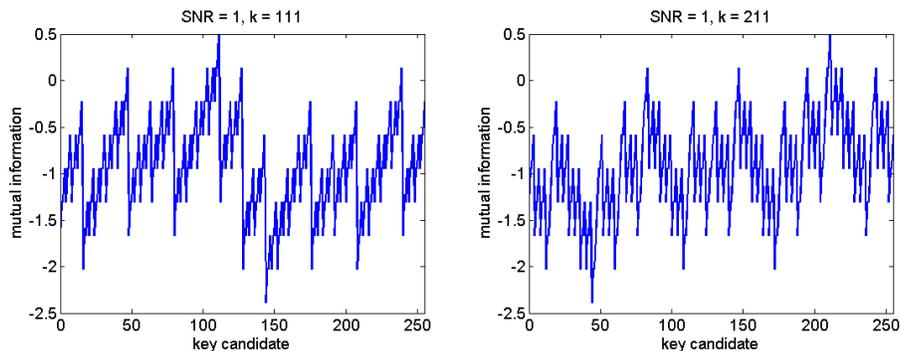


Fig. 7. Exemplary lines of the mutual information matrix  $MIM_{k,-}$  (low noise).

**a. Why MI is not enough?** Whenever trying to exploit both side-channel leakage and brute-force computation (e.g. key enumeration) the most challenging aspect of the problem is to capture how measurements and computation actually combine. This is easily illustrated with the following example. Imagine two hypothetical side-channel attacks that both succeed with probability  $1/100$ . In the first case, the adversary gains nothing with probability  $99/100$  and the full key with probability  $1/100$ . In the second case, he always gains a set of 100 equally likely keys. Clearly, enumeration will be pretty useless in the first case, while extremely powerful in the second one. More generally, such examples essentially suggest that the computational cost of an enumeration does not only depend on the informativeness of the leakage function (e.g. measured with the MI) but also on its shape. For illustration, a line of the mutual information matrix computed from Hamming weight leakages for two different values of  $k$  is given in Figure 7, where we can clearly identify the patterns due to this leakage model (i.e. close key candidates  $k^*$  are those for which the Hamming distance  $HW(k \oplus k^*)$  is low). Similar plots for a larger noise are given in Appendix, Figure 14. While  $MIM_{k,k}$  only corresponds to a single value of the matrix line (here for  $k = 111$  and  $k = 211$ ), which bounds the measurement complexity to recover the corresponding key without additional computation (as previously discussed), how helpful enumeration will additionally depend on the relative distance between the  $MIM_{k,k}$  and  $MIM_{k,k^*}$  values [80]. Therefore, this example incidentally puts forward some limitations of the probing leakage model when measuring computational cost, since it describes an all-or-nothing strategy – as already mentioned in Section 4.1, Paragraph (c) – which is not the case for the noisy leakage setting. Hence, whereas the probing model is easier to manipulate in proofs, and therefore useful to obtain asymptotic results, noisy leakages are a more accurate tool to quantify concrete security levels as in this section.

**b. Measurement and computational bounds per S-box.** Interestingly, one can easily derive heuristic bounds for attacks combining side-channel measurements and enumeration power against a single S-box, by re-using the same material as we anyway need to estimate the MI metric for a single secret share.

For this purpose, the main idea is to define a new “aggregated key variable”  $K_{\text{agg}}^c$  such that each time a leakage  $\mathbf{l}_y$  is observed for an intermediate value  $y$ , the probability of the aggregated key corresponds to the probability of the  $c$  most likely candidates  $y^*$ . Concretely, this first requires to characterize the distance between any intermediate candidate  $y$  and its close candidates  $y^*$ , which can be done by computing a MIM for the random variable  $Y$ , defined as:

$$\text{MIM}_{y,y^*} = H[Y] + \sum_{\bar{y} \in \mathcal{Y}^d} \Pr[\bar{y}] \cdot \sum_{\bar{\mathbf{l}}_y \in \mathcal{L}^d} \Pr[\bar{\mathbf{l}}_y | \bar{y}] \cdot \log \Pr[y^* | \bar{\mathbf{l}}_y]. \quad (22)$$

The latter can be computed as explained in Section 4.1, Paragraph (a), Equation (17). We then sort its lines in order to obtain vectors  $s_y = \text{sort}(\text{MIM}_{y,-})$ .<sup>5</sup> We further denote with  $k_{s_y(1)}^*$  the key candidate giving rise to the correct  $y$  value,  $k_{s_y(2)}^*$  the key candidate giving rise to the  $y^*$  candidate that is the closest to  $y$ ,  $k_{s_y(3)}^*$  the key candidate giving rise to the  $y^*$  candidate that is the second closest to  $y$ ,  $\dots$ . For example, in the case of Hamming weight leakages as in our previous experiments, the close  $y^*$ ’s are the ones with Hamming weight  $\text{HW}(y^*)$  close to  $\text{HW}(y)$ .<sup>6</sup> Based on these  $s_y$  vectors, we can compute the conditional probabilities of the aggregated key variable  $K_{\text{agg}}^c$  as follows:

$$\Pr[K_{\text{agg}}^c = k | x, \bar{\mathbf{l}}_y] = \frac{\sum_{i=1}^c \Pr[K = k_{s_y(i)}^* | x, \bar{\mathbf{l}}_y]}{\sum_{k^* \in \mathcal{K}} \Pr[K = k^* | x, \bar{\mathbf{l}}_y]}. \quad (23)$$

This means that for a leakage vector  $\bar{\mathbf{l}}_y$ , the probability of the aggregated key variable corresponds to the sum of the probabilities for the key candidates corresponding to the  $c$  most likely  $y^*$  candidates given by  $\text{MIM}_{y,y^*}$ . Based on these probabilities, we finally compute the Normalized Aggregated MI (NAMI) as:

$$\begin{aligned} \text{NAMI}(K_{\text{agg}}^c; X, \bar{\mathbf{L}}_Y) &= \frac{H[K]}{H[K_{\text{agg}}^c]} \left( H[K_{\text{agg}}^c] + \sum_{k \in \mathcal{K}} \Pr[k] \cdot \sum_{x \in \mathcal{X}, \bar{y} \in \mathcal{Y}^d} \Pr[x, \bar{y}] \cdot \right. \\ &\quad \left. \sum_{\bar{\mathbf{l}}_y \in \mathcal{L}^d} \Pr[\bar{\mathbf{l}}_y | k, x, \bar{y}] \cdot \log_2 \Pr[K_{\text{agg}}^c = k | x, \bar{\mathbf{l}}_y] \right), \quad (24) \end{aligned}$$

where  $H[K_{\text{agg}}^c] = -\log(c/2^n)$  for uniformly distributed keys, that we will denote  $\text{NAMI}(c)$  for short. It captures the (normalized) amount of information the adversary obtains about a set of  $c$  key candidates that he then has to enumerate. Two

<sup>5</sup> For low noise levels for which this value is not defined (because of  $y^*$  candidates with probability 0), simpler (e.g. Euclidean) distances can serve as heuristic alternatives.

