A Formal Practice-Oriented Model for the Analysis of Side-Channel Attacks

François-Xavier Standaert[1⋆], Tal G. Malkin[2], Moti Yung[2,3]

[1] UCL Crypto Group, Université Catholique de Louvain.
[2] Dept. of Computer Science, Columbia University.
[3] RSA Laboratories.

e-mails: fstandae@uclouvain.be, tal,moti@cs.columbia.edu

Version 1.6, June 8, 2007.

Abstract. Formal models that allow one to understand side-channel attacks and are also directly meaningful to practice have been an open question. Motivated by this challenge, this work proposes a practice oriented framework for the analysis of cryptographic implementations against such attacks. It is illustratively applied to block ciphers, although it could be used to analyze other cryptosystems. The model is based on weak and commonly accepted hypotheses about side-channels that computations give rise to. It allows us to quantify the effect of practically relevant leakage functions with a combination of security and information theoretic metrics. From a practical point of view, the model suggests a unified evaluation methodology for side-channel attacks. From a theoretical point of view, it allows discussing the fundamental tradeoffs in such attacks, namely flexibility *vs.* efficiency and information *vs.* computation.

1 Introduction

Traditionally, cryptographic algorithms provide security against an adversary who has only black box access to cryptographic devices. That is, the only thing the adversary can do is to query the cryptographic algorithm on inputs of its choice and analyze the responses, which are always computed according to the correct original secret information. However, such a model does not always correspond to the realities of physical implementations. During the last decade, significant attention has been paid to the physical security evaluation of cryptographic devices. In particular, it has been demonstrated that actual attackers may be much more powerful than what can be captured by the black box model. In this paper, we investigate the security of cryptographic implementations with respect to side-channel attacks, in which adversaries are enhanced with the possibility to exploit physical leakages such as power consumption or electromagnetic radiation. A large body of experimental work has been created on the subject, and although numerous countermeasures are proposed in the literature, protecting implementations against such attacks is usually difficult and expensive. Moreover, most proposals we are aware of only increase the difficulty of performing the attacks, but do not fundamentally prevent them [13].

⋆ Postdoctoral researcher of the Belgian Fund for Scientific Research (FNRS).

Perhaps surprisingly (and to the best of our knowledge), there have been only a few attempts to model such physical attacks properly, and to provably address their security. A notable example is the work of Micali and Reyzin who initiated a theoretical analysis of side-channels, taking the modularity of physically observable computations into account. It notably defines the notion of *physical computer* that is basically the combination of an abstract computer (*i.e.* a Turing machine) and a leakage function. The model in [24] is very general, capturing almost any conceivable form of physical leakage. However, arguably because of the great generality of the assumptions, the obtained positive results (*i.e.* leading to useful constructions) are quite restricted in nature, and it is not clear how they apply to practice. This is especially true for primitives such as modern block ciphers for which even the black box security cannot be proven. Thus, the study of more specialized contexts and specific scenarios which may lead to practical applications was suggested as an open question. Motivated by this challenge, we propose to analyze side-channel attacks in a model of computation that captures the structure and operations of modern block ciphers. Still, the model is general and can be used to analyze other cryptosystems.

Our results are in three parts. First, we restrict the model of Micali and Reyzin to reasonable (*i.e.* practically relevant) adversaries. Namely, we focus on the side-channel key recovery that is considered in most present attacks. Consequently, we introduce a combination of metrics to measure both the amount of information provided by a given physical computer and the extent to which an adversary can turn this information into a practical attack. Second, we relate the introduced formal definitions to more intuitive physical notions. In particular, we translate physical computers into a combination of signals and operations and describe the different steps in a side-channel attack (*e.g.* characterization, profiling, measurement, statistical comparison, ...). Consequently to these practice-oriented definitions, we put forward a number of practical features of our evaluation metrics. Finally, we introduce a unified evaluation methodology for side-channel attacks, combining theoretical and practical aspects of our model.

2 Formal definitions

In this first section, we introduce the theoretical notions necessary for the analysis of side-channel attacks. We start with the definition of a leakage function introduced by Micali and Reyzin in [24]. The leakage function models the physical observations of a target device. Then, in order to evaluate the extent to which these leakages can be exploited by a given adversary, we formally define the notion of security against side-channel key-recovery in Section 2.2, with the necessary computational assumptions. This definition was selected for practical reasons (most present side-channel attacks purpose key recovery) and as a first step towards the theoretical modelling of physically observable cryptography. Finally, in the last part of this section, we discuss the evaluation criteria for side-channel attacks. We introduce a combination of information theoretic and security metrics in order to evaluate respectively the amount of information leaked by a device and the strength of a side-channel adversary.

2.1 Background: Micali-Reyzin computational model

In order to enable the analysis of physically observable cryptography, Micali and Reyzin introduced a model of computation of which we recall certain definitions of interest with respect to our following results. It is based on the five informal axioms given in Appendix B. From these axioms, an *abstract computer* was defined in [24] as a collection of special Turing machines, which invoke each other as subroutines and share a special common memory. Each member of the collection is denoted as an *abstract virtual-memory Turing machine* (abstract VTM or simply VTM for short). One writes $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ to mean that an abstract computer α consists of abstract VTMs $\alpha_1, \alpha_2, ..., \alpha_n$. All VTM inputs and outputs are binary strings always residing in some virtual memory. Abstract computers and VTMs are not physical devices: they only represent logical computation and may have many different physical realizations.

Then, to model the physical leakage of any particular instantiation of an abstract computer, the notion of *physical VTM* was introduced. A physical VTM is a pair (L_i, α_i), where α_i is an abstract VTM and L_i is a leakage function. If $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ is an abstract computer, then $\varphi_i = (\mathsf{L}_i, \alpha_i)$ represents one physical realization of α_i and $\varphi = (\varphi_1, \varphi_2, ..., \varphi_n)$ is defined as a physical realization of the abstract computer α, also called physical computer for short. It can be denoted as the combination $\varphi = (\alpha, \mathsf{L})$ with $\mathsf{L} = (\mathsf{L}_1, \mathsf{L}_2, \ldots, \mathsf{L}_n)$.

In these definitions, the relation between an abstract computing machine and a physical realization is only determined by the leakage function that is qualitatively defined as a function of three inputs, $\mathsf{L}(C_\alpha, M, R)$:

- The first input is the current internal configuration C_α of an abstract computer α, which incorporates anything that is in principle measurable.
- The second input M is the setting of the measuring apparatus (*i.e.* a specification of what and how the adversary chooses to measure).
- The third input R is a random string to model the randomness of the measurement process, *e.g.* the noise that affects the useful leakage signal.

2.2 Definition of security against side-channel key recovery

As a matter of fact, the previous definition of leakage function models the physical observations of a target device, but does not specify how an adversary could exploit this side-channel information. This section consequently intends to define a side-channel key recovery adversary. As most cryptanalytic techniques, side-channel attacks are usually based on a divide-and-conquer strategy in which different (computationally tractable) parts of a secret key are recovered separately. In general, the attack defines a function $\delta : \mathcal{K} \rightarrow \mathcal{S}$ which maps each key k onto an equivalent key class $s_g = \delta(k)$, such that $|\mathcal{S}| \ll |\mathcal{K}|$.

