The Indiscernibility Methodology: quantifying information leakage from side-channels with no prior knowledge

Yoann Marquer, Olivier Zendra, Annelie Heuser

The Indiscernibility Methodology: quantifying information leakage from side-channels with no prior knowledge[*]

Yoann Marquer[1][†], Olivier Zendra[2][‡], Annelie Heuser[2][§]
[1]*SVV, SnT, University of Luxemburg, Luxemburg*
[2]*Inria, Univ. Rennes, CNRS, IRISA, France*

September 30, 2022

Abstract

Cyber security threats are important and growing issues in computing systems nowadays. Among them are the side-channel attacks, made possible by information leaking from computing systems through non-functional properties like execution time, consumed energy, power profiles, etc. These attacks are especially difficult to protect from, since they rely on physical measurements not usually envisioned when designing the functional properties of a program. Furthermore, countermeasures are usually dedicated to protect a particular program against a particular attack, lacking universality.

To help fight these threats, we propose in this paper the Indiscernibility Methodology, a novel methodology to quantify with no prior knowledge the information leaked from programs, thus providing the developer with valuable security metrics, derived either from topology or from information theory. Our original approach considers the code to be analyzed as a completely black box, only the public inputs and leakages being observed. It can be applied to various types of side-channel leakages: time, energy, power, EM, etc.

In this paper, we first present our Indiscernibility Methodology, including channels of information and our threat model. We then detail the computation of our novel metrics, with strong formal foundations based both on topological security (with distances defined between secret-dependent observations) and on information theory (quantifying the remaining secret information after observation by the attacker). Then we demonstrate the applicability of our approach by providing experimental results for both time and power leakages, studying both average case-, worst case- and indiscernible information metrics.

Index terms — Computer security; Information security; Information leakage; Mutual information; Side-channel attacks; Indiscernibility Methodology; IIR metric

[*] This work was supported by the EU Horizon 2020 project *TeamPlay* (`https://www.teamplay-h2020.eu`), grant number 779882.

[†] yoann.marquer@uni.lu Most of the work was done while Y. Marquer was at Inria[2].

[‡] Olivier.Zendra@inria.fr

[§] annelie.heuser@irisa.fr

Contents

1 Introduction

1.1 Context

Cybersecurity and cyber threats are important and growing issues in Information and Communications Technology (ICT) systems these years. Amongst these threats are the side channel attacks, that exploit information leaking from ICT systems through non-functional properties such as timing, power and energy profiles, etc. Indeed, by observing these physical channels from outside the system, either with measuring apparatus or remotely by getting access to hardware information counters, an attacker is able to gain information on the system execution with little or even no visible impact in the system. This makes these attacks very stealthy and difficult to protect from. Furthermore, these observation methods are not usually envisioned by the systems designers and developer, who generally focus more on the purely functional aspects of the program (what it does) and not the non-functional ones (how it does it).

1.2 Contributions

To help fight this threat, we propose in this paper the Indiscernibility Methodology, a novel methodology to quantify the information leaked from programs. Our goal is to provide ICT systems designers and developers with formally sound, yet practically usable, security metrics based on physical measurements and empirical validation. Our approach is original in the sense that it considers the code to be analyzed as a completely black box: only the input channels and leakages are needed. It requires no prior knowledge, and quantifies directly the indiscernibility of secret-dependent observations. Moreover, our methodology is universal because it does not depend on the type of observations performed by the attacker, i.e. timing, energy, power, electro-magnetic (EM) emanations, etc. We obtain security metrics derived from a topological or information-theoretic analysis, providing insight on useful security aspects of the studied programs.

1.3 Organization of the paper

Section 2 provides background and related work on non-functional properties, side-channels and related security metrics. Section 3 explains our Indiscernibility Methodology, including channels of information and our threat model. The following sections detail the computation of our novel security metrics, with strong formal foundations based in Section 4 on topological security and in Section 5 on information theory. Section 6 shows the applicability of our methodology by providing experimental results for both time and power leakages, studying average-case metrics, worst-case metrics and indiscernible information metrics. Finally, Section 8 concludes.

2 Background

A *side-channel* is a way to transmit information (purposely or not) to another system, out of the standard (intended) communication channels. *Side-channel attacks* rely on the relationship between information leaked through a side-channel and the secret data to obtain confidential (non-public) information. In particular, they can be used to break cryptography by exploiting information that is observed during an algorithm physical execution. To be successful, an attacker must be able 1) to extract information about the secret key through side-channel observations, and 2) to effectively recover the secret from the extracted information.

The first part, the ability to extract information about the secret key through side-channel observations, has been achieved in practice by side-channel attacks based on execution time [Koc96] or power profiles, e.g. SPA (Simple Power Analysis) [KJJ99], CPA (Correlation Power Analysis) [BCO04] or MIA (Mutual Information Analysis) [GBTP08], the latter being detailed in Subsection 7.2 and compared with our methodology. The information flow reflecting the attacker's point of view is displayed in Figure 1. The attacker is assumed to have knowledge of the public input X while observing leakage information L. The secret input K is unknown and is the target of the attacker. To mount attacks like CPA and MIA, an attacker iterates over all possible values[1] Moreover, to be successful the attacker requires information on intermediate values, which in real-world applications can be a strong requirement.

The second part, the ability to effectively recover the secret from the extracted information, can be estimated by using *security metrics*, i.e. quantification of security property or information leakage. Security usually deals with specific attacks and dedicated countermeasures, hence security metrics tend to be based on the difficulty to perform an attack, like the number of measurements for the attacker's remaining uncertainty [KB07], the Signal-to-Noise Ratio (SNR) [MOP07] or the related [APS19] success rate [SMY09]. Instead of attacking, another approach is to detect the amount of exploitable leakage information independent of an attack strategy. Test Vector Leakage Assessment (TVLA) [GJJR11, CD13] is one of the most popular leakage detection methodology with several extensions adapted to particular leakage scenarios, but requires [BSS19] knowledge of parameters like SNR, the density, and the degree of dependency of the processed samples. The information flow of methodologies like TVLA is displayed in Figure 2, where only the public input X and the leakages L are assumed to be known. Due to the limited required knowledge, leakage detection methodologies are prone to false positives, i.e. the detection of "vulnerabilities" that are not depending on secret inputs and can therefore not be exploited by an attacker. Security metrics like side-channel vulnerability factor [DMWS12] or perceived information [SMY09] are detailed in Section 7 and compared with our novel security metrics.

3 Indiscernibility Methodology

There is no universal metric for security like seconds for time or watts for power, because security encompasses many aspects. In this paper we do not hope that a "one size fits all" metric will be able to cover all the relevant aspects of security, so we propose several metrics based on a common approach: the *Indiscernibility Methodology*.

[1] To be computationally feasible, an attacker splits the key into chunks (e.g. 8 bits) related to the measured leakage, and attacks each chunk individually.

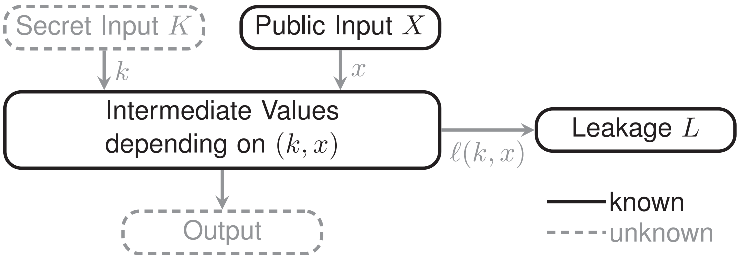

Figure 1: Information Flow for an Attacker

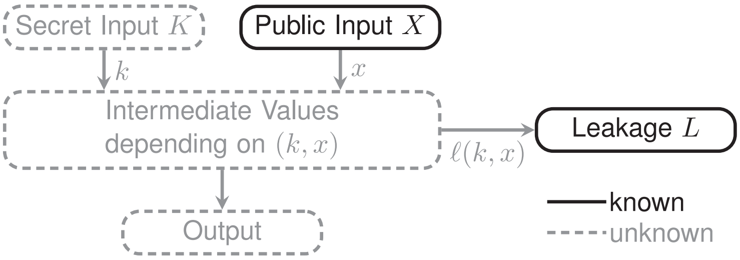

Figure 2: Information Flow for a TVLA

3.1 Scope and purpose

Our novel Indiscernibility Methodology relies on the following principles:

- *Channel-focus*: To remain practical we cannot deal with all the possible aspects of reality, hence we focus on one dedicated leakage channel at a time, e.g. either timing or power side-channels in Section 6.

- *Channel-universality*: We do not exploit any particular property of any side-channel (e.g. Maxwell's equations for electromagnetic emanations), all are treated only as raw numbers and the analysis can be applied indifferently to one channel or another. This implies that with our methodology several channels of interest can be analyzed independently.