<sup>6</sup> It is worth mentioning that the distribution of the values of  $\text{MIM}_{k,-}$  is independent of the value of  $k$ , because of the key equivalence property discussed in [48]. Intuitively this is because for each  $k$ ,  $\text{MIM}_{k,-}$  is computed by summing over all the  $2^n$  possible  $x$ ’s and  $y$ ’s. By contrast, the distribution of the values of  $\text{MIM}_{y,-}$  depend on the value of  $y$ , since this time the MIM is computed for a fixed value of  $x$  and  $y$  which may lead to more or less informative leakages (see Appendix, Figure 15).

important properties of the NAMI are that it preserves the full informativeness of the leakages (i.e. if  $\text{MI}(K; X, \bar{L}_Y) = \text{H}[K]$ , then  $\text{NAMI}(c) = \text{H}[K]$  for all  $c$ 's) and their non informativeness (i.e. if  $\text{MI}(K; X, \bar{L}_Y) = 0$ , then  $\text{NAMI}(c) = 0$  for all  $c$ 's). These properties are best illustrated with examples. First, say we have a (non-informative) conditional probability for a 2-bit key  $\Pr[k|x, \bar{l}_y] = \frac{1}{4}$ , such that  $\text{MI}(k; x, \bar{l}_y) = 0$ . Then, aggregation with  $c = 2$  leads to  $\Pr[k_{\text{agg}}^c|x, \bar{l}_y] = \frac{1}{2}$ , corresponding to  $\text{NAMI}(c) = 0$  (since both the entropy and the conditional entropy of the aggregated key are reduced to 1). Second, say we have a (fully informative) conditional probability for a 2-bit key  $\Pr[k|x, \bar{l}_y] = 1$ , such that  $\text{MI}(k; x, \bar{l}_y) = 2$  (i.e.  $\text{NAMI}(1) = 2$  thanks to normalization). Then, the aggregation with  $c = 2$  leads to  $\Pr[k_{\text{agg}}^c|x, \bar{l}_y] = 1$ , which again corresponds to  $\text{NAMI}(c) = 2$ .

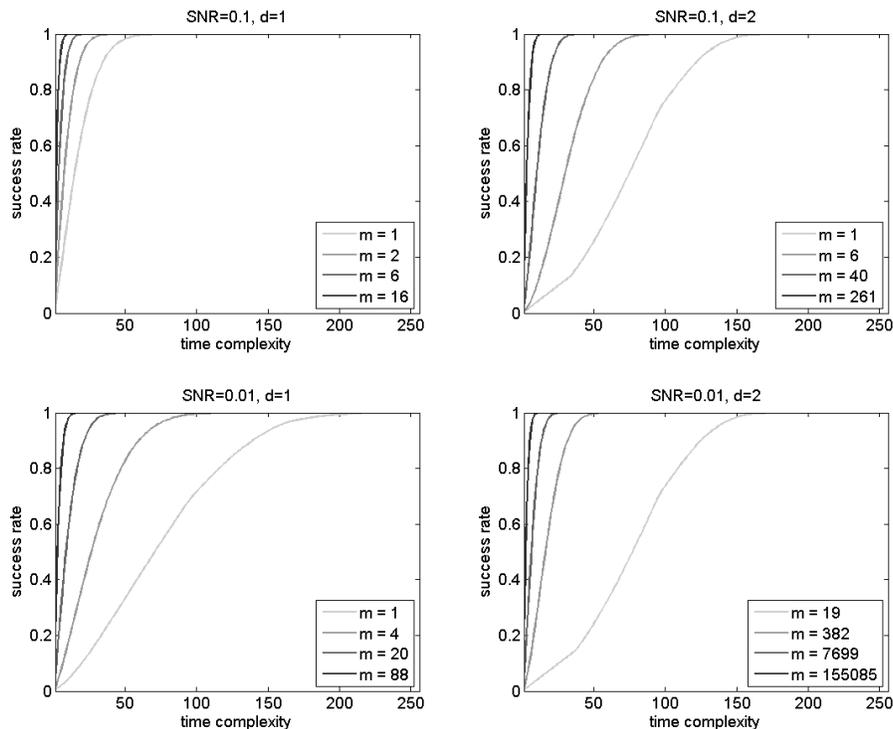
Note that as in Section 4.1, Paragraph (c), we could normalize the aggregated MI with  $\frac{1}{\text{H}[K_{\text{agg}}^c]}$  in order to have better approximations even in the case of low-noise leakages (although only the normalization with  $\frac{\text{H}[K]}{\text{H}[K_{\text{agg}}^c]}$  strictly follows the bounds). Note also that the NAMI is not always increasing with  $c$ .