Let $\mathsf{E}_K(.) = \{\mathsf{E}_k\}_{k \in \mathcal{K}}$ be a family of cryptographic abstract computers indexed by a variable key K. Let $(\mathsf{E}_K, \mathsf{L})$ be the physical computer corresponding to the association of E_K with a leakage function L. We define a side-channel

key recovery adversary as an algorithm $\mathsf{A}_{\mathsf{E}_K,\mathsf{L}}$ with time complexity τ, memory complexity m and q queries to the target physical computer. The aim of a side-channel adversary is to guess a key class $s_g = \delta(k)$ with non negligible probability. It yields the experiment:

Experiment $\mathbf{Exp}^{\text{sc-kr}}_{\mathsf{E}_K,\mathsf{L}}$
$k \xleftarrow{R} \mathcal{K}$;
$s_g = \delta(k)$;
$s^* \leftarrow \mathsf{A}_{\mathsf{E}_k,\mathsf{L}}$;
if $s_g = s^*$ **then** return 1;
else return 0;

The success rate and advantage of the side-channel key recovery adversary $\mathsf{A}_{\mathsf{E}_K,\mathsf{L}}$ against a key class $\delta(K)$ are respectively defined as:

$$\mathbf{Succ}^{\text{sc-kr-}\delta(K)}_{\mathsf{A}_{\mathsf{E}_K,\mathsf{L}}} = \Pr\ [\mathbf{Exp}^{\text{sc-kr}}_{\mathsf{E}_K,\mathsf{L}} = 1] \tag{1}$$

$$\mathbf{Adv}^{\text{sc-kr-}\delta(K)}_{\mathsf{A}_{\mathsf{E}_K,\mathsf{L}}} = \Pr\ [\mathbf{Exp}^{\text{sc-kr}}_{\mathsf{E}_K,\mathsf{L}} = 1] - 1/|\mathcal{S}| \tag{2}$$

And for any τ, q, m, we define the δ-key recovery advantage of a cryptographic physical computer $(\mathsf{E}_K, \mathsf{L})$ against side-channel adversaries as:

$$\mathbf{Adv}^{\text{sc-kr-}\delta(K)}_{\mathsf{E}_K,\mathsf{L}} = \max_{\mathsf{A}}\{\mathbf{Adv}^{\text{sc-kr-}\delta(K)}_{\mathsf{A}_{\mathsf{E}_K,\mathsf{L}}}\} \tag{3}$$

Finally, if a side-channel attack is used to recover a part of an n-bit key and allows to recover the full key after having checked N_x candidates on average, the gain of the attack expressed in bits was defined in [6] as: $G = -\log_2 \frac{2 \cdot N_x - 1}{2^n}$. As a matter of fact, different attacks with small gains can be combined in order to increase the overall gain and reduce the computational cost of a key recovery.

Computational restrictions Similarly to black box security, computational restrictions have to be imposed on side-channel adversaries in order to capture the reality of physically observable cryptographic devices. Namely, the attack time complexity and memory complexity (mainly dependent on the number of key classes $N_s = |\mathcal{S}|$) are limited by present computer technologies. The number of queries q is limited by the adversary's abilities to monitor the device.

2.3 Evaluation criteria for side-channel attacks

From the previous definitions, it directly follows that there are two aspects to consider in the physical security evaluation of a cryptographic device, namely the quality of the physical computer and the strength of the adversary. They lead to the following questions that we would like to quantify in this section:

1. What is the amount of information provided by a given leakage function?
2. How successfully can this information be turned into a practical attack?

1. Information theoretic metric: the conditional entropy. Following the standard approach in information theory, we use Shannon's definition of entropy to quantify the amount of information leaked by a cryptographic device.

Let S_g (*resp.* S) be a discrete variable denoting the correct target key class (*resp.* a key class candidate) in a side-channel attack and s_g (*resp.* s) be a realization of this variable. Let $\mathbf{X}_q = [X_1, X_2, \ldots, X_q]$ be a vector of variables containing a sequence of inputs to the target physical computer and $\mathbf{x}_q = [x_1, x_2, \ldots, x_q]$ be a realization of this vector. Let $\mathbf{L}^q_{s_g}$ be a random vector denoting the side-channel observations generated by the correct key class s_g with q queries to the target physical computer and $\mathbf{l}^q_{s_g} = [l^1_{s_g}, l^2_{s_g}, \ldots, l^q_{s_g}]$ be a realization of this random vector, *i.e.* one actual output of the leakage function L. Let finally $\Pr[S = s|\mathbf{L}^q_{s_g} = \mathbf{l}^q_{s_g}]$ be the probability of a key class s given a leakage $\mathbf{l}^q_{s_g}$. We define the conditional entropy matrix as:

$$\mathbf{H}^q_{s_g,s} = \mathop{\mathbf{E}}_{\mathbf{l}^q_{s_g}} - \log_2 \Pr[S = s|\mathbf{L}^q_{s_g} = \mathbf{l}^q_{s_g}], \tag{4}$$

from which we derive Shannon's conditional entropy:

$$\mathrm{H}[S_g|\mathbf{L}^q_{s_g}] = \mathop{\mathbf{E}}_{s_g} \mathbf{H}^q_{s_g,s_g} \tag{5}$$

We note that this definition is equivalent to the classical one (see Appendix C). Then, we define an entropy reduction matrix: $\mathbf{H}^{*q}_{s_g,s} = \mathbf{H}[S_g] - \mathbf{H}^q_{s_g,s}$, where $\mathbf{H}[S_g]$ is a diagonal matrix of which all the non-zero entries are worth the entropy of the key class S_g before any side-channel attack has been performed, namely $\mathrm{H}[S_g] = \mathbf{E}_{s_g} - \log_2 \Pr[S_g = s_g]$. It directly yields the mutual information:

$$\mathrm{I}(S_g; \mathbf{L}^q_{s_g}) = \mathrm{H}[S_g] - \mathrm{H}[S_g|\mathbf{L}^q_{s_g}] = \mathop{\mathbf{E}}_{s_g} \mathbf{H}^{*q}_{s_g,s} \tag{6}$$

2. Security metric: the adversary's advantage or success rate. Following the standard approach in cryptography, we use the key recovery advantage (or success rate) of the side-channel adversary defined in Section 2.2 to quantify the security of a cryptographic device.

3. Relations between security and information. The intuition behind the proposed evaluation criteria for side-channel attacks is summarized in Figure 1. As already mentioned, security and information theoretic metrics quantify different aspects of a physical computer. Namely, the mutual information measures the average amount of information that is available in the observations while the success rate measures how efficiently an actual adversarial strategy can turn this information into a successful key recovery. By combining both measurements, one can analyze both the quality of a physical computer and the strength of a side-channel adversary. We now discuss how these notions are related through the asymptotic success rate, $\mathbf{Succ}^{\text{sc-kr-}S_g}_{\mathsf{A}_{\mathsf{E}_K},\mathsf{L}}(q \to \infty)$.

For this purpose, let us consider a Bayesian side-channel adversary that selects a key class $s^* = argmax_s \Pr[S = s|\mathbf{L}^q_{s_g} = \mathbf{l}^q_{s_g}]$. We say that a leakage function is *sound* if the asymptotic success rate of this Bayesian adversary equals one. It directly leads to the relation between information and security, namely: *under the condition that noise in the leakages is independent of key classes in the target devices, a leakage function is sound if and only if* $argmax_s \mathbf{H}^{*1}_{s_g,s} = s_g$, $\forall s_g$.

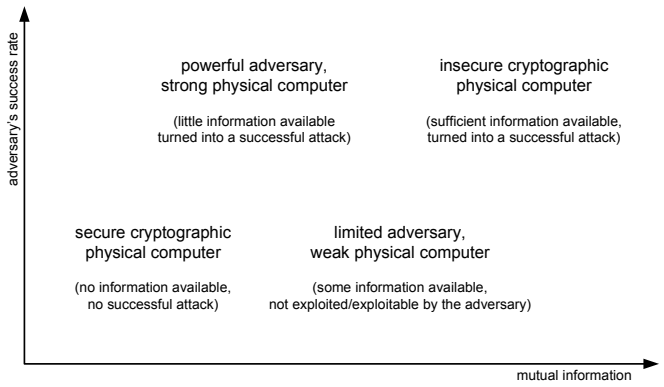

Fig. 1. Intuitive summary of side-channel evaluation criteria.