- *Defender-oriented*: The methodology is focused on the system to be evaluated and not potential attackers, so the analysis is performed using only the information available to an evaluator of the system, without assumption on the attacker capabilities beside access to the considered leakage channel. This ensures a more intrinsic evaluation of the system instead of focusing on the many and always new attacks or variants.

- *Black-box approach*: Leibniz's *identity of indiscernibles* principle states that from a given point of view if the observed properties of two systems are the same then the systems should be considered the same. Thus, only the information accessible to an attacker, i.e. inputs and leakages, are relevant for the analysis. Therefore we do not consider intermediate values by themselves, considering them only through their impact on what an attacker can observe. This leads to a more straightforward analysis where no knowledge on the internal system is assumed.

- *Secret/public dichotomy*: We assume that the secret inputs of the evaluated system have been identified, and according to *Kerckhoffs's principle* all other inputs are considered public, thus potentially known by the attacker.

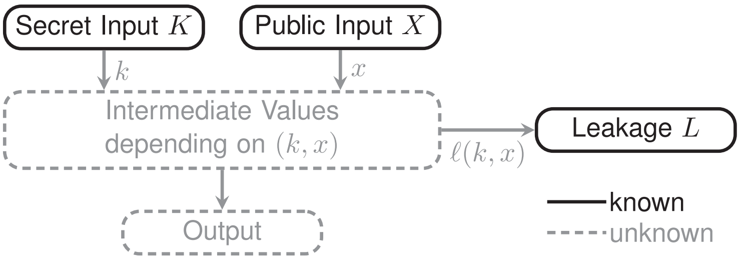

Figure 3: Information Flow for our Indiscernibility Methodology

- *Security level*: We quantify information leakage through the considered leakage channel by a number[2], which can then be used to compare systems or countermeasures, depending on the considered use-case.

3.2 Channels of Information

In the following, channels of information are represented by using discrete random variables. If X denote a given random variable, then $\mathcal{X}$ denotes its domain and p_X its distribution, so $\mathrm{p}_X(x)$ denotes the probability that the channel of information X takes on a value $x \in \mathcal{X}$. $\mathrm{E}_{\mathrm{p}_X}[X] \overset{\text{def}}{=} \sum_{x \in \mathcal{X}} \mathrm{p}_X(x)x$ denotes the *expected value* of the channel X.

The inputs $(k^1, \dots, k^m, x^1, \dots, x^n)$ of the program are split between m secret and n public ones. Let $K^1, \dots, K^m$ be the channels corresponding to the secret variables, and $X^1, \dots, X^n$ be the channels corresponding to the public variables. We consider only finite sampling of the inputs, thus we assume that their domains are finite.

$K \overset{\text{def}}{=} (K^1, \dots, K^m)$ denotes the joint *secret channel*, $\mathcal{K}$ its domain, and a tuple $k = (k^1, \dots, k^m)$ is called a *secret input*. $X \overset{\text{def}}{=} (X^1, \dots, X^n)$ denotes the joint *public channel*, $\mathcal{X}$ its domain, and a tuple $x = (x^1, \dots, x^n)$ is called a *public input*. From an input $(k, x) \in (\mathcal{K}, \mathcal{X})$ the attacker observes a leakage $\ell(k, x)$, which can be any side-channel like:

- the execution time of the studied program,
- the energy consumed by the studied program,
- the power measured by an oscilloscope, in that case the timestamps of the power traces are considered public and included in the public input x so that the following analysis can be done directly on power values instead of traces, as in Subsection 6.2.

For sake of simplicity, we assume in the following that $\ell(k, x)$ is a real number, eventually up to a given resolution. $L \overset{\text{def}}{=} \ell(K, X)$ denotes the *leakage channel*, and if a secret input $k \in \mathcal{K}$ is fixed we denote $L_k \overset{\text{def}}{=} \ell(k, X)$ the corresponding leakage channel.

[2] Our topological metrics in Section 4 are not normalized, thus can have arbitrary high values with appropriate inputs, while the IIR information-theoretic metric presented in Section 5 is normalized by exploiting entropy properties.

3.3 Threat Model

In this paper we assume the role of the evaluator from the defending point of view, thus we consider attacks from potentially multiple and/or various attackers. We use a black box approach: we consider only the *input channel* (K, X) of the program and the leakage channel L observed by a potential attacker; all other aspects of the computation are ignored. The information flow diagram of our threat model is displayed in Figure 3. The attacker wants to know the secret channel K, and we apply Kerckhoffs's principle: the attacker knows the leakage channel L and the public channel X, thus the *attacker channel* (L, X). We do not make more assumptions, especially regarding attacker capabilities: the attacker's computational power might be unlimited, and noise may be negligible.

Taking into account a noise channel N would imply that the leakage channel is $L = \ell(K, X, N)$, the attacker would know (L, X) and will obtain information on (K, N). So the analysis would be similar, except that the attacker has no way of discerning K from N thus this noise N would act like a countermeasure protecting K. Therefore, assuming no noise $L = \ell(K, X)$ is a worst-case scenario for the victim.

4 Novel Topological Security Metrics

Let k_1, k_2 be two secret inputs, and L_{k_1}, L_{k_2} the corresponding leakage channels. According to the identity of indiscernibles principle, if for the attacker leakage channels $L_{k_1} \approx L_{k_2}$ are similar enough then the attacker must confuse $k_1 = k_2$.

Thus, in this section we estimate information leakage by using different distances $d(L_{k_1}, L_{k_2})$ and ways to aggregate the distances obtained from the various pairs k_1, k_2 of secret inputs. We focus in Subsection 4.1 on distances obtained from norms, but we generalize this approach by using matrices in Subsection 4.2. The ACDL (Average-Case Discernible Leakage) security metric corresponds to the average distance, while the the WCDL (Worst-Case Discernible Leakage) corresponds to the worst one. Finally we estimate the cost to compute these novel topological security metrics in Subsection 4.3.

4.1 Norm-based Distances

Because the channel X is public, for a given secret input $k \in \mathcal{K}$, the corresponding leakage channel $L_k = \ell(k, X)$ can be seen as a vector $(\ell(k, x_1), \ldots, \ell(k, x_{\mathrm{card}(X)}))$ in a $\mathrm{card}(\mathcal{X})$-dimensional space.

We consider only possible public inputs[3], i.e. for every $x \in \mathcal{X}$ we have $0 < \mathrm{p}_X(x)$, so the following bilinear form is an inner product:

$$\langle L_{k_1} | L_{k_2} \rangle \overset{\text{def}}{=} \sum_{x \in \mathcal{X}} \mathrm{p}_X(x) \ell(k_1, x) \ell(k_2, x)$$

[3] If it was not the case then the bilinear form would be only semi-definite, leading to a semi-norm $\|.\|_{2\text{-wgt}}$ and a pseudometric $d(.,.)_{2\text{-wgt}}$.

from which we can generate a weighted[4] norm:

$$\|L_k\|_{2\text{-wgt}} \overset{\text{def}}{=} \sqrt{\langle L_k | L_k \rangle} = \sqrt{\sum_{x\in\mathcal{X}} \mathrm{p}_X(x)\ell(k,x)^2}$$

More generally, for every $1 \leq q < \infty$, the following function is a norm:

$$\|L_k\|_{q\text{-wgt}} \overset{\text{def}}{=} \sqrt[q]{\sum_{x\in\mathcal{X}} \mathrm{p}_X(x)\,|\ell(k,x)|^q}$$

which is a weighted variant of the common $\mathcal{L}^q$ norms. If $q \to \infty$ then the latter norms generate the uniform (or Chebyshev) norm:

$$\|L_k\|_\infty \overset{\text{def}}{=} \max_{x\in\mathcal{X}} |\ell(k,x)|$$

In the following we focus on the weighted Euclidean norm for the average-case and the uniform norm for the worst-case discernible leakages. From a norm we can generate a distance:

$$d(L_{k_1}, L_{k_2}) \overset{\text{def}}{=} \|L_{k_1} - L_{k_2}\|$$

Thus we obtain the *weighted Euclidean* and the *Chebyshev distances*:

$$d(L_{k_1}, L_{k_2})_{2\text{-wgt}} = \sqrt{\sum_{x\in\mathcal{X}} \mathrm{p}_X(x)\,|\ell(k_1,x) - \ell(k_2,x)|^2}$$
$$d(L_{k_1}, L_{k_2})_\infty = \max_{x\in\mathcal{X}} |\ell(k_1,x) - \ell(k_2,x)|$$

Every distance satisfies the symmetry $d(L_{k_1}, L_{k_2}) = d(L_{k_2}, L_{k_1})$ and identity of indiscernibles $d(L_{k_1}, L_{k_2}) = 0 \Rightarrow k_1 = k_2$ properties, so only the distances $d(L_{k_i}, L_{k_j})$ for $i < j$ are relevant for the computation, which avoids duplication and dilution.