The next step is to translate the NAMI into success rate curves for a single S-box. Here, we obtain (pessimistic) bounds by simply assuming that the adversary can test (i.e. brute force) the  $c$  candidates, each of them having a probability of success defined by  $\text{NAMI}(c)$ . That is, assuming a single measurement, the success rate  $\text{SR}^{\text{dc}}(m = 1, c = 1) \leq \text{MI}^d$  of Section 4.1 becomes  $\text{SR}^{\text{dc}}(m = 1, c) \leq 1 - (1 - \text{NAMI}(c)^d)^c$ , which equals  $\text{MI}^d$  in case  $c = 1$ . Here, the dc superscript recalls the specialization to divide-and-conquer attacks. We then generalize this success rate to multiple measurements as in Equation (19), leading to:

$$\begin{aligned} \text{SR}^{\text{dc}}(m, c) &\leq 1 - \left(1 - \text{SR}^{\text{dc}}(m = 1, c)\right)^m + \frac{c}{2^n}, \\ &\leq 1 - \left(1 - \text{NAMI}(c)^d\right)^{m \cdot c} + \frac{c}{2^n}, \end{aligned} \quad (25)$$

where the additional term  $\frac{c}{2^n}$  corresponds to the exhaustive search in case leakages are not informative. A couple of such bounds are given in Figure 8 for illustration, where we can see the impact of increasing the number of shares  $d$ , number of measurements  $m$  and noise level (here reported with the SNR). For example, the linearly shaped curves (as in the lower right plot, for  $m = 19$ ) typically indicate that the leakages are not informative enough and that the additive exhaustive search term  $\frac{c}{2^n}$  dominates in the success rate equation.

Note that despite requiring similar characterization efforts, these bounds are conceptually different from the previous approaches to approximate the success rate of side-channel attacks. In particular, works like [25, 32, 45, 65] are specific to popular distinguishers (and usually require specialized assumptions about the distribution of these distinguishers), while our results directly connect to security proofs that are independent of the adversarial strategy and hold for any leakage distribution. In other words, these related works bring a different tradeoff



**Fig. 8.** Single S-box key recovery success rate derived from the approximated NAMI, in function of the time complexity  $c$  for various SNRs and number of measurements  $m$ .

by providing more accurate and specific estimations.<sup>7</sup> They are anyway complementary, since the only requirement to analyze the combination of multiple S-boxes as proposed in the next paragraph (c) is to have success rates curves for each S-box. So while the previous proposal and Equation (25) provide an efficient way to build such curves, the following contribution is general, and could be used together with any security evaluation obtained for separate S-boxes.

**c. Combining multiple S-boxes.** We finally generalize our analysis of the previous paragraph to the case where we target  $N_s$  S-boxes (e.g.  $N_s = 16$  for the AES), gained information about their respective input key bytes, and want to recover the full master key. We assume that we perform the same amount of measurements  $m$  on each S-box. This can be easily justified in practice, since a leakage trace usually contains samples corresponding to all S-boxes. By contrast, we make no assumption about how informative the leakages of each S-box

<sup>7</sup> How specific is the estimation actually depends on the distinguishers. For example, if the estimated success rate corresponds to a template attack, such as discussed in [45], Section 6, it is then very similar to the previous information theoretic estimations and may be use to obtain tighter estimations than given by our bounds.

are. For example, it could happen that one S-box is very leaky, and another one perfectly protected (so that enumeration is the only option to recover its corresponding key byte). For this purpose, we first characterize the measurement vs. complexity tradeoff with  $N_s$  success rate matrices  $\text{SR}_i^{\text{dc}}(m, c_i)$  such that  $1 \leq i \leq N_s$  and  $1 \leq c_i \leq 2^n$  (as just explained). We then aim to bound or approximate the total success rate  $\text{SR}^{\text{dc}}(m, c)$ , such that  $1 \leq c_i \leq 2^{N_s \cdot n}$ .

The problem of evaluating the remaining time complexity for brute-forcing a key after some partial knowledge has been obtained thanks to side-channel analysis has been introduced in the literature as the “rank estimation problem” [76]. Intuitively, it can be viewed as the evaluator’s counterpart to the (adversary’s) problem of “key enumeration” [75]. The main difference between rank estimation and key enumeration is that in the first case, the value of the key is known to the evaluator (as in the discussions of this paper) and the algorithm is only looking for its position in the list of all key candidates. By contrast, in the second case the key is unknown and the goal of the algorithm is to list the most likely candidates up to some bounded rank (corresponding to the adversary’s computing power). Concretely, rank estimation usually takes vectors of key bytes probabilities as input, from which it estimates a rank for the master key. Several efficient solutions have been introduced for this purpose, e.g. [13, 35, 50]. Yet, in order to produce a security graph, it is then needed to repeat attacks and rank estimation multiple times in order to estimate a success rate – a task that can become cumbersome as the security of an implementation increases. Motivated by this drawback Ye et al. proposed an alternative solution, where the success rate is first estimated for every key byte independently, and then combined – but could only derive success rate lower bounds [82]. More recently, Poussier et al. proposed a collection of tools allowing to derive lower and upper bounds on the adversary’s global success rate based on the key bytes’ success rates [58]. In the following, we will combine these tools with our bounds or approximations of the S-box success rates, in order to produce different types of security graphs. More precisely, we will consider the following four combinations:

1. *MI bound, SR bound*, where we use the single S-box success rate bound (with the aggregated MI normalized with  $\frac{H[K]}{H[K_{\text{agg}}^c]}$ ) and the multiple S-boxes success rate upper bound in [58], which corresponds to a worst-case scenario.
2. *MI bound, SR heuristic*, where we use the single S-box success rate bound and the multiple S-boxes success rate lower bound in [58], which leads to a less pessimistic view from the time complexity point-of-view.
3. *MI approximation, SR bound*, where we use the single S-box success rate approximation (with the aggregated MI normalized with  $\frac{1}{H[K_{\text{agg}}^c]}$ ) and the multiple S-boxes success rate upper bound in [58], which leads to a less pessimistic view from the measurement complexity point-of-view.
4. *MI approximation, SR heuristic*, where we use the single S-box success rate approximation and the multiple S-boxes success rate lower bound in [58], which leads to a less pessimistic (and hopefully realistic) view from both the time complexity and the measurement complexity points-of-view.

---

**Algorithm 1** Metric-based success rate lower bound.

---

**Input:** Lists of success rates  $SR_i$ .

**Output:** A lower bound on the master key success rate.

```

 $SR_{low} \leftarrow \text{downs\_log}(SR_0);$ 
for  $i = 1$  to  $N_s - 1$  do
   $SR_{low}^i \leftarrow \text{downs\_log}(SR_i);$ 
   $SR_{low} \leftarrow \text{comb\_log}(SR_{low}, SR_{low}^i);$ 
end for
return  $SR_{low}$ 

```

---

Concretely, the lower and upper bounds for the multiple S-boxes success rates are obtained with simple manipulations of the single S-box success rate curves, namely logarithmic downsampling, logarithmic combination, logarithmic indexing and convolution (see the details in [58]). For completeness, we provide a high-level description of their computation in Algorithms 1 and 2.