Indeed, let us consider a correct target key class s_g and a leakage vector $\mathbf{l}^q_{s_g}$. A Bayesian adversary having access to these leakages is successful if and only if:

$$s_g = \underset{s}{argmax} \ \Pr[S = s | \mathbf{L}^q_{s_g} = \mathbf{l}^q_{s_g}]$$
$$s_g = \underset{s}{argmax} \ \frac{\Pr[\mathbf{L}^q_{s_g} = \mathbf{l}^q_{s_g} | S = s] \cdot \Pr[S = s]}{\Pr[\mathbf{L}^q_{s_g} = \mathbf{l}^q_{s_g}]}$$

Assuming that the probabilities $\Pr[S = s]$ are equal and since $\Pr[\mathbf{L}^q_{s_g} = \mathbf{l}^q_{s_g}]$ is independent of s (it only depends on the correct class s_g), it directly yields:

$$s_g = \underset{s}{argmax} \ \Pr[\mathbf{L}^q_{s_g} = \mathbf{l}^q_{s_g} | S = s]$$

Then assuming independent leakages for the different queries, we find:

$$s_g = \underset{s}{argmax} \ \prod_{i=1}^{q} \Pr[L^i_{s_g} = l^i_{s_g} | S = s]$$

When considering an asymptotic attack, each query to the target physical computer determines a leakage trace $l^i_{s_g}$ picked up from a leakage distribution $\Pr[L^i_{s_g}]$. Therefore, an asymptotic attack is successful if and only if:

$$s_g = \underset{s}{argmax} \ \prod_{l^i_{s_g}} \Pr[L^i_{s_g} = l^i_{s_g} | S = s]^{\Pr[L^i_{s_g} = l^i_{s_g}]}$$
$$s_g = \underset{s}{argmax} \ \prod_{l^i_{s_g}} \Pr[S = s | L^i_{s_g} = l^i_{s_g}]^{\Pr[L^i_{s_g} = l^i_{s_g}]}$$
$$s_g = \underset{s}{argmax} \ \sum_{l^i_{s_g}} \Pr[L^i_{s_g} = l^i_{s_g}] \cdot -\log_2 \Pr[S = s | L^i_{s_g} = l^i_{s_g}] \tag{7}$$

If the previous condition holds for all classes s_g, the side-channel attack is asymptotically successful. Therefore, the leakage function is sound if and only if $argmax_s \, \mathbf{H}^{*1}_{s_g,s} = s_g$, $\forall s_g$. There are three important remarks:

1. q queries to a target device can be seen both as q realizations of a single query leakage vector $\mathbf{L}_{s_g}^1$ or as a single realization of a q-query leakage vector $\mathbf{L}_{s_g}^q$.
2. The condition on the entropy reduction matrix $\mathbf{H}_{s_g,s}^{*1}$ is stated for $q = 1$ since a leakage trace $l_{s_g}^i$ in Equation (7) corresponds to a single query vector $\mathbf{l}_{s_g}^1$. The condition for $q = 1$ involves the condition for any $q > 1$.
3. *Most importantly:* since the condition of independent leakages is conditional to the key classes s, it only requires that the noise in the observations is independent of these classes. With respect to the definition of a leakage function, it means that we assume $\mathsf{L}(C_\alpha, M, R) = \mathsf{L}'(C_\alpha, M) + \mathsf{L}''(R)$, *i.e.* the leakage function the sum of a deterministic part and a random part. We note that this condition is expected to hold to a sufficient degree for the relation between information and security to be meaningful in most applications.

3 Practice-oriented definitions

In this second section, we relate the previous formal definitions to intuitive physical notions. First, we translate the concept of physical computer into a concrete description of target circuit. Second, we refine the definition of a side-channel adversary in order to take the different steps of a practical attack into account. Finally, we discuss the use of the previously defined evaluation criteria for practical applications. The following three subsections can consequently be viewed as the practical counterparts of the previous ones.

3.1 From physical computers to actual implementations

From the definitions in Section 2.1, we propose to model a target abstract computer as the combination of signals and operations schematized in Figure 2. First, the set of all time-dependent signals in the circuit is denoted as:

$$\Sigma(t) = (\sigma_1(t), \sigma_2(t), \sigma_3(t), ..., \sigma_s(t))$$

Second, the cryptographic device can apply operations to the signals. A number of operations are actually included in the black box model. For example, if we consider block ciphers, a black box attacker could perform queries and obtain plaintext/ciphertext pairs. As a circuit could contain several such operations, we define the set of black box oracles β as the set of operations that can be queried in the black box model: $\beta = (\Omega_1, \Omega_2, ..., \Omega_b)$. In actual implementations, oracles are made of operations that cannot be queried by the black box attacker (but possibly by the side-channel one) because they apply to the circuit inner signals:

$$\Omega_i = (\omega_i^1, \omega_i^2, \omega_i^3, ..., \omega_i^{o_i})$$

We mention that we make no hypothesis about the actual form of the elementary operations ω_i^j's, although Figure 2 suggests that they represent logic gates. For clarity purposes, we represented our cryptographic implementation as a hardware circuit where every ω_i^j is physically implemented. However, in practice,

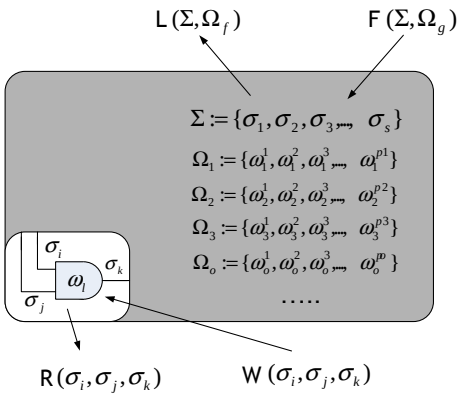

Fig. 2. Circuit model including physical threats.

different operations could be performed by the same hardware resource, *e.g.* in software-programmed processors, the ω_i^j's represent instructions applied sequentially to the signals rather than physical resources. Based on these definitions, we can consider different types of physical adversaries. An invasive probing attack gives read/write access to a limited subset of signals in the device (*i.e.* the functions R and W in the figure) [4]. A fault attack applies some probabilistic function F to the signals or operations [12]. The side-channel attacks we consider in this paper enhance the opponent with a leakage function L, *e.g.* [2, 19, 20].

It is important to observe that such a description can be efficiently translated into the formalism of [24]. The oracles Ω_i's can be simulated with abstract computers and the elementary operations ω_i^j's with VTMs. Signals are simply the inputs and outputs of the VTMs. In the following, we will denote *abstract computers* as *cryptographic primitives* and *physical computers* as *implementations*. The next section details this model for an exemplary target block cipher.

3.2 Exemplary target block cipher

A block cipher transforms a plaintext block x of a fixed bit length n_b into a ciphertext block y of the same length, under the influence of a cipher key k, of bit length n_k. We denote the forward operation of a block cipher as the encryption: $y = \mathsf{E}_k(x)$ and the reverse operation as the decryption: $x = \mathsf{E}_k^{-1}(y)$.

In practice, modern block ciphers are usually composed of several identical transforms, denoted as the encryption (*resp.* decryption) rounds. If such a cipher applies the same round function r times to the cipher state, it is necessary to expand the cipher key k into different round keys b_i's. This is done by means of a key round. The round and key round functions are respectively denoted as:

$$a_{i+1} = \mathsf{R}(a_i, b_i),$$
$$b_{j+1} = \mathsf{KR}(b_j),$$

where the a_i's represent the cipher states, with $a_0 = x$, $a_{r+1} = y$ and $b_0 = k$. Finally, we model our round and key round functions as made of 3 different operations: non-linear substitution, linear diffusion and bitwise XOR . Those are usual components of present block ciphers, *e.g.* the AES Rijndael [16].