These distances now have to be aggregated into a security metric. In order to obtain a weight w_{k_i,k_j} proportional to $\mathrm{p}_K(k_i)\mathrm{p}_K(k_j)$ and such that $\sum_{i<j} \mathrm{w}_{k_i,k_j} = 1$, the *Average-Case Discernible Leakage* is computed as:

$$\text{ACDL} \overset{\text{def}}{=} \sum_{(k_i,k_j)\in\mathcal{K}^2 \,|\, i<j} \mathrm{w}_{k_i,k_j} d(L_{k_i}, L_{k_j})_{2\text{-wgt}}$$
$$\text{where } \mathrm{w}_{k_i,k_j} \overset{\text{def}}{=} \frac{\mathrm{p}_K(k_i)\mathrm{p}_K(k_j)}{\sum_{(k_{i'},k_{j'})\in\mathcal{K}^2 \,|\, i'<j'} \mathrm{p}_K(k_{i'})\mathrm{p}_K(k_{j'})}$$

It quantifies how difficult it is for the attacker to find a relevant point of interest. The higher is the ACDL, the less secure is the system. Note that if the distribution of public inputs is uniform, since the number of considered distances is $\frac{1}{2}\text{card}(\mathcal{K})\,(\text{card}(\mathcal{K}) - 1)$, which is huge, then the median[5] can be used

[4] In contrast to the common Euclidean norm $\|L_k\|_2 = \sum_{x\in\mathcal{X}} \ell(k,x)^2$.

[5] The median is faster to compute and statistically more robust than the average. But, because it is the limit of the non-weighted average, it does not take into account the distribution of public inputs.

as a shortcut to compute the ACDL. In that case, its interpretation is more to provide a representative case.

The *Worst-Case Discernible Leakage* is computed as:

$$\text{WCDL} \overset{\text{def}}{=} \max_{(k_i,k_j)\in\mathcal{K}^2 \,|\, i<j} d(L_{k_i}, L_{k_j})_\infty$$

It quantifies the worst possible discernibility, when the corresponding point of interest has been found by the attacker. The higher is the WCDL, the less secure is the system.

Note that for both ACDL and WCDL security metrics, the unit of measurement is the same as the leakage unit, e.g. clock cycles for time as in Subsection 6.1 and watts[6] for power as in Subsection 6.2.

4.2 Matrix-based Distances

The previous weighted inner product:

$$\langle L_{k_1} | L_{k_2} \rangle = \sum_{x\in\mathcal{X}} \mathrm{p}_X(x)\ell(k_1,x)\ell(k_2,x)$$

can be seen as a matrix multiplication:

$$\begin{aligned}\langle L_{k_1} | L_{k_2} \rangle_M &\overset{\text{def}}{=} {L_{k_1}}^{\mathrm{T}} M L_{k_2} \\ &= \sum_{x_1\in\mathcal{X}} \ell(k_1,x_1) \sum_{x_2\in\mathcal{X}} M(x_1,x_2)\ell(k_2,x_2)\end{aligned}$$

where the matrix $M = P$ is the diagonal probability matrix $P(x_1,x_2) = \mathrm{p}_X(x_1)$ if $x_1 = x_2$ and 0 otherwise. Similarly, the more common inner product $\langle L_{k_1} | L_{k_2} \rangle = \sum_{x\in\mathcal{X}} \ell(k_1,x)\ell(k_2,x)$ corresponds to $M = I$ the identity matrix. As before, an inner product generates a norm then a distance:

$$\begin{aligned}d(L_{k_1}, L_{k_2})_M &\overset{\text{def}}{=} \sqrt{\langle L_{k_1} - L_{k_2} | L_{k_1} - L_{k_2} \rangle_M} \\ &= \sqrt{\begin{aligned}&\sum_{x_1\in\mathcal{X}} (\ell(k_1,x_1) - \ell(k_2,x_1)) \times \\ &\quad \sum_{x_2\in\mathcal{X}} M(x_1,x_2)(\ell(k_1,x_2) - \ell(k_2,x_2))\end{aligned}}\end{aligned}$$

In particular, we have $d(.,.)_I = d(.,.)_2$ the common Euclidean distance, and $d(.,.)_P = d(.,.)_{\text{2-wgt}}$ the weighted Euclidean distance used to compute the ACDL.

More generally, if a matrix M is positive-definite, i.e. for every $L_k \neq \vec{0}$ we have ${L_k}^{\mathrm{T}} M L_k > 0$, then $\langle . | . \rangle_M$ is an inner product and thus $d(.,.)_M$ is a distance. If M is symmetric, i.e. $M(x_1,x_2) = M(x_2,x_1)$, then the Sylvester's criterion can be used to determine whether M is positive-definite, thus we can compute whether a given matrix generates a distance.

In this subsection, if a public input $x \in \mathcal{X}$ is fixed we denote $L_x \overset{\text{def}}{=} \ell(K,x)$ the corresponding leakage channel. Let C be the covariance matrix $C(x_1,x_2) =$

[6] Measured indirectly as voltage for a constant intensity.

$\mathrm{E}_K[(L_{x_1} - \mathrm{E}_K[L_{x_1}])(L_{x_2} - \mathrm{E}_K[L_{x_2}])]$. Covariance matrix C is always positive semi-definite, i.e. $L_k{}^{\mathrm{T}} M L_k \geq 0$. A matrix is positive-definite if and only if it is invertible and positive semi-definite, so if C is invertible then it is positive-definite. Moreover, the inverse of a positive-definite matrix is also positive-definite. So, if covariance matrix C is invertible then its inverse C^{-1} (called concentration or precision matrix) is positive-definite, therefore $d(.,.)_{C^{-1}}$ is a distance called the *Mahalanobis distance* [Mah36].

Note that if card(X) is large enough then it is likely that covariance matrix C would be invertible[7]. Thus, we can use the Mahalanobis distance instead of the weighted Euclidean distance to compute the ACDL. Indeed, if each of the axes of the public input space is rescaled to have unit variance then the Mahalanobis distance corresponds to the standard Euclidean distance in the transformed space. For instance, if the leakages L_x are independent then the Mahalanobis distance is simply:

$$d(L_{k_1}, L_{k_2})_M = \sqrt{\sum_{x \in \mathcal{X}} \frac{|\ell(k_1, x) - \ell(k_2, x)|^2}{\sigma_x^2}}$$

$$\text{where } \sigma_x^2 = \mathrm{E}_K\left[(L_x - \mathrm{E}_K[L_x])^2\right] \text{ is the variance}$$

In other words, the Mahalanobis distance corresponds to a space distorted by the deviations due to the public inputs, thus the computed distances express only the deviations due to the secret inputs. It has been used to implement an efficient template attack [CK14], and it is closely related to Hotelling's T^2 distribution [Hot31], which is used for the multivariate variant [BSS19] of the Welch's t-test [Wel47], commonly used in the Test Vector Leakage Assessment (TVLA) methodology [GJJR11, CD13].

In the experimental results in Section 6 we use the weighted Euclidean distance for sake of simplicity and because it is less computationally heavy than the Mahalanobis distance, especially without the independent signal assumption [BSS19].

4.3 Complexity

Let $n_K = \mathrm{card}(\mathcal{K})$ and $n_X = \mathrm{card}(\mathcal{X})$. In this subsection we assume at worst a total of card$(\mathcal{K}, \mathcal{X}) = n_K \times n_X$ observations. Computing a distance costs $\mathcal{O}(n_X)$ operations if it is norm-based, but $\mathcal{O}(n_X^2)$ if it is matrix-based. Then there are $\mathcal{O}(n_K^2)$ distances to be computed and $\mathcal{O}(n_K^2)$ other operations to compute the metrics. Therefore the ACDL and the WCDL costs $\mathcal{O}(n_K^2 \times n_X)$ for norm-based distances and $\mathcal{O}(n_K^2 \times n_X^2)$ for matrix-based distances, i.e. are quadratic in the number of observations.

5 Novel Information-theoretic Security Metric

In this section we quantify the information obtained on secret channel K from attacker channels L and X, by using the framework of information theory in Subsection 5.1. Then we apply this framework in Subsection 5.2 to estimate

[7] Based on the Sylvester's criterion, it can be checked by recursively accepting a public input x_i only if the determinant of the new covariance matrix is non-zero.

information leakage by determining the reduction of the search space for the attacker after observation. This approach requires a way to cluster attacker's observations, detailed in Subsection 5.3. Finally we estimate the cost to compute this novel information-theoretic security metric in Subsection 5.4.