---

**Algorithm 2** Metric-based success rate upper bound.

---

**Input:** Lists of derivative success rates  $\Delta SR_i$ .

**Output:** An upper bound on the master key success rate.

```

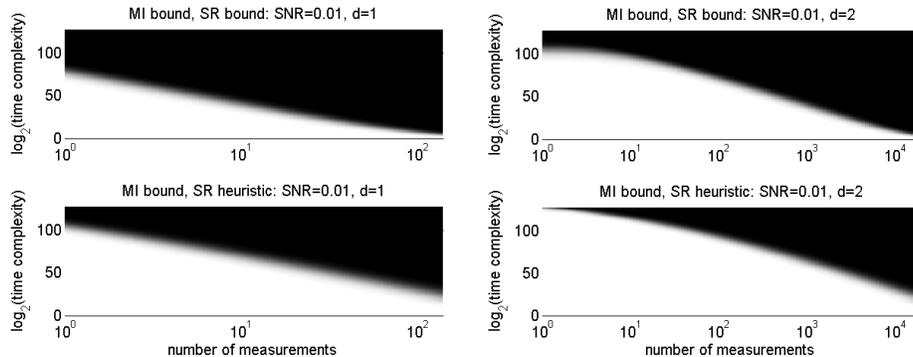
 $LI_{curr} = \text{index\_log}(\Delta SR_0);$ 
for  $i = 1$  to  $N_s - 1$  do
   $LI_i \leftarrow \text{index\_log}(\Delta SR_i);$ 
   $LI_{curr} \leftarrow \text{conv}(LI_{curr}, LI_i);$ 
end for
 $SR_{up}[0] \leftarrow \sum_{j=0}^{j=\frac{1}{w}-1} LI_{curr}[j];$ 
for  $i = 1$  to  $N_s \cdot n - 1$  do
   $SR_{up}[i] \leftarrow SR_{up}[i - 1] + \sum_{j=\frac{1}{w} \cdot i}^{j=\frac{1}{w} \cdot (i+1) - 1} LI_{curr}[j];$ 
end for
 $SR_{up}[n] \leftarrow SR_{up}[n - 1] + LI_{curr}[n];$ 
return  $SR_{up}$ 

```

---

Examples of security graphs are given in Figure 9 for  $d = 1$  and  $d = 2$  shares, and in Figure 10 for  $d = 3$  and  $d = 4$  shares, where the upper (resp. lower) parts of the figures correspond to our first (resp. second) combination. They lead to a number of interesting observations. First, they confirm the exponential security increase that masking provides thanks to noise amplification. Second, they show that even taking conservative bounds, it is possible to obtain acceptable security levels under reasonable parameters (although larger than usually considered in the state-of-the-art literature, e.g.  $\text{SNR} = 0.01$  and  $d > 4$ ). Third, they illustrate that the upper bound of Algorithm 2 is quite pessimistic (as witnessed by the “plateau” regions in Figure 10, where leakages are not informative enough and success is obtained thanks to exhaustive search – which should only succeed for complexities close to  $N_s \cdot n$  as in the lower part of the figure). Finally, and most importantly, these figures were all obtained in seconds of computations using a prototype code that we also release in open source for further evaluations [1]. So

the tools proposed in this final section are perfectly suited to efficiently capture the main security parameters of a leaking implementation. Additional security graphs are given for the same SNR and number of shares but using combinations 3 and 4 in Appendix, Figures 16 and 17, which naturally leads to less conservative estimations of the security level. Finally, and using the same combinations (3 and 3), we also evaluated the security for a lower SNR = 0.1 in Figures 18 and 19.

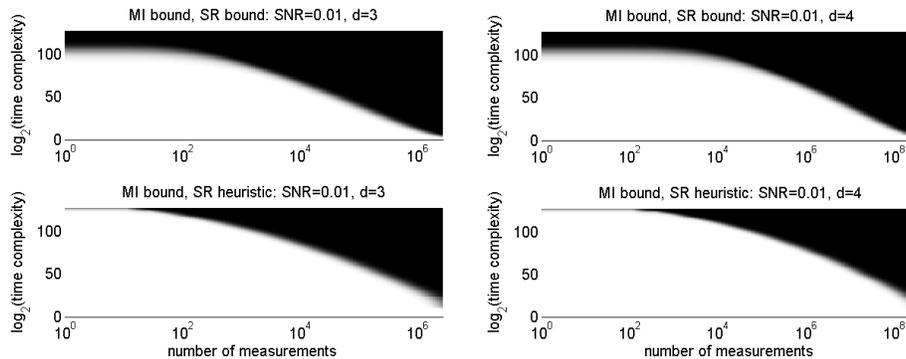


**Fig. 9.** Example of security graph for security orders  $d = 1, 2$ .

## 5 Conclusion

Our results show that the (complex) task of evaluating the worst-case security level of a masked implementation against side-channel attacks can be simplified into the evaluation of a couple of MI values, even in contexts where the independence assumption is not fulfilled. This provides a solid foundation for the Eurocrypt 2009 evaluation framework. It also makes it easier to perform comprehensive evaluations of divide-and-conquer DPA since in this case, success rate curves for full keys can now be derived from the MI values as well, rather than sampled experimentally by repeating (many) subkey recovery experiments and key rank estimations, which is an expensive task. Taking advantage of the tools in this paper therefore allow reducing both the number of measurements and the time needed to evaluate leaking devices. Applying these tools to concrete implementations protected with various types of countermeasures, in particular for contexts where the independence assumption is not perfectly respected (as discussed in Section 4.2), is an interesting scope for further investigation.

**Acknowledgements.** Alexandre Duc is supported by the Swiss National Science Foundation, grant 200021 143899/1. Sebastian Faust received funding from the Marie Curie IEF/FP7 project GAPS (grant 626467). François-Xavier Standaert is a research associate of the Belgian Fund for Scientific Research. This work has been funded in parts by the ERC project 724725 (SWORD).