More specifically, the substitution layer S consists of the parallel application of substitution boxes s to the b-bit blocks of the state:

$$\mathsf{S} : (\mathbb{Z}_{2^b})^{\frac{n_b}{b}} \rightarrow (\mathbb{Z}_{2^b})^{\frac{n_b}{b}} : x \rightarrow y = \mathsf{S}(x) \Leftrightarrow y^i = \mathsf{s}\big(x^i\big), \quad 0 \leq i \leq \frac{n_b}{b} - 1,$$

where x^i is the ith b-bit block of the state vector x. The small S-boxes s are assumed to have good non-linearity, differential profile, non-linear order *etc.*

The linear diffusion layer D applies to the whole state and is assumed to have good diffusion properties (*e.g.* avalanche effect, high branch number, *etc.*):

$$\mathsf{D} : \mathbb{Z}_{2^{n_b}} \rightarrow \mathbb{Z}_{2^{n_b}} : x \rightarrow y = \mathsf{D}(x)$$

Finally, the bitwise XOR layer $\oplus$ is denoted as:

$$\oplus : \mathbb{Z}_2^{n_b} \times \mathbb{Z}_2^{n_b} \rightarrow \mathbb{Z}_2^{n_b} : x, y \rightarrow z \Leftrightarrow z(i) = x(i) \oplus y(i), \quad 0 \leq i \leq n_b - 1$$

where $x(i)$ is the ith bit of the state vector x. For example, a 3-round block cipher, is represented in Figure 3. With respect to the model of Section 3.1, the complete block cipher is an oracle E_K and a possible division in elementary operations would be $\mathsf{E}_k := \{\mathsf{R}_1, \mathsf{R}_2, \mathsf{R}_3, \mathsf{KR}_1, \mathsf{KR}_2, \mathsf{KR}_3\}$. Another division (with smaller operations) is $\mathsf{E}_k := \{\oplus_1, ..., \oplus_4, \mathsf{S_A}, ..., \mathsf{S_F}, \mathsf{D}_1, ..., \mathsf{D}_6\}$.

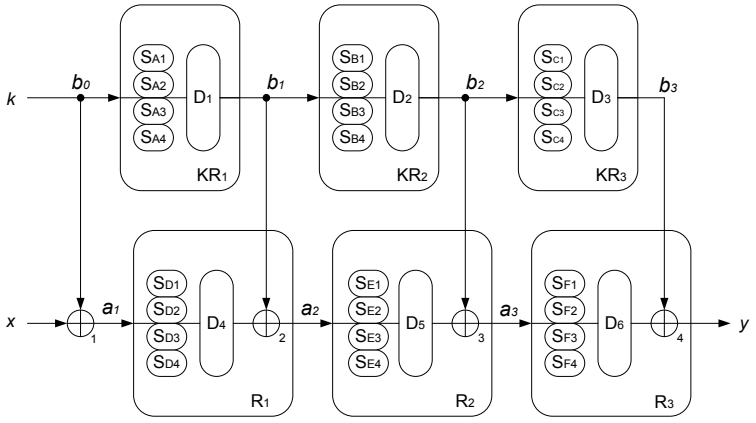

Fig. 3. A 3-round block cipher

3.3 Detailed description of a side-channel adversary

From the definition of Section 2.2, a side-channel key recovery adversary is defined as an algorithm trying to recover a key class s_g from a number of queries to an implementation $(\mathsf{E}_K, \mathsf{L})$. In this section, we aim to give a more detailed description of such an adversary, considering the different steps in the side-channel attack illustrated in Figure 4. It actually consists in two phases that we respectively denote as the exploitation phase (which is the main core of the attack) and the preparation phase (which the counterpart of the learning phase in artificial intelligence problems). We first describe the exploitation phase:

1. Input selection: the adversary selects its (possibly adaptive) q queries $\mathbf{x}_q$ (defined in Section 2.3) to the target device thanks to an algorithm I.
2. Values derivation: for each key class s, the adversary predicts some values within the computer's internal configuration with an algorithm V. As a result, it obtains N_s vectors $\mathbf{v}_s^q = \mathsf{V}(s, \mathbf{x}_q)$ containing N_v-element predictions v_s^i, $i \in [1, q]$, where N_v is the number of internal values predicted per query.
3. Leakages modeling: for each key class candidate, the adversary models a part/function of the actual leakages in the target device. Depending on the attack context, the model is either the multivariate probability density function of the leakages $\tilde{\mathbf{l}}^q$ generated by a key class candidate: $\mathsf{M}(s, \tilde{\mathbf{l}}^q)$, as in template-like attacks [8, 26]. In this context, $\tilde{\mathbf{l}}^q = [\tilde{l}^1, \tilde{l}^2, \ldots, \tilde{l}^q]$ is the vector of leakage samples that are actually modeled by the adversary and $\tilde{l}^i$ is an N_m-element trace corresponding to the i^{th} query to the target device (N_m is the number of samples modeled per query). Or the model is a deterministic function (*e.g.* Hamming weight) of the previously defined values: $\mathsf{M}(s, \mathbf{v}_s^q)$.
4. Leakages observation (or measurement): the adversary monitors the leakages of the target device containing the correct key class s_g. It stores these observations in the previously defined vector $\mathbf{l}_{s_g}^q$ containing N_l-sample traces $l_{s_g}^i$, $i \in [1, q]$, where N_l is the number of leakage samples stored per query.
5. Leakages transform: since the leakages and predictions possibly have different number of samples $N_l \neq N_m$, a transform T is used to transform the leakages such that $\mathsf{T}(l_{s_g}^i)$ is a N_m-sample trace. Additionally, the mapping possibly includes the post-processing of the traces, *e.g.* filtering, averaging.
6. Statistical comparison: for each key class s, the adversary applies a statistic C to compare the the model $\mathsf{M}(s, .)$ with the transformed leakages. It obtains a N_s-element vector $\mathbf{c}_s^q = \mathsf{C}(\mathsf{M}(s, .), \mathsf{T}(\mathbf{l}_{s_g}^q))$ containing the comparison result.
7. Decision: from the previous result, the adversary selects a key candidate (*i.e.* does a hard decision) or a list of key candidates (*i.e.* does soft decision) and stores them in a N_d-element vector $\mathbf{d}_s^q = \mathsf{D}(\mathbf{c}_s^q)$.
8. Offline computation: if a soft strategy is applied, the adversary finally tests the remaining candidates by a number of executions of the target algorithm.

Different illustrations of how the description of practical side-channel attacks can be integrated within this framework can be found in [14].

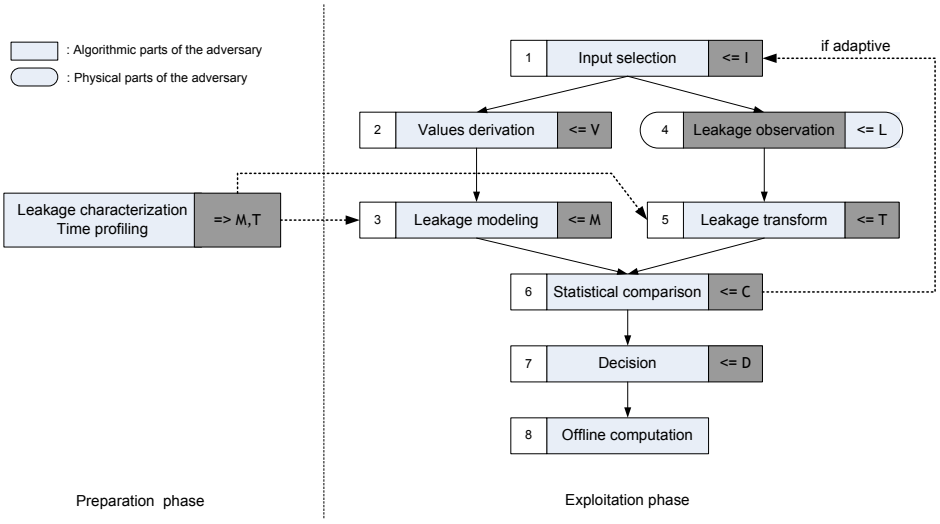

Fig. 4. Practical side-channel adversary.

Preliminary to the exploitation phase, the preparation phase produces the leakage model M and transform T, *e.g.* by profiling and characterizing the device. As a matter of fact, deriving these functions may involve the same steps as the exploitation phase. But since the preparation can be performed once and then used in several exploitations, it is interesting to separate the complexities for both phases: $\tau{=}\tau_P{+}\tau_E$, $m{=}m_P{+}m_E$, $q{=}q_P{+}q_E$. We note that one could also design attacks in which preparation and exploitation are closely connected and therefore, only the overall complexities are relevant.

Importantly, a side-channel adversary is composed of a physical part modeled by the leakage function and an algorithmic part modeled by all the steps but the 4^{th} in Figure 4. Since the definition of a leakage function in [24] includes measurement setups, it depends on the actual adversary's ability to perform good measurements. Therefore, in this practice-oriented view, the leakage function is not an oracle accessed by the adversary (as in the more theoretical view of Section 2.2) but a part of it. But this is not in contradiction with Micali and Reyzin: once the leakage function has been determined, the algorithmic part of the adversary in Figure 4 can theoretically access it as an oracle.