5.1 Information Theory

The quantity of information obtained from the observation of a value $x \in \mathcal{X}$ is $\mathrm{info}(x) = \log_2 \frac{1}{\mathrm{p}_X(x)}$ bits. The *entropy* of a channel X is defined as the expected value of the information:

$$\mathrm{H}(X) \overset{\mathrm{def}}{=} \mathrm{E}_X[\mathrm{info}(X)] = \sum_{x \in \mathcal{X}} \mathrm{p}_X(x) \log_2 \frac{1}{\mathrm{p}_X(x)}$$

This formula can also be used for the joint distribution (X, Y) of two channels X and Y to define the *joint entropy*:

$$\mathrm{H}(X, Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \mathrm{p}_{X,Y}(x, y) \log_2 \frac{1}{\mathrm{p}_{X,Y}(x, y)}$$

Similarly, the *conditional entropy* of X knowing Y is defined by:

$$\mathrm{H}(Y \mid X) \overset{\mathrm{def}}{=} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \mathrm{p}_{P_{X,Y}}(x, y) \log_2 \frac{\mathrm{p}_{P_X}(x)}{\mathrm{p}_{P_{X,Y}}(x, y)}$$

and satisfies $\mathrm{H}(X, Y) = \mathrm{H}(X) + \mathrm{H}(Y \mid X)$.

Note that X and Y are statistically independent if and only if for every $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ we have $\mathrm{p}_{X,Y}(x, y) = \mathrm{p}_X(x)\mathrm{p}_Y(y)$. The *mutual information* is the Kullback–Leibler divergence between the joint distribution X, Y and the product of marginal distributions X and Y:

$$\mathrm{MI}(X; Y) \overset{\mathrm{def}}{=} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \mathrm{p}_{X,Y}(x, y) \log_2 \frac{\mathrm{p}_{X,Y}(x, y)}{\mathrm{p}_X(x)\mathrm{p}_Y(y)}$$

It quantifies the amount of information (in bits) obtained about X through observing Y. The mutual information is symmetric $\mathrm{MI}(X; Y) = \mathrm{MI}(Y; X)$ and satisfies $\mathrm{H}(Y) = \mathrm{H}(Y \mid X) + \mathrm{MI}(X; Y)$. Moreover, we have $0 \leq \mathrm{MI}(X; Y) \leq \max(\mathrm{H}(X), \mathrm{H}(Y))$, such that $\mathrm{MI}(X; Y) = 0$ if and only if X and Y are statistically independent, and $\mathrm{MI}(X; Y) = \mathrm{H}(Y)$ if and only if there exists a function f such that $Y = f(X)$.

To facilitate understanding, entropies and mutual information are usually represented as an information diagram in Figure 4.

5.2 Application to Security

The attacker has access to the leakage and public channels L, X and wants to know the secret channel K. $\mathrm{H}(K)$ can be seen as the quantity of information necessary to describe the exploration space for the secret information. Because $\mathrm{H}(K) = \mathrm{H}(K \mid L, X) + \mathrm{MI}(K; L, X)$, the relative proportion of information about the secret is split from the attacker's point of view as:

$$\frac{\mathrm{H}(K \mid L, X)}{\mathrm{H}(K)} + \frac{\mathrm{MI}(K; L, X)}{\mathrm{H}(K)} = 1$$

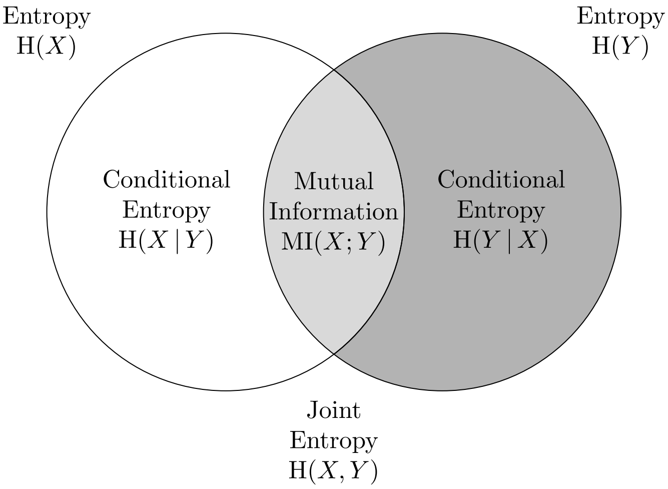

Figure 4: Information diagram

We define the quantity on the left[8] as the *Indiscernible Information Ratio* (IIR), i.e. the proportion of what remains of the exploration space for the secret after observation by the attacker:

$$\mathrm{IIR}(K; L, X) \overset{\text{def}}{=} \frac{\mathrm{H}(K \mid L, X)}{\mathrm{H}(K)}$$

Because $0 \leq \mathrm{H}(K \mid L, X) \leq \mathrm{H}(K)$ the IIR is normalized: $0 \leq \mathrm{IIR}(K; L, X) \leq 1$. It is a security level, the higher the better, such that $\mathrm{IIR}(K; L, X) = 1$ if and only if the secret channel K is statistically independent from the attacker channels L, X, and $\mathrm{IIR}(K; L, X) = 0$ if and only if there exists any function f such that $K = f(L, X)$.

This security level is compatible with the highest attacker's capabilities. Indeed, it assumes that the attacker is able to perfectly exploit the information available from accessible channels, without bound on computational power nor limitation due to noise in measurements.

We assumed in Subsection 3.3 that leakage channel $L = \ell(K, X)$ is entirely determined by the input channels. Because $L = \ell(K, X)$ we have $\mathrm{H}(L \mid K, X) = 0$, so $\mathrm{H}(L, K, X) = \mathrm{H}(K, X)$, thus $\mathrm{H}(K \mid L, X) = \mathrm{H}(K, X) - \mathrm{H}(L, X)$.

We demonstrate now that the values of secret and public inputs do not matter: only the probabilities $\mathrm{p}_{K,X}(k, x)$ of inputs and the leakage values $\ell(k, x)$ are relevant to compute these entropies. Because $L = \ell(K, X)$ the joint distribution of the system is:

$$\mathrm{p}_{L,K,X}(\ell, k, x) = \begin{cases} \mathrm{p}_{K,X}(k, x) & \text{if } \ell = \ell(k, x) \\ 0 & \text{otherwise} \end{cases}$$

Hence the attacker marginal distribution:

$$\mathrm{p}_{L,X}(\ell, x) = \sum_{k \in \mathcal{K}} \mathrm{p}_{L,K,X}(\ell, k, x) = \sum_{k \in \mathcal{K} \,\mid\, \ell(k,x) = \ell} \mathrm{p}_{K,X}(k, x)$$

[8] Which corresponds to a sharper variant [KSAG03] $\frac{\mathrm{H}(X,Y) - \mathrm{MI}(X;Y)}{\max(\mathrm{H}(X),\mathrm{H}(Y))}$ of the Jaccard distance for information, but focused on the secret channel.

Note that if $\ell \notin \ell(\mathcal{K}, x)$ then $\mathrm{p}_{L,X}(\ell, x) = 0$.

For a given $x \in \mathcal{X}$, we denote $k_1 \approx_x k_2$ two secret inputs $k_1, k_2 \in \mathcal{K}$ that are indiscernible from the leakage observation $\ell(k_1, x) \approx \ell(k_2, x)$, where the $\approx$ relation will be more detailed in the next subsection on clustering. $\approx_x$ is assumed to be an equivalence relation. We denote $C_x(k) \overset{\text{def}}{=} \{k' \in \mathcal{K} \mid k' \approx_x k\}$ the equivalence class of $k \in \mathcal{K}$ for the $\approx_x$ relation, and $\mathcal{K}/\approx_x$ the set of representatives for these classes. Thus, we obtain the attacker entropy as a function of input distribution and leakage observations:

$$\begin{aligned}
&\mathrm{H}(L, X) \\
&= \sum_{x \in \mathcal{X}} \sum_{\ell \in \ell(\mathcal{K}, x)} \mathrm{p}_{L,X}(\ell, x) \log_2 \frac{1}{\mathrm{p}_{L,X}(\ell, x)} \\
&= \sum_{x \in \mathcal{X}} \sum_{\ell \in \ell(\mathcal{K}, x)} \sum_{\substack{k_2 \in \mathcal{K} \,\mid \\ \ell(k_2, x) = \ell}} \mathrm{p}_{K,X}(k_2, x) \log_2 \frac{1}{\sum\limits_{\substack{k' \in \mathcal{K} \,\mid \\ \ell(k', x) = \ell}} \mathrm{p}_{K,X}(k', x)} \\
&= \sum_{x \in \mathcal{X}} \sum_{k_1 \in \mathcal{K}/\approx_x} \sum_{\substack{k_2 \in \mathcal{K} \,\mid \\ k_2 \approx_x k_1}} \mathrm{p}_{K,X}(k_2, x) \log_2 \frac{1}{\sum\limits_{\substack{k' \in \mathcal{K} \,\mid \\ k' \approx_x k_1}} \mathrm{p}_{K,X}(k', x)} \\
&= \sum_{x \in \mathcal{X}} \sum_{k \in \mathcal{K}} \mathrm{p}_{K,X}(k, x) \log_2 \frac{1}{\sum\limits_{k' \in C_x(k)} \mathrm{p}_{K,X}(k', x)}
\end{aligned}$$