**Fig. 10.** Example of security graph for security orders  $d = 3, 4$ .

## References

1. <http://perso.uclouvain.be/fstandae/PUBLIS/154.zip>.
2. Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template Attacks in Principal Subspaces. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2006.
3. Thomas Baignères, Pascal Junod, and Serge Vaudenay. How Far Can We Go Beyond Linear Cryptanalysis? In Pil Joong Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450. Springer, 2004.
4. Josep Balasch, Sebastian Faust, Benedikt Gierlich, and Ingrid Verbauwhede. Theory and Practice of a Leakage Resilient Masking Scheme. In Wang and Sako [79], pages 758–775.
5. Josep Balasch, Benedikt Gierlich, Vincent Grosso, Oscar Reparaz, and François-Xavier Standaert. On the cost of lazy engineering for masked software implementations. In Joye and Moradi [43], pages 64–81.
6. Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong non-interference and type-directed higher-order masking. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 116–129. ACM, 2016.
7. Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017*, volume 10210 of *Lecture Notes in Computer Science*, pages 535–566, 2017.
8. Lejla Batina and Matthew Robshaw, editors. *CHES 2014*, volume 8731 of *Lecture Notes in Computer Science*. Springer, 2014.
9. Alberto Battistello, Jean-Sébastien Coron, Emmanuel Prouff, and Rina Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. In Benedikt Gierlich and Axel Y. Poschmann, editors, *CHES 2016*, volume 9813 of *Lecture Notes in Computer Science*, pages 23–39. Springer, 2016.

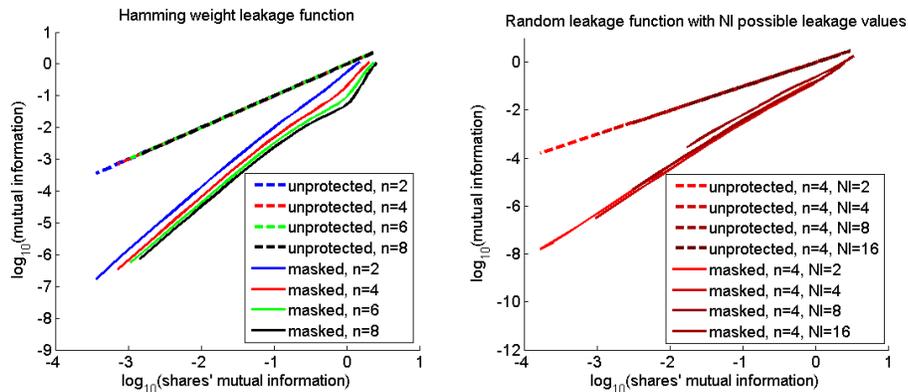
10. Sonia Belaïd, Vincent Grosso, and François-Xavier Standaert. Masking and Leakage-Resilient Primitives: One, the Other(s) or Both? *Cryptography and Communications*, 7(1):163–184, 2015.
11. Mihir Bellare, Stefano Tessaro, and Alexander Vardy. A Cryptographic Treatment of the Wiretap Channel. *IACR Cryptology ePrint Archive*, 2012:15, 2012.
12. Mihir Bellare, Stefano Tessaro, and Alexander Vardy. Semantic Security for the Wiretap Channel. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 294–311. Springer, 2012.
13. Daniel J. Bernstein, Tanja Lange, and Christine van Vredendaal. Tighter, faster, simpler side-channel security evaluations beyond computing power. *IACR Cryptology ePrint Archive*, 2015:221, 2015.
14. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.
15. Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Kernel discriminant analysis for information extraction in the presence of masking. In Kerstin Lemke-Rust and Michael Tunstall, editors, *CARDIS 2016*, volume 10146 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2016.
16. Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Housseem Maghrebi. Leakage Squeezing: Optimal Implementation and Security Evaluation. *J. Mathematical Cryptology*, 8(3):249–295, 2014.
17. Claude Carlet, Louis Goubin, Emmanuel Prouff, Michaël Quisquater, and Matthieu Rivain. Higher-Order Masking Schemes for S-Boxes. In Anne Canteaut, editor, *FSE 2012*, volume 7549 of *Lecture Notes in Computer Science*, pages 366–384. Springer, 2012.
18. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis. In Wiener [81], pages 398–412.
19. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.
20. Thomas De Cnudde, Begül Bilgin, Benedikt Gierlichs, Ventzislav Nikov, Svetla Nikova, and Vincent Rijmen. Does coupling affect the security of masked implementations? In Sylvain Guilley, editor, *Constructive Side-Channel Analysis and Secure Design - 8th International Workshop, COSADE 2017, Paris, France, April 13-14, 2017, Revised Selected Papers*, volume 10348 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2017.
21. Jean-Sébastien Coron, Christophe Giraud, Emmanuel Prouff, Soline Renner, Matthieu Rivain, and Praveen Kumar Vadnala. Conversion of Security Proofs from One Leakage Model to Another: A New Issue. In Werner Schindler and Sorin A. Huss, editors, *COSADE*, volume 7275 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2012.
22. Jean-Sébastien Coron, Emmanuel Prouff, and Matthieu Rivain. Side Channel Cryptanalysis of a Higher Order Masking Scheme. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 28–44. Springer, 2007.
23. Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-Order Side Channel Security and Mask Refreshing. In Shiho Moriai, editor, *FSE 2013*, volume 8424 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 2013.