3.4 Practical limits & convenient features of the evaluation criteria

1. Computing the entropy. An important fact related to the definition of a leakage function is that in practice, it is only known through its physical representation (*e.g.* the power measurement of a smart card). Therefore, the probability density function $\Pr[S = s|\mathbf{L}^q_{s_g} = \mathbf{l}^q_{s_g}]$ necessary to evaluate the entropy is generally only known through a number of samples obtained with an acquisition device, *e.g.* an oscilloscope. Since the dimensions of a leakage trace (*e.g.* the one

illustrated in Appendix D, Figure 6) are generally too large (*e.g.* $N_l \propto 10^5$) for $\Pr[S = s|\mathbf{L}_{s_g}^q = \mathbf{l}_{s_g}^q]$ to be tractable, actual adversaries use a transform T to reduce this dimensionality to smaller values N_m. As a consequence, the information leakages in a side-channel attack can only be approximated with reasonable heuristics, *e.g.* template attacks [5, 8] or stochastic models [18, 26]. "How to best approximate and exploit high dimension physical leakages" is an open question.

2. Detecting sound leakage functions. As mentioned in Section 2.3, an interesting property of the entropy reduction matrix is that it allows to detect sound leakage functions. A consequence of the previous remark is that the quality of this soundness detection depends on the extent to which the leakages probabilities and therefore the entropy can be properly estimated.

3. Finding correlations between key classes. Another interesting property of the entropy matrix $\mathbf{H}_{s_g,s}^q$ is that they directly allow evaluating possible correlations between a correct key class s_g and any class s. Note that a similar confusion matrix can be built for the security metric as well (see Appendix E).

4 Evaluation methodology

In this section, we finally turn our previous definitions into an evaluation methodology for side-channel attacks. It is summarized in Figure 5 and holds in five steps:

1. We define the target implementation as modeled by Micali and Reyzin. That is, we define the combination of an abstract computer and a leakage function. In practice, the target implementation is a physical object, *e.g.* a smart card, FPGA or ASIC running some cryptographic primitive associated with some measurement setup. This determines the physical part of the adversary.
2. We define the target secret class s for the side-channel attack.
3. Once the target has been specified, we answer the first question in our evaluation, namely: "*What is the amount of information contained in the physical observations obtained from a leaking device?*". For this purpose, we use the mutual information. As mentioned in Section 3.4 and pictured on the figure arrows, in practice it can only be approximated from a number of leakage samples, through an actual (template-like) adversary's measurements. Alternatively and as a preliminary step in the evaluations, it can also be estimated theoretically by using a model $\mathsf{M}(s, .)$ (*e.g.* Hamming weight-based, SPICE simulations, ...) in place of the actual leakage function.
4. We complete the definition of the side-channel adversary (*i.e.* with its algorithmic part mainly) and specify its black box adversarial context and physical potential. The latter are actually the most delicate part of an evaluation and will be shortly discussed in the following of the section.
5. We finally answer the second question in our evaluation, namely: "*How successfully can an adversary turn this information into a practical attack?*". For this purpose, we use the advantage of the side-channel key recovery adversary (or its success rate) defined in Section 2.3.

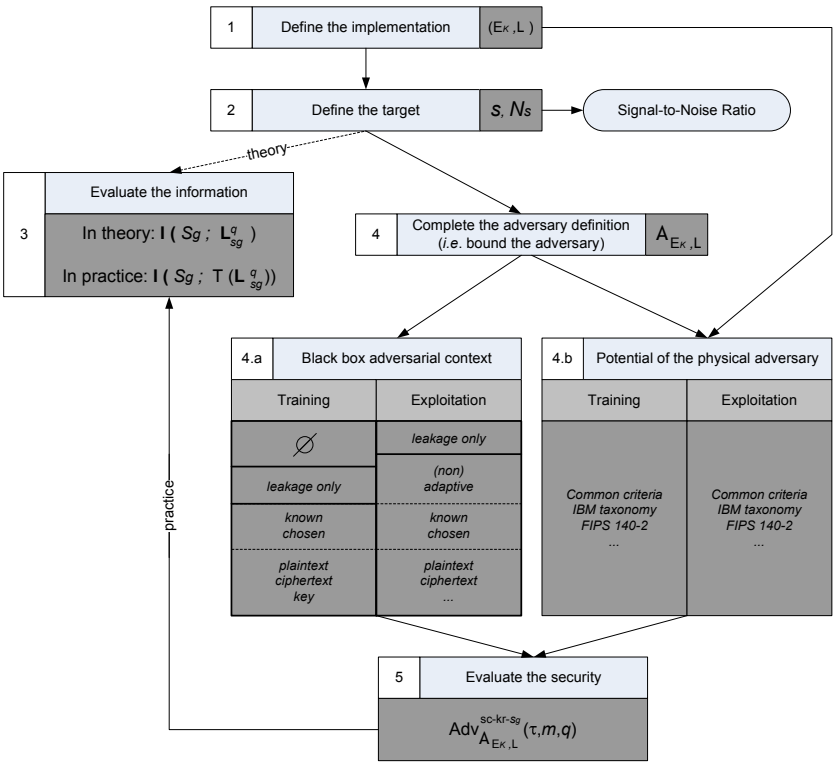

Fig. 5. Evaluation methodology for side-channel attacks.

Figure 5 typically indicates that the information theoretic metric can be used to measure an adversary's physical part while the security metric is rather useful to evaluate its algorithmic part. Otherwise said, information can discriminate different implementations while security can discriminate different adversaries, for a given implementation. Intuitively, these criteria have a direct counterpart in communication theory. Namely, the mutual information measures the maximum information that can be exploited by a side-channel adversary (*i.e.* what is available on the channel) while the success rate measures the efficiency of this adversary, just as the Bit-Error-Rate does in communication problems [28].

Additionally to these evaluation criteria, it is often interesting to define a Signal-to-Noise Ratio (SNR) in order to determine the *fraction of useful signal in the side-channel observations*. For example, an SNR was defined in [22] as the ratio between the leakage (*e.g.* the power consumption) caused by the attacked intermediate result in an implementation and the additive noise. Although such an SNR may be independent of the previously discussed information and security issues in certain contexts (*e.g.* in Appendix F), it can be used to plot the information theoretic and security evaluation metrics with its respect.

As already mentioned, the most delicate part in this methodology is to describe the black box adversarial context and physical potential of a side-channel adversary. Black box assumptions relate to the adversary's abilities to monitor and tamper with the primitives inputs and outputs. For this purpose, we refer to classical notions (*e.g.* non adaptive/adaptive, known/chosen, plaintext/ciphertext, ...) with the additional possibility to have known or chosen keys during the training phase. The physical potential of an adversary relates to its level of expertise, the cost of its equipment, ... Since quantifying such potential is typically the tasks assigned the standardization bodies, we refer to the common criteria [9] and FIPS 140-2 documents [15] (or alternatively to the IBM taxonomy [1]) for these purposes. In general, the benefit of the presently introduced model is not to solve these practical issues but to state the side-channel problem in a sound framework for its analysis. Namely, it is expected that the proposed security and information theoretic metrics can be used for the fair analysis, evaluation and comparison of any physical implementation or countermeasure against any type of side-channel attack.

We finally mention that in a number of applications, the information cannot be evaluated properly since the probability densities $\Pr[S = s|\mathbf{L}_{s_g}^q = \mathbf{l}_{s_g}^q]$ are not accessible. In these contexts, only the security evaluation can be conducted. Otherwise said, template attacks are not always practically achievable.

5 Conclusion and open problems

A formal practice-oriented model for the analysis of cryptographic implementations against side-channel attacks is introduced as a specialization of Micali and Reyzin's "physically observable cryptography" paradigm [24]. It is based on a theoretical framework in which the effect of practically relevant leakage functions is evaluated with a combination of security and information theoretic metrics. The model allows both the practical evaluation of actual side-channel attacks and the understanding of the underlying tradeoffs in physically observable cryptography, namely "*flexibility vs. efficiency*" and "*information vs. computation*".