The input entropy is:

$$\mathrm{H}(K, X) = \sum_{x \in \mathcal{X}} \sum_{k \in \mathcal{K}} \mathrm{p}_{K,X}(k, x) \log_2 \frac{1}{\mathrm{p}_{K,X}(k, x)}$$

Thus the indiscernable information is:

$$\begin{aligned}
\mathrm{H}(K \mid L, X) &= \mathrm{H}(K, X) - \mathrm{H}(L, X) \\
&= \sum_{x \in \mathcal{X}} \sum_{k \in \mathcal{K}} \mathrm{p}_{K,X}(k, x) \log_2 \frac{\sum\limits_{k' \in C_x(k)} \mathrm{p}_{K,X}(k', x)}{\mathrm{p}_{K,X}(k, x)}
\end{aligned}$$

Finally, the secret entropy is obtained from the secret marginal distribution:

$$\begin{aligned}
\mathrm{p}_K(k) &= \sum_{x \in \mathcal{X}} \mathrm{p}_{K,X}(k, x) \\
\mathrm{H}(K) &= \sum_{k \in \mathcal{K}} \mathrm{p}_K(k) \log_2 \frac{1}{\mathrm{p}_K(k)}
\end{aligned}$$

To facilitate understanding, these quantities are represented as a discernibility diagram in Figure 5. The Indiscernible Information Ratio is computed as:

$$\begin{aligned}
\mathrm{IIR}(K; L, X) &= \frac{\mathrm{H}(K \mid L, X)}{\mathrm{H}(K)} \\
&= \frac{\sum\limits_{x \in \mathcal{X}} \sum\limits_{k \in \mathcal{K}} \mathrm{p}_{K,X}(k, x) \log_2 \frac{\sum_{k' \in C_x(k)} \mathrm{p}_{K,X}(k', x)}{\mathrm{p}_{K,X}(k, x)}}{\sum\limits_{k \in \mathcal{K}} \mathrm{p}_K(k) \log_2 \frac{1}{\mathrm{p}_K(k)}}
\end{aligned}$$

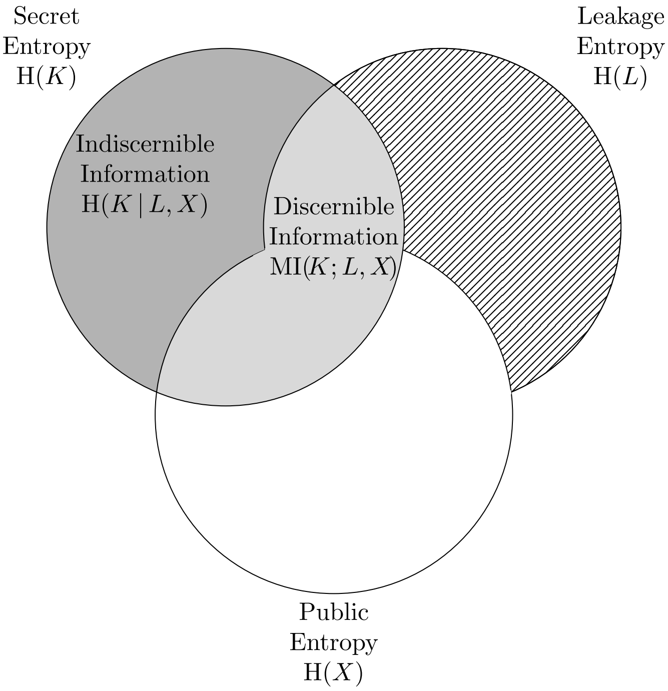

Figure 5: Discernibility diagram

Let $x \in \mathcal{X}$ and $k \in \mathcal{K}$ be some input. Because $\approx_x$ is an equivalence relation, we have $k \in C_x(k)$, so $\mathrm{card}(C_x(k)) \geq 1$.

If $C_x(k)$ is a singleton, i.e. $C_x(k) = \{k\}$, then $\sum_{k' \in C_x(k)} \mathrm{p}_{K,X}(k', x) = \mathrm{p}_{K,X}(k, x)$, so the logarithm is zero, and the input does not contribute to the indiscernable information. Otherwise, the logarithm is positive, and because we considered only possible inputs, the input contributes to the indiscernable information. In other words, only the secret inputs producing similar observations contribute to the indiscernable information.

Conversely, if $C_x(k) = \mathcal{K}$ is the whole domain of the secret inputs, then $\sum_{k' \in C_x(k)} \mathrm{p}_{K,X}(k', x) = \mathrm{p}_X(x)$ is the marginal probability, so:

$$\mathrm{H}(L, X) = \sum_{x \in \mathcal{X}} \sum_{k \in \mathcal{K}} \mathrm{p}_{K,X}(k, x) \log_2 \frac{1}{\mathrm{p}_X(x)} = \mathrm{H}(X)$$

Because $\mathrm{H}(L, X) = \mathrm{H}(X) + \mathrm{H}(L \mid X)$, in that case $\mathrm{H}(L \mid X) = 0$. The indiscernable information is $\mathrm{H}(K \mid L, X) = \mathrm{H}(K, X) - \mathrm{H}(L, X) = \mathrm{H}(K, X) - \mathrm{H}(X) - \mathrm{H}(L \mid X)$ where the three entropies are positive or zero. So, in that case the only composant depending on L is $\mathrm{H}(L \mid X) = 0$. Therefore, if $C_x(k) = \mathcal{K}$, then the indiscernable information is maximal.

In other words, there are two extrema: 1) if all equivalence classes are singletons then the indiscernable information is zero, and 2) if there is only one equivalence class per public input then the indiscernable information is maximal.

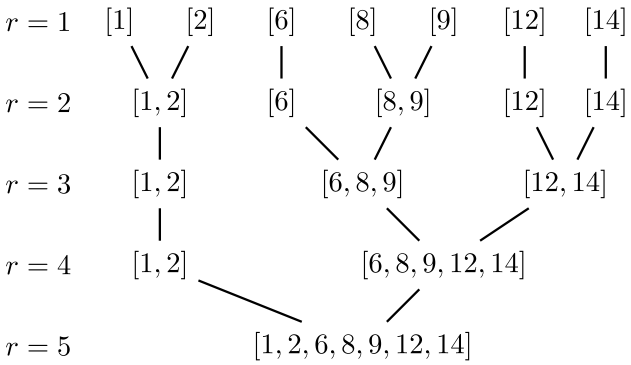

Figure 6: Dendrogram clusterings for various resolutions

5.3 Leakage clustering

Let $x \in \mathcal{X}$ be a given public input. Remains the question of defining the classes of observations $\ell(k_1, x) \approx \ell(k_2, x)$ such that $\approx_x$ is an equivalence relation. We do that by partitioning $\ell(\mathcal{K}, x)$ in clusters.

We use a *dendrogram clustering*, i.e. for a given resolution r the leakages ℓ_1 and ℓ_2 will be in the same cluster if and only if $|\ell_1 - \ell_2| < r$, obtaining a tree of the possible clusterings for various r in Figure 6.

For a given public input x, let $[\ell_0, \ldots, \ell_{n_L - 1}]$ be the list of the n_L leakage values observed by the attacker, sorted by increasing order. We then compute the list Diff_x of the $n_L - 1$ differences between two consecutive leakages, i.e. for every $0 \leq i \leq n_L - 2$ we have $\mathrm{Diff}_x[i] = \ell_{i+1} - \ell_i$.

Let I be the set of considered indices, starting at $[0, \ldots, n_L - 2]$. At first leakages are all considered to be discernible so we initialize the clustering with singletons $\mathrm{Clus}_x = [[\ell_0], \ldots, [\ell_{n_L - 1}]]$. For every step, we get the indices of the minimal difference:

$$d_{\min} = \min_{i \in I} \mathrm{Diff}_x[i]$$

$$I_{\min} = \{i \in I \mid \mathrm{Diff}_x[i] = d_{\min}\}$$

Then, if $d_{\min} = 0$ or the termination condition is not met, then a merging step is done. During a merging step, for every $i \in I_{\min}$ the clusters $\mathrm{Clus}_x[i]$ and $\mathrm{Clus}_x[i+1]$ are merged and the occurrences of $d_{\min}$ in Diff_x are removed, leaving $I \setminus I_{\min}$ as remaining indices for the next step. The condition $d_{\min} = 0$ ensures that identical leakage values are in the same cluster[9], in other words that the identity is always a refinement of the $\approx_x$ relation. The merging steps are repeated until the termination condition is met, i.e. $d_{\min} \geq r$ the expected resolution, or if the number of clusters after the next merging step would be strictly below the expected number c of clusters.