24. Thomas Cover and Joy Thomas. *Elements of Information Theory*. Wiley, 2006.
25. A. Adam Ding, Liwei Zhang, Yungsi Fei, and Pei Luo. A Statistical Model for Higher Order DPA on Masked Devices. In Batina and Robshaw [8], pages 147–169.
26. Yevgeniy Dodis. Shannon Impossibility, Revisited. In Adam Smith, editor, *ICITS 2012*, volume 7412 of *Lecture Notes in Computer Science*, pages 100–110. Springer, 2012.
27. Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying Leakage Models: From Probing Attacks to Noisy Leakage. In Nguyen and Oswald [55], pages 423–440.
28. François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to Certify the Leakage of a Chip? In Nguyen and Oswald [55], pages 459–476.
29. François Durvaux, François-Xavier Standaert, and Santos Merino Del Pozo. Towards easy leakage certification: extended version. *J. Cryptographic Engineering*, 7(2):129–147, 2017.
30. Stefan Dziembowski, Sebastian Faust, and Maciej Skorski. Noisy leakage revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015*, volume 9057 of *Lecture Notes in Computer Science*, pages 159–188. Springer, 2015.
31. Sebastian Faust, Tal Rabin, Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting Circuits from Leakage: the Computationally-Bounded and Noisy Cases. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 135–156. Springer, 2010.
32. Yungsi Fei, Qiasi Luo, and A. Adam Ding. A Statistical Model for DPA with Novel Algorithmic Confusion Analysis. In Emmanuel Prouff and Patrick Schaumont, editors, *CHES 2012*, volume 7428 of *Lecture Notes in Computer Science*, pages 233–250. Springer, 2012.
33. Guillaume Fumaroli, Ange Martinelli, Emmanuel Prouff, and Matthieu Rivain. Affine Masking against Higher-Order Side Channel Analysis. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *SAC 2010*, volume 6544 of *Lecture Notes in Computer Science*, pages 262–280. Springer, 2010.
34. Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual Information Analysis. In Oswald and Rohatgi [57], pages 426–442.
35. Cezary Glowacz, Vincent Grosso, Romain Poussier, Joachim Schüth, and François-Xavier Standaert. Simpler and more efficient rank estimation for side-channel security assessment. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *Lecture Notes in Computer Science*, pages 117–129. Springer, 2015.
36. Louis Goubin and Ange Martinelli. Protecting AES with Shamir’s Secret Sharing Scheme. In Preneel and Takagi [59], pages 79–94.
37. Vincent Grosso, Emmanuel Prouff, and François-Xavier Standaert. Efficient Masked S-Boxes Processing - A Step Forward -. In David Pointcheval and Damien Vergnaud, editors, *AFRICACRYPT 2014*, volume 8469 of *Lecture Notes in Computer Science*, pages 251–266. Springer, 2014.
38. Vincent Grosso and François-Xavier Standaert. Masking proofs are tight (and how to exploit it in security evaluations). *IACR Cryptology ePrint Archive*, 2017:116, 2017.
39. Vincent Grosso, François-Xavier Standaert, and Emmanuel Prouff. Low Entropy Masking Schemes, Revisited. In Aurélien Francillon and Pankaj Rohatgi, editors, *CARDIS 2013*, volume 8419 of *Lecture Notes in Computer Science*, pages 33–43. Springer, 2013.
40. Sylvain Guilley, Annelie Heuser, Olivier Rioul, and François-Xavier Standaert. Template attacks, optimal distinguishers & the perceived information metric. CryptArchi 2015, Leuven, Belgium, July 2015.

41. Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.
42. Thomas Johansson and Phong Q. Nguyen, editors. *EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*. Springer, 2013.
43. Marc Joye and Amir Moradi, editors. *CARDIS 2014*, volume 8968 of *Lecture Notes in Computer Science*. Springer, 2015.
44. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In Wiener [81], pages 388–397.
45. Victor Lomné, Emmanuel Prouff, Matthieu Rivain, Thomas Roche, and Adrian Thillard. How to Estimate the Success Rate of Higher-Order Side-Channel Attacks. In Batina and Robshaw [8], pages 35–54.
46. Stefan Mangard. Hardware Countermeasures against DPA ? A Statistical Analysis of Their Effectiveness. In Tatsuaki Okamoto, editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.
47. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks - Revealing the Secrets of Smart Cards*. Springer, 2007.
48. Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for All - All for One: Unifying Standard Differential Power Analysis Attacks. *IET Information Security*, 5(2):100–110, 2011.
49. Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.
50. Daniel P. Martin, Jonathan F. O’Connell, Elisabeth Oswald, and Martijn Stam. Counting keys in parallel after a side channel attack. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015*, volume 9453 of *Lecture Notes in Computer Science*, pages 313–337. Springer, 2015.
51. Luke Mather, Elisabeth Oswald, and Carolyn Whitnall. Multi-target DPA attacks: Pushing DPA beyond the limits of a desktop computer. In Sarkar and Iwata [68], pages 243–261.
52. Marcel Medwed and François-Xavier Standaert. Extractors against Side-Channel Attacks: Weak or Strong? *J. Cryptographic Engineering*, 1(3):231–241, 2011.
53. Amir Moradi and Oliver Mischke. Glitch-Free Implementation of Masking in Modern FPGAs. In *HOST*, pages 89–95. IEEE, 2012.
54. Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TISCCS 2016 Vienna, Austria, October, 2016*, pages 5–15. ACM, 2016.
55. Phong Q. Nguyen and Elisabeth Oswald, editors. *EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*. Springer, 2014.
56. Svetla Nikova, Vincent Rijmen, and Martin Schl affer. Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. *J. Cryptology*, 24(2):292–321, 2011.
57. Elisabeth Oswald and Pankaj Rohatgi, editors. *CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*. Springer, 2008.
58. Romain Poussier, Vincent Grosso, and François-Xavier Standaert. Comparing approaches to rank estimation for side-channel security evaluations. In Naofumi Homma and Marcel Medwed, editors, *CARDIS 2015*, volume 9514 of *Lecture Notes in Computer Science*, pages 125–142. Springer, 2015.
59. Bart Preneel and Tsuyoshi Takagi, editors. *CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*. Springer, 2011.