The flexibility *vs.* efficiency tradeoff typically relates to the adversarial context considered. As a matter of fact, an adaptive adversary using a carefully profiled leakage prediction function will generally recover (much) more information from side-channel measurements than a non-adaptive one, using a non profiled leakage prediction. However, simpler models do not only involve a sub-optimal information extraction from side-channel traces. They may also be more easily reproducible to different devices. As a typical example, a DPA only assumes that somewhere in a physical observation, the leakage will depend on a single bit value. The simplicity of this assumption made it straightforwardly applicable to a wide range of devices, without any adaptation. Correlation attacks [7], template attacks [8] or stochastic models [26] are trading some of this flexibility for a more efficient information extraction.

The information *vs.* computation tradeoff rather relates to the decision strategy considered. As a matter of fact, for comparable amounts of side-channel queries q, a soft strategy trying to extract a list of key candidates including the correct one will generally have a higher success rate than a hard strategy, trying to extract the correct key value only. However, if this list of candidates can be tested with some computational power, it can be turned into a successful key recovery. Otherwise said, a lack of information can be overcome by a more computationally intensive adversarial strategy.

Open questions derive from this model in different directions. A first one relates the best exploitation of side-channel leakages, *i.e.* to the construction of (ideally) optimal side-channel adversaries. This requires to investigate the best heuristics to deal with high dimensional leakages data. A second one relates to the investigations of stronger security notions than side-channel key recovery. That is, the different security notions considered in the black box model (*e.g.* the undistinguishability from random permutations for block ciphers) should be considered in the physical world, as initiated in [24]. A third directions relates to the construction of implementations with provable (or arguable) security against side-channel attacks, *e.g.* as proposed in [25]. We note that proving the security of an implementation assumes a given measurement setup (that is included in the definition of a leakage function). An even better (but hardly achievable) result would be to prove the security independently of the measurement parameter in the leakage function. A fourth direction would be the evaluation of actual side-channel attacks and countermeasures within the model in different implementation contexts, in particular those for which the security evaluation remains an open question. A first example of application was given in [27]. Similarly, applying our information theoretic metric to the context of dual rail pre-charge logic styles, *e.g.* [29] in order to discriminate different implementations would improve their evaluation. Such evaluations have been initiated in [21] with simulated leakages an need to be extended to actual measurements. Finally the extension to other physical adversaries (*e.g.* fault-based) in order to unify all physical adversaries is a long term scope for cryptographic research.

Acknowledgements: The authors would like to thank Christophe Petit and Leonid Reyzin for their valuable contributions to the understanding of the issues discussed in this paper. We also thank Cédric Archambeau, Nenad Dedic, Marc Joye, François Koeune, Stefan Mangard, Elisabeth Oswald, Eric Peeters, Gilles Piret, Emmanuel Prouff, anonymous reviewers and the ECRYPT project partners for interesting discussions about preliminary versions of this work.

References

1. D.G. Abraham, G.M. Dolan, G.P. Double, J.V. Stevens, *Transaction Security System*, in IBM Systems Journal, vol 30, num 2, pp 206- 229, 1991.
2. D. Agrawal, B. Archambeault, J. Rao, P. Rohatgi, *The EM Side-Channel(s)*, in the proceedings of CHES 2002, Lecture Notes in Computer Science, vol 2523, pp 29-45, Redwood City, California, USA, August 2002.
3. D. Agrawal, J. Rao, P. Rohatgi, *Multi-channel Attacks*, in the proceedings of CHES 2003, LNCS, vol 2779, pp 2-16, Cologne, Germany, Sept. 2003.
4. R. Anderson, M. Kuhn, *Tamper Resistance - a Cautionary Note*, in the proceedings of the USENIX Workshop on Electronic Commerce, pp 1-11, Oakland, California, USA, November 1996.
5. C. Archambeau, E. Peeters, F.-X. Standaert, J.-J. Quisquater, *Template Attacks in Principal Subspaces*, in the proceedings of CHES 2006, Lecture Notes in Computer Science, vol 4249, pp. 1–14, Yokohama, Japan, October 2006.
6. A. Biryukov, C. De Cannière, M. Quisquater, *On Multiple Linear Approximations*, in the proceedings of Crypto 2004, Lecture Notes in Computer Science, vol 3152, pp 1-22, Santa Barbara, California, USA, August 2004.
7. E. Brier, C. Clavier, F. Olivier, *Correlation Power Analysis with a Leakage Model*, in the proceedings of CHES 2004, Lecture Notes in Computer Science, vol 3156, pp 16-29, Boston, Massachusetts, USA, August 2004.
8. S. Chari, J. Rao, P. Rohatgi, *Template Attacks*, in the proceedings of CHES 2002, Lecture Notes in Computer Science, vol 2523, pp 13-28, CA, USA, August 2002.
9. *Application of Attack Potential to Smart Cards*, Common Criteria Supporting Document, Version 1.1, July 2002, http://www.commoncriteriaportal.org
10. J.S. Coron, P. Kocher, D. Naccache, *Statistics and Secret Leakage*, in the proceedings of Financial Crypto 2000, Lecture Notes in Computer Science, vol 1972, pp 157-173, Anguilla, British West Indies, February 2000.
11. T.M. Cover, J.A. Thomas, *Information Theory*, Wiley and Sons, New York, 1991.
12. D. Boneh, R. De Millo, R. Lipton, *On the Importance of Checking Cryptographic Protocols for Faults*, proceedings of Eurocrypt 1997, Lecture Notes in Computer Science, vol 1233, pp 37-51, Konstanz, Germany, May 1997.
13. ECRYPT Network of Excellence in Cryptology, *The Side-Channel Cryptanalysis Lounge* , http://www.crypto.ruhr-uni-bochum.de/en_sclounge.html.
14. ECRYPT Network of Excellence in Cryptology, *Intermediate Report on the Application of a Theoretical Model for the Analysis of Side-Channel Attacks to a Suite of Algorithms*, available from http://www.dice.ucl.ac.be/ fstandae/tsca.
15. FIPS 140-2, *Security Requirements for Cryptographic Modules*, Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, December 3, 2002.
16. FIPS 197, *"Advanced Encryption Standard,"* Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, November 26, 2001.
17. R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, T. Rabin, *Algorithmic Tamper-Proof Security: Theoretical Foundations for Security Against Hardware Tampering*, in the proceedings of TCC 2004, Lecture Notes in Computer Science, vol 2951, pp 258-277, Cambridge, MA, USA, February 2004.
18. B. Gierlichs, K. Lemke, C. Paar, *Templates vs. Stochastic Methods*, in the proceedings of CHES 2006, Lecture Notes in Computer Science, vol 4249, pp 15-29, Yokohama, Japan, October 2006.
19. P. Kocher, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems*, in the proceedings of Crypto 1996, Lecture Notes in Computer Science, vol 1109, pp 104-113, Santa Barbara, California, USA, August 1996.

20. P. Kocher, J. Jaffe, B. Jun, *Differential Power Analysis*, in the proceedings of Crypto 1999, Lecture Notes in Computer Science, vol 1666, pp 398-412, Santa-Barbara, California, USA, August 1999.
21. F. Macé, F.-X. Standaert, J.-J. Quisquater, *Information Theoretic Evaluation of Side-Channel Resistant Logic Styles*, to appear in the proceedings of CHES 2007.
22. S. Mangard, *Hardware Countermeasures against DPA - A Statistical Analysis of Their Effectiveness*, in the proceedings of CT-RSA 2004, Lecture Notes in Computer Science, vol 2964, pp 222-235, San Francisco, CA, USA, February 2004.
23. T.S. Messerges, E.A. Dabbish, R.H. Sloan, *Examining Smart-Card Security under the Threat of Power Analysis Attacks*, IEEE Transactions on Computers, vol 51, num 5, pp 541-552, May 2002.
24. S. Micali, L. Reyzin, *Physically Observable Cryptography*, in the proceedings of TCC 2004, Lecture Notes in Computer Science, vol 2951, pp 278-296, Cambridge, Massachusetts, USA, February 2004.
25. C. Petit, F.-X. Standaert, O. Pereira, T.G. Malkin, M. Yung, *A Block Cipher based PRNG Secure Against Side-Channel Key Recovery*, available from http://www.dice.ucl.ac.be/ fstandae/tsca.
26. W. Schindler, K. Lemke, C. Paar, *A Stochastic Model for Differential Side-Channel Cryptanalysis*, in the proceedings of CHES 2005, Lecture Notes in Computer Science, vol 3659, pp 30-46, Edinburgh, Scotland, September 2005.
27. F.-X. Standaert, E. Peeters, C. Archambeau, J.-J. Quisquater, *Towards Security Limits in Side-Channel Attacks*, in the proceedings of CHES 2006, Lecture Notes in Computer Science, vol 4249, pp. 30–45, Yokohama, Japan, October 2006.
28. F.-X. Standaert, *Testing Integrated Circuits Against Side-Channel Attacks*, available from http://www.dice.ucl.ac.be/ fstandae/tsca.
29. K. Tiri, M. Akmal, I. Verbauwhede, *A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards*, in the proceedings of ESSCIRC 2003.