The resolution r and number c of clusters can be provided by the developper. If none is provided, then we assume a default value for the number of clusters:

$$c_{\text{default}} = \lfloor \log_2 n_K \rfloor + 1$$

[9] If there are identical leakages then this is always the first step of our dendrogram algorithm, thus it can be seen as an alternative initialization for the clusters, instead of singletons.

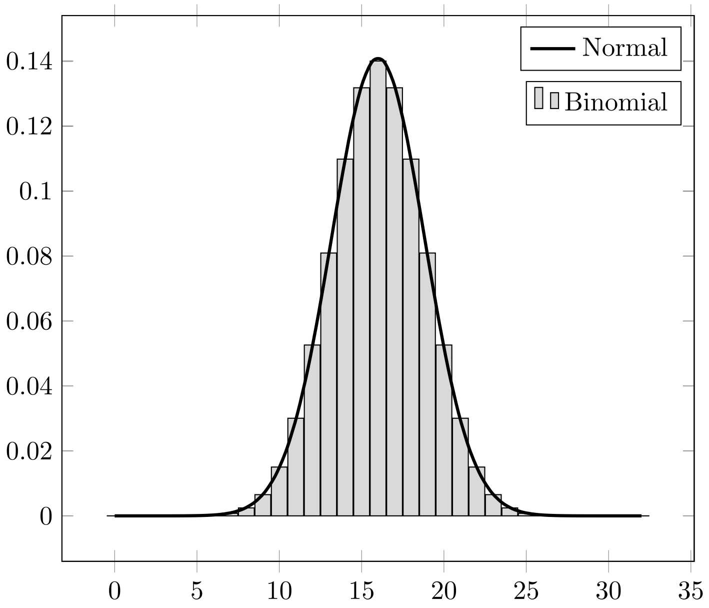

Figure 7: Normal and binomial distributions (32 bits)

that corresponds to the number of Hamming weights used in the exhaustive case in Section 6. More generally, if we assume by default a normal distribution for observations $\ell(\mathcal{K}, x)$, this distribution can be discretized as in Figure 7 for n_{bits} bits by a binomial distribution with $c = n_{\text{bits}} + 1$ clusters for a total of $n_K = 2^{n_{\text{bits}}}$ observations, hence the default value.

5.4 Complexity

We remind that $n_K = \text{card}(\mathcal{K})$, $n_X = \text{card}(\mathcal{X})$ and we assume at worst a total of $\text{card}(\mathcal{K}, \mathcal{X}) = n_K \times n_X$ observations. For a given $x \in \mathcal{X}$, sorting the data then the clusters costs $\mathcal{O}(n_K \times \log n_K)$ operations. So the cost of clustering is $\mathcal{O}(n_K \times \log n_K \times n_X)$. Then the cost of the computation of $\text{IIR}(K; L, X)$ is linear in number of observations $\mathcal{O}(n_K \times n_X)$. Therefore, the indiscernible information ratio costs $\mathcal{O}(n_K \times \log n_K \times n_X)$ operations, thus is almost linear in the number of observations.

6 Experimental results

In this section we validate the relevance of our security metrics by comparing their values for a security use-case: the modular exponentiation, commonly used in crypto-systems like RSA [RSA78]. We investigate timing leakages in Subsection 6.1 and power leakages in Subsection 6.2. For sake of simplicity the experiments are done by assuming a uniform distribution for secret and public inputs, non-uniform distributions being future work.

Let k be a secret variable, a and n be two public variables. The *square-and-multiply* (sqmul) program described in Listing 1 computes the (left-to-right) modular exponentiation $a^k \bmod n$. The `loopbound` pragma is used to indicate

Listing 1: Square-and-multiply

```
unsigned int x = 1;

_Pragma("loopbound min 32 max 32");
for (i = 31 ; i >= 0 ; i--)// bits of k from left to right
{
    x = mod_barrett(x*x, n, shift, mu);

    if ((k & (1 << i)) != 0)// current bit is 1
    {
        x = mod_barrett(a*x, n, shift, mu);
    }
}
// x = a^k % n
```

Listing 2: Montgomery ladder

```
unsigned int x = 1;
unsigned int y = a;

_Pragma("loopbound min 32 max 32");
for (i = 31 ; i >= 0 ; i--)// bits of k from left to right
{
    if ((k & (1 << i)) != 0)// current bit is 1
    {
        x = mod_barrett(x*y, n, shift, mu);
        y = mod_barrett(y*y, n, shift, mu);
    }
    if ((k & (1 << i)) == 0)// current bit is 0
    {
        y = mod_barrett(y*x, n, shift, mu);
        x = mod_barrett(x*x, n, shift, mu);
    }
}
// x = a^k % n
```

the analysis tools that the iteration is done 32 times. For every iteration i, `k & (1 << i)` computes bit i of the secret key k. The squaring `x*x` is always computed, but the multiplication `a*x` is computed only if the current bit is 1. This can be detected by observing execution time [Koc96] or power profiles by means of e.g. SPA (Simple Power Analysis) [KJJ99], leading to information leakage from both time and power side-channel attacks. The *Montgomery ladder* [LM87] described in Listing 2 computes also the (left-to-right) modular exponentiation $a^k \bmod n$, but is more regular, thus less leaky.

Our timing and power measurements target the ARM Cortex-M0 processor. Unfortunately, the modulus is not a native operation, thus is computed by default by using a function call to a library containing a costly and leaky implementation. We use in Listing 3 a more secure and efficient modulus operation implemented as a Barrett's reduction [MvOV96], where *shift* is precomputed as the smallest integer s such that $n < 2^s$, and *mu* as $\frac{2^{2\times\text{shift}}}{n}$ rounded down.

Finally, we observed that a standard `if then else` structure generates a conditional jump performed only if the condition is false, thus costing a bit more time in the `else` branch than in the `then` branch, leading to an unexpected timing

Listing 3: Barrett's reduction

```
int mod_barrett(unsigned int v, unsigned int n, unsigned int shift, unsigned int
    mu) {
   unsigned int dummy = v, r;
   r = v - (((v >> (shift - 1)) * mu) >> (shift + 1))*n;
   // require at most 2 more subtractions
   if (r < n)
      dummy = dummy - n;// dummy operation
   if (r >= n)
      r = r - n;
   if (r < n)
      dummy = dummy - n;// dummy operation
   if (r >= n)
      r = r - n;
   return r;// = v % n
}
```

imbalance in the Montgomery ladder variant. To fix that, we used a `if then; if not then` structure instead.

6.1 Timing experiments

According to Kerckhoffs's principle, the attacker is aware of the studied programs, hence can deduce the expected complexity. In both programs the number of modular squarings is constant, but in Listing 1 the number of modular multiplications depends on the Hamming weight[10] $\mathrm{HW}(k)$ of the secret key k, while in Listing 2 it is constant. Thus, execution time is expected to be linear in $\mathrm{HW}(k)$ for Listing 1 but constant for Listing 2.

Knowing that, an attacker might try to profile the chip by studying only one key per Hamming weight. 0 is the only key k such that $\mathrm{HW}(k) = 0$. For $\mathrm{HW}(k) = 1$ we chose the key $k = 1$ because it is odd. We choose all other keys k so they have a 1 on the leftmost bit to ensure a sufficient size, and a 1 on the rightmost bit to ensure the key is odd, the remaining 1s being selected randomly.

In the crypto-system RSA [RSA78], the modulus is a product $n = pq$ of distinct prime numbers, that we generated such that every key is prime with Euler's totient number $\phi(n) = (p-1)(q-1)$. For a given n we precomputed the corresponding *shift* and *mu* values for the Barrett's reduction [MvOV96]. Finally, we randomly generated values for a such that $1 < a < n$.

Our cycle-accurate timing observations (in clock cycles), obtained from the ARM SystemC Cycle Model[11] for ARM Cortex-M0, correspond to the expected complexity:

$$\mathrm{time}_{\mathrm{sqmul}}(k) = 35 \times \mathrm{HW}(k) + 1357$$
$$\mathrm{time}_{\mathrm{ladder}} = 2899$$

We experimentally confirmed that execution time does not depend on the public values (a and n), and that different keys with the same Hamming weight produce

[10] The Hamming weight is the number of non-zero symbols in the representation. Because we use binary, this is the number of 1s.

[11] https://developer.arm.com/tools-and-software/simulation-models/cycle-models/arm-systemc-cycle-models

		ACDL	WCDL	IIR
Selective (32 bits)	sqmul	396.67	1120	0
	ladder	0	0	1
Exhaustive (8 bits)	sqmul	55.20	280	0.682
	ladder	0	0	1

Table 1: Security Metrics from Timing Measurements

the same execution time. The time analysis thus depends only on the Hamming weight, not the actual value of the key.