60. Emmanuel Prouff and Matthieu Rivain. Masking against Side-Channel Attacks: A Formal Security Proof. In Johansson and Nguyen [42], pages 142–159.
61. Emmanuel Prouff and Thomas Roche. Attack on a Higher-Order Masking of the AES Based on Homographic Functions. In Guang Gong and Kishan Chand Gupta, editors, *INDOCRYPT 2010*, volume 6498 of *Lecture Notes in Computer Science*, pages 262–281. Springer, 2010.
62. Mathieu Renauld, Dina Kamel, François-Xavier Standaert, and Denis Flandre. Information Theoretic and Security Analysis of a 65-Nanometer DDSLL AES S-Box. In Preneel and Takagi [59], pages 223–239.
63. Mathieu Renauld, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Algebraic Side-Channel Attacks on the AES: Why Time also Matters in DPA. In Christophe Clavier and Kris Gaj, editors, *CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 97–111. Springer, 2009.
64. Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 109–128. Springer, 2011.
65. Matthieu Rivain. On the Exact Success Rate of Side Channel Analysis in the Gaussian Model. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC 2008*, volume 5381 of *Lecture Notes in Computer Science*, pages 165–183. Springer, 2008.
66. Matthieu Rivain and Emmanuel Prouff. Provably Secure Higher-Order Masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2010.
67. Thomas Roche and Emmanuel Prouff. Higher-order Glitch Free Implementation of the AES using Secure Multi-Party Computation Protocols - Extended Version. *J. Cryptographic Engineering*, 2(2):111–127, 2012.
68. Palash Sarkar and Tetsu Iwata, editors. *ASIACRYPT 2014*, volume 8873 of *Lecture Notes in Computer Science*. Springer, 2014.
69. Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.
70. François-Xavier Standaert and Cédric Archambeau. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In Oswald and Rohatgi [57], pages 411–425.
71. François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.
72. François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The World is Not Enough: Another Look on Second-Order DPA. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.
73. François-Xavier Standaert, Eric Peeters, Gaël Rouvroy, and Jean-Jacques Quisquater. An Overview of Power Analysis Attacks against Field Programmable Gate Arrays. *Proceedings of the IEEE*, 94(2):383–394, 2006.

74. François-Xavier Standaert, Christophe Petit, and Nicolas Veyrat-Charvillon. Masking with Randomized Look Up Tables - Towards Preventing Side-Channel Attacks of All Orders. In David Naccache, editor, *Cryptography and Security: From Theory to Applications*, volume 6805 of *Lecture Notes in Computer Science*, pages 283–299. Springer, 2012.
75. Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renaud, and François-Xavier Standaert. An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2012.
76. Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Security Evaluations beyond Computing Power. In Johansson and Nguyen [42], pages 126–141.
77. Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Soft analytical side-channel attacks. In Sarkar and Iwata [68], pages 282–296.
78. Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against Side-Channel Attacks: A Comprehensive Study with Cautionary Note. In Wang and Sako [79], pages 740–757.
79. Xiaoyun Wang and Kazue Sako, editors. *ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*. Springer, 2012.
80. Carolyn Whitnall and Elisabeth Oswald. A Comprehensive Evaluation of Mutual Information Analysis Using a Fair Evaluation Framework. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 316–334. Springer, 2011.
81. Michael J. Wiener, editor. *CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.
82. Xin Ye, Thomas Eisenbarth, and William Martin. Bounded, yet sufficient? how to determine whether limited side channel information enables key recovery. In Joye and Moradi [43], pages 215–232.



**Fig. 11.** MI metric in fct. of the shares' MI. HW (left) and random (right) leakages.

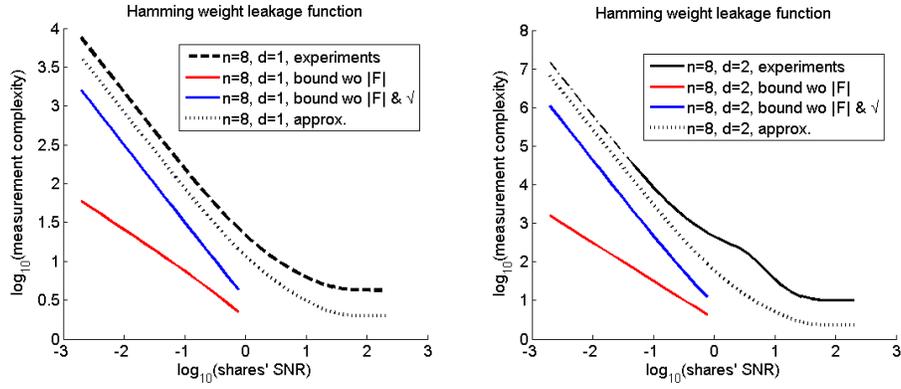


Fig. 12. Measurement complexity and bounds/approximations for concrete TA.

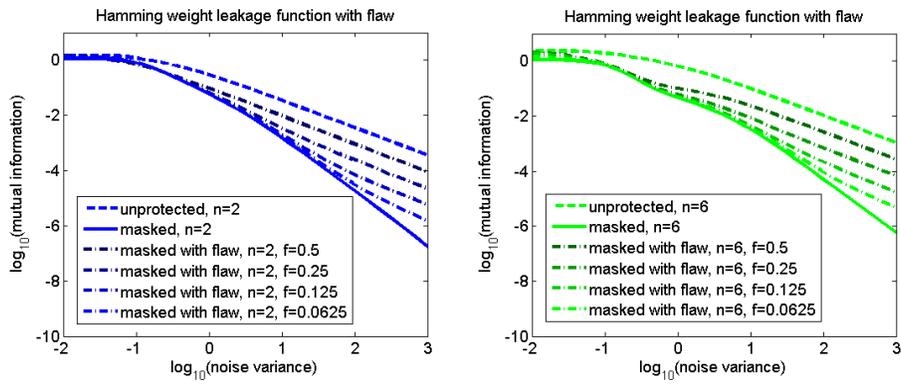


Fig. 13. MI metric for masked implementation with flaw ( $n = 2, 6$ ).

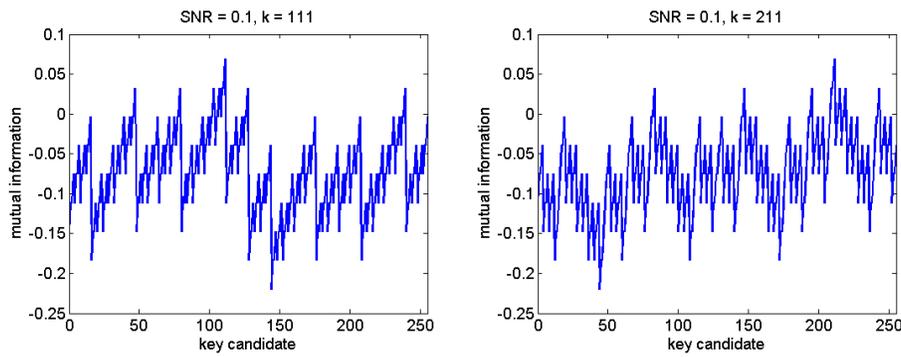


Fig. 14. Exemplary lines of the mutual information matrix  $\text{MIM}_{k,-}$  (medium noise).