A Notation index

In general and excepted if explicitly mentioned otherwise, capital letters represent variables X, small letters represent particular values of the variables x and sets or alphabets are denoted with calligraphic letters $\mathcal{X}$. Bold letters denote vectors and matrices $\mathbf{X}$. Sans serif fonts are used for algorithms and functions X,x. Finally, Greek letters represent abstract computers, physical computers and their practical counterparts (*i.e.* signals and operations).

α	Abstract computer/cryptographic primitive	pp 3, sec 2.1
α_i	Virtual Memory Turing Machine (VTM)	pp 3, sec 2.1
$\mathbf{Adv}^{\text{sc-kr-}\delta(K)}_{\mathsf{A}_{\mathsf{E}_K},\mathsf{L}}$	Advantage of a side-channel key recovery adversary $\mathsf{A}_{\mathsf{E}_K,\mathsf{L}}$ against an implementation $(\mathsf{E}_K,\mathsf{L})$	pp 4, sec 2.2
$\mathbf{Adv}^{\text{sc-kr-}\delta(K)}_{\mathsf{E}_K,\mathsf{L}}$	Advantage of an implementation $(\mathsf{E}_K,\mathsf{L})$ against	pp 4, sec 2.2
$\mathsf{A}_{\mathsf{E}_K,\mathsf{L}}$	Side-channel adversary side-channel key recovery adversaries	pp 4, sec 2.2
C	Statistical comparison algorithm	pp 10, sec 3.3
$\mathbf{C}^q_{s_g,s}$	Confusion matrix for the success rate	pp 22, app E
δ	Key classification function	pp 3, sec 2.2
D	Decision function	pp 10, sec 3.3
E_K	Family of cryptographic abstract computers indexed by a variable key K	pp 3, sec 2.2
F	Fault insertion function	pp 8, sec 3.1
$\mathbf{H}^q_{s_g,s}$	Conditional entropy matrix	pp 5, sec 2.3
$\mathbf{H}^{*q}_{s_g,s}$	Entropy reduction matrix	pp 5, sec 2.3
$\mathrm{H}[S_g\|\mathbf{L}^q_{s_g}]$	Conditional entropy	pp 5, sec 2.3
I	Input selection algorithm	pp 10, sec 3.3
$\mathrm{I}(S_g;\mathbf{L}^q_{s_g})$	Mutual information	pp 5, sec 2.3
$\mathsf{L}(C_\alpha, M, R)$	Leakage function	pp 3, sec 2.1
$\mathbf{L}^q_{s_g}$, $\mathbf{l}^q_{s_g}$	Side-channel observations vector	pp 5, sec 2.3
$\mathsf{M}(s,.)$	Leakage model for a key class s	pp 10, sec 3.3
Ω_i	Black box oracle in a target implementation	pp 7, sec 3.1
ω^j_i	Elementary operation in a target implementation	pp 7, sec 3.1
φ	Physical computer/cryptographic implementation	pp 3, sec 2.1
φ_i	Physical Virtual Memory Turing Machine	pp 3, sec 2.1
$\Pr[s\|\mathbf{l}^q_{s_g}]$	Probability of a key class s given a leakage $\mathbf{l}^q_{s_g}$	pp 5, sec 2.3
R	Probing read function	pp 8, sec 3.1
Σ	Set of signals in a target implementation	pp 7, sec 3.1
σ_i	Individual signal in a target implementation	pp 7, sec 3.1
$\mathbf{Succ}^{\text{sc-kr-}\delta(K)}_{\mathsf{A}_{\mathsf{E}_K},\mathsf{L}}$	Success rate of a side-channel key recovery adversary $\mathsf{A}_{\mathsf{E}_K,\mathsf{L}}$ against an implementation $(\mathsf{E}_K,\mathsf{L})$	pp 4, sec 2.2
T	Leakage transform	pp 10, sec 3.3
V	Values derivation algorithm	pp 10, sec 3.3
W	Probing write function	pp 8, sec 3.1

B Micali & Reyzin's informal axioms

Axiom 1. Computation and only computation leaks information.

That is, we assume that it is possible to store some secret information securely in a cryptographic device. No leakages will compromise this secret as long as it is not used in any computation. This implies that probing attacks are out of the scope of our analysis and we rely on physical protections to prevent them.

Axiom 2. The same computation leaks different information on different computers.

In other words, an algorithm is an abstraction: a set of general instructions whose physical implementation may vary. As a result, the same elementary operation may leak different information on different platforms.

Axiom 3. The information leakage depends on the chosen measurement.

The amount of information that is recovered by an adversary during a side-channel attack depends on the measurement process, that possibly introduces some randomness due to the presence of noise.

Axiom 4. The information leakage is local.

In other words, the maximum amount of information that may be leaked by a physically observable device is the same in any execution of the algorithm with the same inputs, since it relates to the target device's internal configuration.

Axiom 5. All the information leaked through physical observations can be efficiently computed from a target device's internal configuration.

That is, given a physical computer, the information leakage is a polynomial time computable function of (1) the computer's internal configuration (because of Axiom 4), (2) the chosen measurement (because of Axiom 3), and possibly (3) some randomness outside anybodys control (also because of Axiom 3).

We note that, from the practical point of view, these axioms may not reflect the entire physical phenomenons observed. For example, as far as Axiom 1 is concerned, volatile memories such as RAMs regularly require a small amount of energy to refresh their values and this could be used to mount a side-channel attack. However, such leakages are significantly more difficult to exploit than computational leakages. Our expectation is therefore that these axioms approximates the physical reality to a sufficient degree.

C Equivalent definition of conditional entropy

In classical textbooks, *e.g.* [11], the entropy of a random variable X is defined as:

$$\mathrm{H}[X] = \sum_x \Pr[x] \cdot -\log_2(\Pr[x])$$

And the conditional entropy:

$$\begin{aligned}\mathrm{H}[Y|X] &= \sum_x \Pr[x] \cdot \mathrm{H}[Y|X=x] \\ &= \sum_x \Pr[x] \cdot \sum_y \Pr[y|x] \cdot -\log_2[\Pr(y|x)]\end{aligned}$$