We used 33 keys (secret tuples) with Hamming weights from 0 to 32, 4 values for a and 4 values for n (hence 16 public tuples) with uniform distribution to compute the ACDL, WCDL and IIR security metrics in row "Selective (32 bits)" of Table 1.

Because the execution time for the ladder variant is constant, both ACDL and WCDL are zero, indicating a perfect protection for this side-channel. The WCDL for the sqmul variant corresponds to the difference of the execution times between a Hamming weight of 0 and a Hamming weight of 32, i.e. $35 \times 32 = 1120$ clock cycles. The IIR was computed with the default number of clusters for the sqmul variant, obtaining a resolution $r = 35$ clock cycles for a default attacker able to discriminate between all the Hamming weights.

For a fair comparison, we used this resolution to compute the IIR for the ladder variant, even if in that case the constant-time determines a IIR $= 1$ for any value of the resolution. IIR $= 0$ for the sqmul variant indicate that from the timing observations the attacker is able to determine the corresponding key amongst the tested ones, while IIR $= 1$ for the ladder variant indicates that the attacker will confuse all the tested keys. Note that because there has been only one key per Hamming weight, IIR $= 0$ indicates that all the information on Hamming weights has leaked, but not information on a particular key with a given Hamming weight.

To get more variability in the results, we demonstrate a case where the attacker obtain some but not all information from the timing leakage. We assume now a brute-force attacker testing all the possible keys for a smaller number of bits, so several keys share the same Hamming weights. More precisely, for b bits the number of keys with Hamming weight h is:

$$\binom{b}{h} = \frac{b!}{h!(b-h)!}$$

We used all the $2^8 = 256$ possible 8-bits keys (secret tuples) and the same 16 public tuples with uniform distribution to compute the ACDL, WCDL and IIR security metrics in the row "Exhaustive (8 bits)" of Table 1.

The ACDL for the sqmul variant is lower than in the selective case, because most of the keys share similar Hamming weights, thus execution times, following a shape similar to the binomial distribution in Figure 7. The WCDL for the sqmul variant corresponds to the difference of the execution times between a Hamming weight of 0 and a Hamming weight of 8, i.e. $35 \times 8 = 280$ clock cycles. It is lower than in the selective case because the deviation between Hamming weights is lower. IIR $= 0.68$ indicates that from timing observations

		ACDL	WCDL	IIR
Selective (32 bits)	sqmul	20.33	181	0.932
	ladder	19.53	183	0.938
Selective (32 bits, low sampl.)	sqmul	20.66	170	0.929
	ladder	19.54	177	0.937
Selective (32 bits, smoothed)	sqmul	16.85	84	0.946
	ladder	11.10	89	0.969
Selective (32 bits, filtered)	sqmul	23.35	181	0.909
	ladder	19.98	183	0.935

Table 2: Security Metrics from Power Measurements

the attacker is able to determine the Hamming weight of the secret key, but cannot discern between two tested keys with the same Hamming weight. The results for the ladder variant are (unsurprisingly) similar to the selective case.

6.2 Power experiments

We now perform experiment "Selective (32 bits)" but with power[12] traces instead of execution times. The target is a STM32F071RBT6 (ARM Cortex-M0, 8Mhz) mounted on the NewAE CW308 UFO board, measured with a Keysight DSOS404A oscilloscope at 250MHz sampling frequency.

For sqmul in Listing 1, the execution time of one iteration depends on whether the current bit is a 0 or a 1. Beginnings of iterations may thus not be synchronized for different keys, which means our security metrics would evaluate more traces misalignment than information leakage. To remedy that, we used triggers to capture the traces per iteration, and padding to synchronize the beginning of each iteration, obtaining traces as in Figure 8 for $k = 2520782877$, $a = 75$, and $n = 745$. An attacker may perform this trace alignment preparation step in practice, to more easily compare secret-dependent operations.

The x-axis corresponds to timestamps, while the y-axis corresponds to (integer) position in the screen of the oscilloscope, linearly correlated with differential voltage from a baseline. The vertical lines touching the bottom of the frame correspond to the power consumption of the trigger for each of the 32 iterations. Note that the binary representation 10010110010000000001100000011101 of secret key k can be read directly by looking at the padding between iterations of sqmul (short for 1, long for 0).

We used the same 33 keys, 4 values for a and 4 values for n as in Subsection 6.1 for the "Selective (32 bits)" experiment. As indicated in Subsection 3.2, the timestamps of power traces are included in public input. We obtained 29440 timestamps for sqmul and 32000 timestamps for ladder. Again, we used uniform distribution to compute the ACDL, WCDL and IIR security metrics in row "Selective (32 bits)" of Table 2.

Note that about half of the time correspond to padding (constant value = 0), moreover except for keys 0 and 1 the first and the last bits are always 1 so only 30 over 32 bits are relevant, and finally depending on whether the current bit

[12] We measured a resistance voltage, so technically our results are in volts not in watts, but because the intensity is fixed both units are proportional.

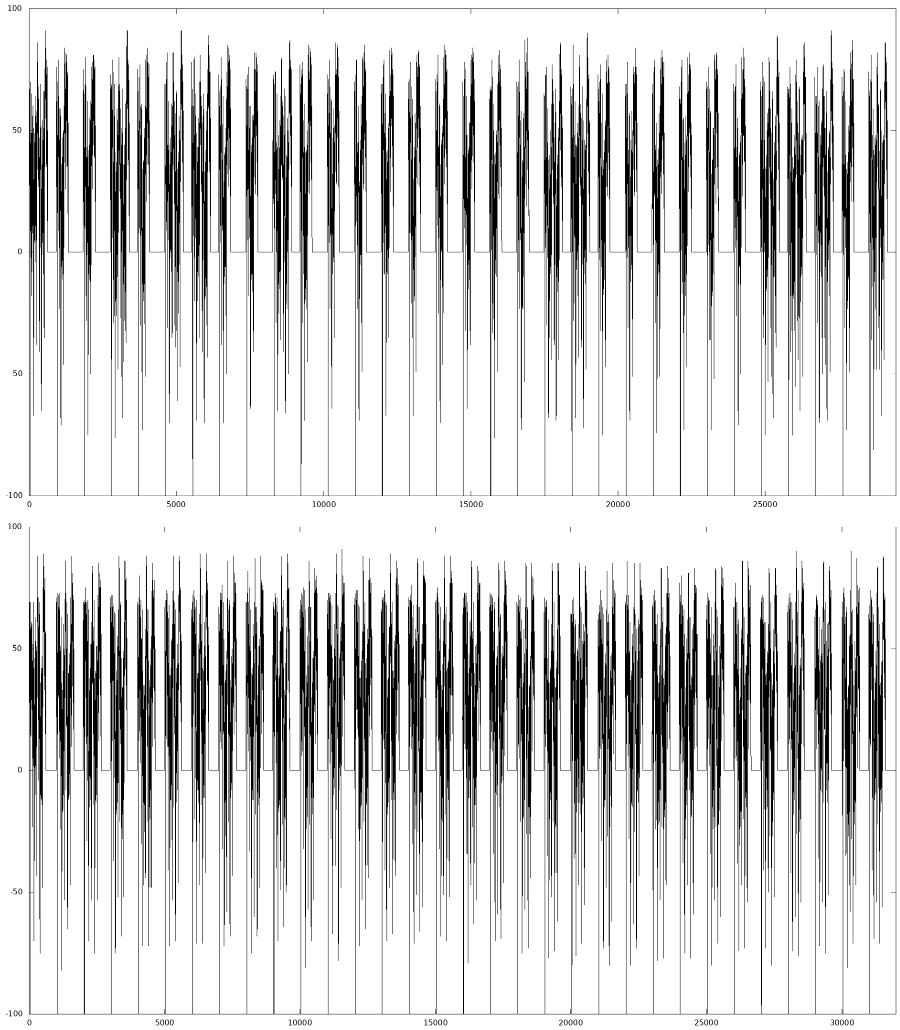

Figure 8: Power profiles for sqmul (above) and ladder (below)

is 0 or 1 two profiles for an iteration will differ only on about a third of their length. Hence sqmul and ladder profiles can differ only by at most 15% of their length, explaining why results are so close for our security metrics.

ACDL reflects the more regular pattern for ladder, while WCDL is similar and even a bit worst, which could be explained by less padding so more timestamps susceptible of variability in power and thus extreme cases.