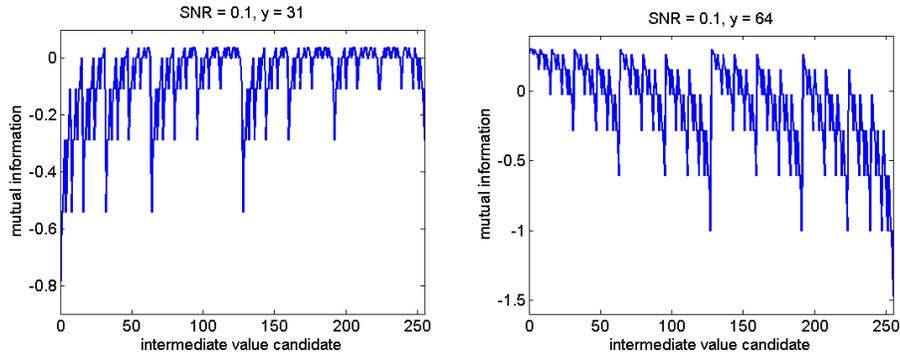


Fig. 15. Exemplary lines of the mutual information matrix  $MIM_{y,-}$  (medium noise).

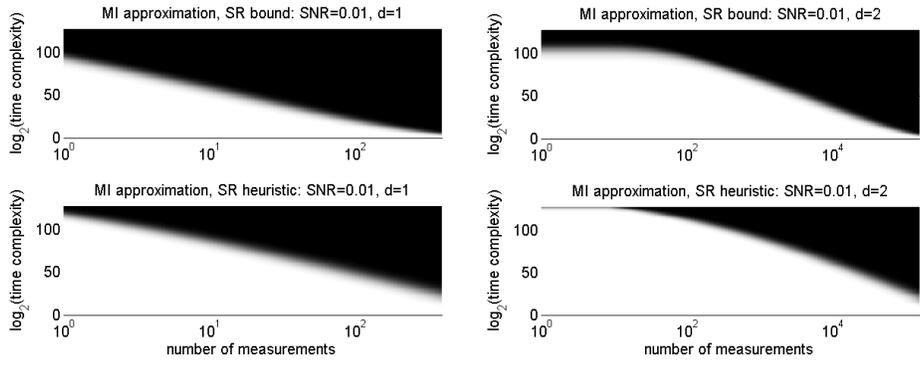


Fig. 16. Example of security graph for security orders  $d = 1, 2$ .

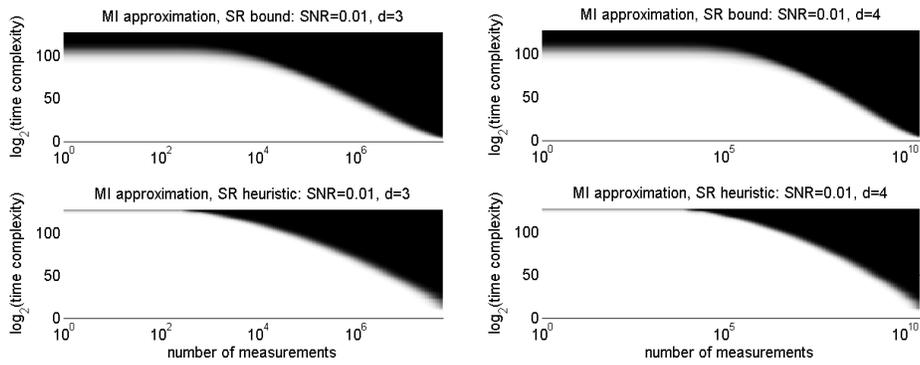
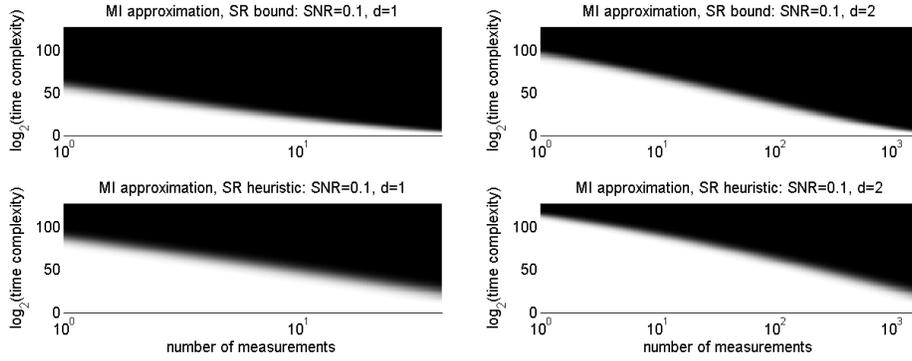
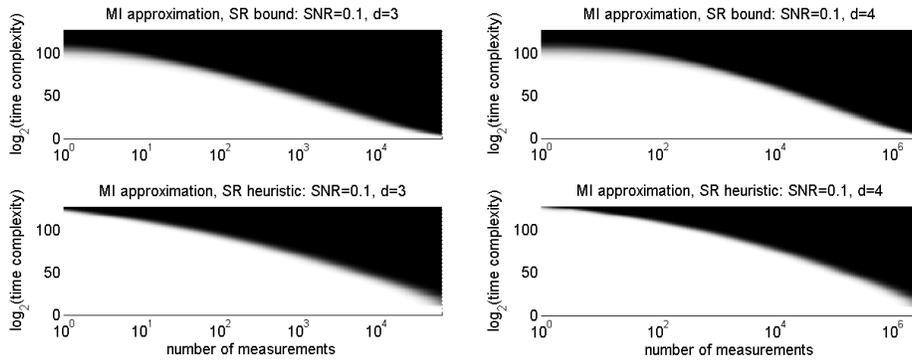


Fig. 17. Example of security graph for security orders  $d = 3, 4$ .



**Fig. 18.** Example of security graph for security orders  $d = 1, 2$ .



**Fig. 19.** Example of security graph for security orders  $d = 3, 4$ .