Taking the notations of Section 2.3, it yields:

$$\begin{aligned}\mathrm{H}[S_g|\mathbf{L}^q_{s_g}] &= \textstyle\sum_{\mathbf{l}^q_{s_g}} \Pr[\mathbf{l}^q_{s_g}] \cdot \mathrm{H}[S_g|\mathbf{L}^q_{s_g} = \mathbf{l}^q_{s_g}] \\ &= \textstyle\sum_{\mathbf{l}^q_{s_g}} \Pr[\mathbf{l}^q_{s_g}] \ \sum_{s_g} \Pr[s_g|\mathbf{l}^q_{s_g}] \cdot -\log_2(\Pr[s_g|\mathbf{l}^q_{s_g}]) \\ &= \textstyle\sum_{\mathbf{l}^q_{s_g}} \Pr[\mathbf{l}^q_{s_g}] \ \sum_{s_g} \frac{\Pr[\mathbf{l}^q_{s_g}|s_g]\cdot\Pr[s_g]}{\Pr[\mathbf{l}^q_{s_g}]} \cdot -\log_2(\Pr[s_g|\mathbf{l}^q_{s_g}]) \\ &= \textstyle\sum_{\mathbf{l}^q_{s_g}} \sum_{s_g} \Pr[\mathbf{l}^q_{s_g}|s_g] \cdot \Pr[s_g] \cdot -\log_2(\Pr[s_g|\mathbf{l}^q_{s_g}]) \\ &= \textstyle\sum_{s_g} \sum_{\mathbf{l}^q_{s_g}} \Pr[\mathbf{l}^q_{s_g}|s_g] \cdot \Pr[s_g] \cdot -\log_2(\Pr[s_g|\mathbf{l}^q_{s_g}]) \\ &= \textstyle\sum_{s_g} \Pr[s_g] \ \sum_{\mathbf{l}^q_{s_g}} \Pr[\mathbf{l}^q_{s_g}|s_g] \cdot -\log_2(\Pr[s_g|\mathbf{l}^q_{s_g}]) \\ &= \textstyle\sum_{s_g} \Pr[s_g] \cdot \mathbf{H}^q_{s_g,s_g} = \mathbf{E}_{s_g} \ \mathbf{H}^q_{s_g,s_g}\end{aligned}$$

It is therefore equivalent to our definition using the entropy matrix.

D Exemplary leakage traces

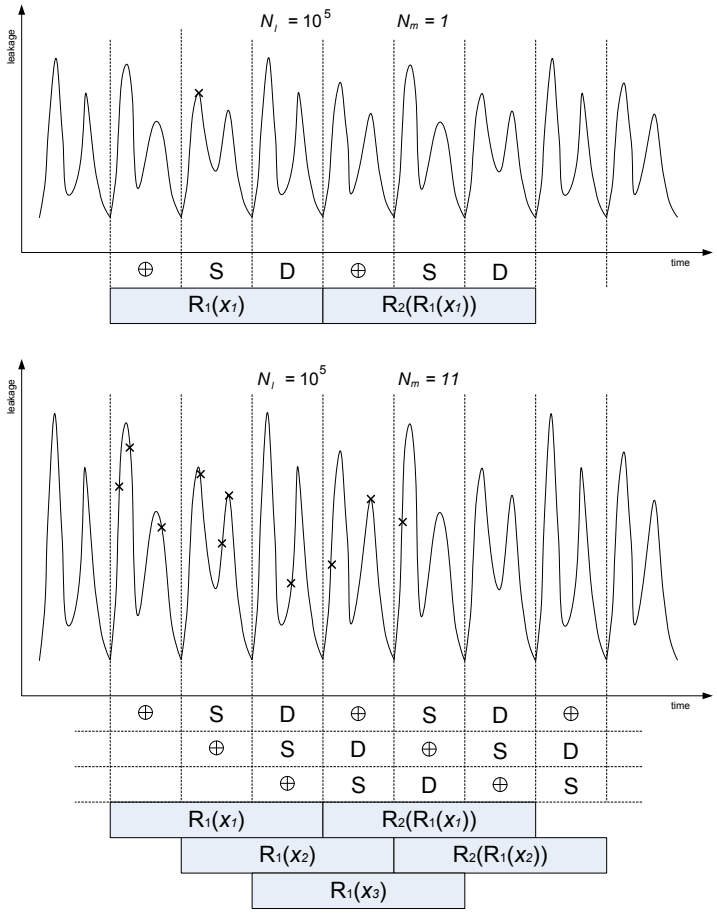

Fig. 6. Exemplary leakage traces of the 3-round cipher of Fig. 3: (up) serial implementation, univariate leakage model, (down) pipeline implementation, multivariate model.

E Confusion matrix for the success rate

Using the notations of Sections 3.3, let us assume an adversary with a hard decision receiving various leakages $\mathbf{l}^q_{s_g}$ that would select the following keys:

$$\mathbf{d}^q_s = \{\hat{s} \mid \hat{s} = \underset{s}{argmax}\ \mathsf{C}(\mathsf{M}(s,.), \mathsf{T}(\mathbf{l}^q_{s_g}))\},$$

If we store the result of the attack for any leakage in an index matrix:

$$\mathbf{I}^q_{s_g,s} = \quad \tfrac{1}{|\mathbf{d}^q_s|} \textbf{ if } s \in \mathbf{d}^q_s, \quad \textbf{else } 0,$$

we can define the success rate of the adversary after q queries:

$$\mathbf{Succ}^{\text{sc-kr-}S_g}_{\mathsf{A}_{f_k,\mathsf{L}}}(q) = \mathop{\mathbf{E}}_{s_g} \mathop{\mathbf{E}}_{\mathbf{l}^q_{s_g}} \mathbf{I}^q_{s_g,s_g} \tag{8}$$

Note that the success rate not only depends on the number of queries q but also on the adversary's time and memory complexities τ, m that are omitted in the formula for clarity reasons. Finally, we can use the complete index matrix to build a confusion matrix:

$$\mathbf{C}^q_{s_g,s} = \mathop{\mathbf{E}}_{\mathbf{l}^q_{s_g}} \mathbf{I}^q_{s_g,s} \tag{9}$$

The success rate simply corresponds to the averaged diagonal of $\mathbf{C}^q_{s_g,s}$.

F Limitations of an exemplary Signal to Noise Ratio

The aim of the signal to noise ratio defined in [22] is to determine the fraction of useful signal in an implementation, no matter if it contains information. For example, in a power analysis attack, it is defined as the ratio between the power consumption caused by the attacked intermediate result Q in an implementation and the additive noise N. It was initially introduced to measure the efficiency of side-channel attacks using the correlation coefficient. Since DC components are not relevant for the computation of this coefficient, only the variance of the signals were considered in the definition:

$$\mathrm{SNR} = \frac{\mathrm{var}(Q)}{\mathrm{var}(N)} \tag{10}$$

We illustrate this definition with the left implementation of Figure 7, in a Hamming weight leakage model. For simplicity, we assume that only the values outside the grey boxes are leaking. The figure illustrates a context where an adversary targets a b-bit S-box that is affected by $3b$ random bits of noise. It typically corresponds to a side-channel attack against a block cipher where the adversary targets one S-box out of four. Consequently, the outputs of the un-targeted S-boxes produce what is usually referred to as algorithmic noise, approximated by the random bits r_1, r_2, r_3. Since we consider a Hamming weight model, the variances of the leakages are easily calculated. Namely the mean Hamming weigh of an n-bit random value is $n/2$ and its variance $n/4$. Therefore, the SNR of the example in Figure 7 is worth $\frac{b/4}{3b/4} = \frac{1}{3}$.

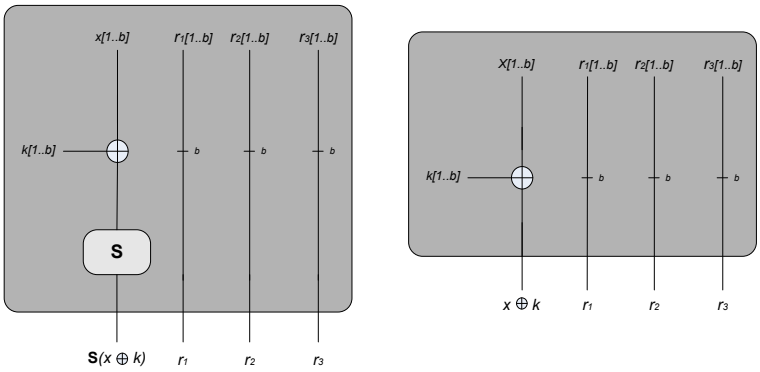

Fig. 7. Illustrative implementation with SNR=1/3.

We can easily observe that this SNR is not an information theoretic metric nor a security metric in itself with the following example. Let us consider the right scheme of Figure 7 in the Hamming distance leakage model. Clearly, the SNR of this example is again $\frac{1}{3}$ while it does not leak any key information. Indeed $W_H(x_1 \oplus k \oplus x_2 \oplus k)$ does not depend on the key. In this example the SNR is independent of the leakage function and statistical tool used by the adversary.