The IIR case was simpler in Subsection 6.1 because measurements were separated by a constant step, while the choice of the resolution is less obvious for these power measurements. Before computing IIR we calibrated the resolution based on a representative iteration: we picked iteration $i = 4$ because it was balanced (17 0s for 16 1s) and generated less padding for sqmul. For a given timestamp we expect power measurements for sqmul to fall in two possible clusters, corresponding to a current bit at 0 or 1, and we focus on the last third of the power profiles where these distinct behaviors can be observed. We obtained a resolution value for every timestamp, and selected the median resolution $r = 8$ as being representative. For sake of fair comparison, we computed IIR for both sqmul and ladder with the same given resolution $r = 8$, obtaining a relative change of only 0.6%.

We experimented with a lower sampling frequency to determine its impact on our security metrics. We selected one timestamp every eight ones so that we have about 4 timestamps per clock cycle. We used again uniform distribution to compute ACDL, WCDL and IIR in row "Selective (32 bits, low sampl.)" of Table 2. Results for WCDL are smaller hence better, indicating the removal of extreme cases. Moreover, results for ACDL and IIR are similar to the high sampling rate case, indicating these metrics are robust.

We also experimented with signal processing. To reduce potential misalignments or noise we smoothed the profiles by computing for each timestamp the median value on a 31 timestamps window, i.e. one clock cycle. We used again uniform distribution to compute ACDL, WCDL and IIR in row "Selective (32 bits, smoothed)" of Table 2. Both ACDL and WCDL decreased, while IIR increased, indicating more similar measurements. The relative change of ACDL between sqmul and ladder went from -3.9% to -34.1%, while the relative change of IIR went from 0.6% to 2.5%, demonstrating that our security metrics improve with less ambiguous measurements.

But the IIR values are still very close to 1, indicating that for many timestamps all measurements fall in the same cluster, due to padding or irrelevant timestamps. Thus for this experiment we removed uninteresting timestamps, i.e. such that for every inputs all the power measurements fall in only one cluster. But this removed only 25% of the timestamps, while we are interested in only 15% of them. We used again uniform distribution to compute ACDL, WCDL and IIR in row "Selective (32 bits, filtered)" of Table 2. Without surprise, the WCDL values have not changed because they indicate differences in measurements hence the corresponding timestamps have not been impacted by the filter. The relative change of ACDL between sqmul and ladder went from -3.9% to -14.4% because the average is less diluted by null distances. The relative change of IIR went from 0.6% to 2.9%, increasing the sensitivity of the IIR even more than smoothing the profiles. This indicates that this security metric could be applied to power analysis, but requires to focus on parts of the traces where the variability depends on the secret.

Computing ACDL and WCDL costed 2 to 3 minutes for the high sampling (250 Mhz) cases and around 20 seconds for the low sampling (1 over 8 timestamps) case, while the IIR costed 2 hours for the high sampling case and 15 minutes for the low sampling case. This may seem to contradict the larger complexity of $\mathcal{O}(n_K^2 \times n_X)$ for ACDL and WCDL in Subsection 4.3 and the smaller complexity of $\mathcal{O}(n_K \times \log n_K \times n_X)$ for IIR in Subsection 5.4. But the high sampling cases used $n_X = 471040$ public tuples for sqmul and 512000 for ladder, while the low sampling cases used eight time less public tuples. Thus, the number $n_K = 33$ of secret tuples is negligible compared to the number of public tuples. Actually, our security metrics took about 8 times more execution time to compute for 8 times more timestamps, which is consistant with our complexity analysis. It appears that the cost of computing entropy is about 45 times larger than the simple arithmetical operations computed for the norm-based topological metrics, thus a larger constant for the IIR complexity.

7 Related Work

We compare in this section our Indiscernibility Methodology with existing techniques or metrics that have a similar scope, like the side-channel vulnerability factor in Subsection 7.1, or borrow from the same information-theoretic background, like the mutual information analysis in Subsection 7.2 and the perceived information in Subsection 7.3.

7.1 Side-channel Vulnerability Factor

The *Side-channel Vulnerability Factor* (SVF) is a security metric for cache attacks introduced in [DMWS12] and criticized in [ZLCL13]. The attacker tends to exploit correlations between patterns observed through side-channel and patterns in the victim execution. These patterns are analyzed as similarity matrices, and the SVF is computed as the correlation coefficient (CC) between the attacker and victim matrices.

This approach is similar to the Indiscernibility Methodology by their dependency over a particular leakage channel, but the Indiscernibility Methodology uses only inputs as labels and for a direct comparison between leakages instead of relying on intermediate values, so the SVF approach cannot be seen as a black-box approach. As explained in [ZLCL13], SVF does not distinguish between public and secret information of the system, while this is a core feature of our Indiscernibility Methodology. The similarity matrices of SVF require the computation of topological distances, like in our novel topological security metrics in Section 4, but we use them directly to compute the metrics instead of using them to compute a CC. Moreover, the CC is able to detect only linear correlations, while our novel information-theoretic security metric in Section 5 uses Mutual Information (MI), which is able to detect any kind of dependency.

7.2 Mutual Information Analysis

The *Mutual Information Analysis* (MIA) has been introduced at CHES 2008 [GBTP08], and consists in using mutual information (MI) instead of correlation coefficient (CC) as distinguisher in Correlation Power Analysis (CPA). The aim was to exploit any kind of dependency (not only linear correlations), with no knowledge about the leakage being required. [VCS09, PR09, BGP+11] indicate that, in a context of high linear correlation between the chosen model and the leakages, CC is more effective than MI as distinguisher to optimise for success ratio, even at second-order [VCS09, BGP+11]. But MIA is still promising for higher orders or if the device is too complex to retrieve a simple model. For instance [PR09] proved that if the attacker model is wrong (e.g. one bit leaks way more than the others) then MI performs arbitrarily better than CC or other divergences. But like CPA, MIA is dependent on the choice of the relevant intermediate values (based on a knowledge of the code) and the choice of the model (based on a knowledge of the device), knowledge which might not be easy to obtain in more exotic examples than AES or DES. These choices help target the attack (which makes it more effective), but are loss of information from an information-theoretic perspective. To claim more generality, [BGP+11] also evaluated the identity model directly corresponding to intermediate values, but [VCS09] remarked that for the attack to succeed the model function should not be injective or be the identity. Moreover, MI between the model and leakage functions should not be negligible, thus a relevant model should be chosen. Our methodology is more direct but less targeted: we evaluate MI directly between secret inputs K and attacker information (L, X) without considering particular intermediate values. Our methodology is thus more general: we do not assume any specific way of retrieving the key, in particular we are model-independent. Our methodology may be less efficient to retrieve a key than targeted methods, but we aim at evaluating the code leakage itself, not at estimating how fast it is to retrieve a key in a particular way. Finally, [VCS09, BGP+11] proposed variants of MIA for higher orders, while our metrics deals natively with any order.

7.3 Perceived Information

In [SMY09] is introduced the following formula for MI between inputs X and attacker observations L:

$$\mathrm{MI}(X; L) = \mathrm{H}(X) - \sum_{x \in \mathcal{X}} \mathrm{p}(x) \sum_{\ell \in \mathcal{L}} \mathrm{p}(\ell \mid x) \log_2 \mathrm{p}(x \mid \ell)$$

where $\mathrm{p}(x)$ depends only on inputs but $\mathrm{p}(\ell \mid x)$ and $\mathrm{p}(x \mid \ell)$ depends on the chip. If $\mathrm{p}(x \mid \ell)$ is replaced by the adversary model estimate obtained by template attack [CRR03], and $\mathrm{p}(\ell \mid x)$ by the sample estimate, then we obtain $\mathrm{PI}(X; L)$ the *perceived information* (PI) from [RSVC+11]. Like our discernible information presented in Section 5, a split is proposed between public inputs X and secret inputs K in order to study $\mathrm{PI}(K; L, X)$, but it is not used in more recent works [LMMS17,KBBS21] while it is fundamental in our methodology. We don't use sample estimate $\mathrm{p}(\ell \mid x)$, we rely more directly on the known distribution of the inputs, and compute $\mathrm{p}(\ell, x)$ from dendrogram clustering in Subsection 5.3. PI is based on the adversary model estimate $\mathrm{p}(x \mid \ell)$ and used for comparison purpose to analyse the effects of noise and platform variations. Our Indiscernible Information Ratio (IIR) has no unit and is designed to be more intrinsic. Our methodology does not require profiling, does not rely on noise analysis, and aims at obtaining a metric more independent from platforms. In other words, even though the mathematical tools are the same, the data and purpose are very different.

8 Conclusion

In this paper, we proposed the *Indiscernibility Methodology*, a novel and original black-box methodology to quantify with no prior knowledge the information intrinsically leaked from programs. The Indiscernibility Methodology can be used to help system developers quantify the vulnerability of their program wrt. side-channel attacks. It is general, making no assumption on the way used by the attacker to retrieve secret information, and universal, being applicable to any type of side-channel leakage (time, energy, power, EM, etc.).

From this methodology, we proposed two kinds of novel security metrics with strong formal foundations. The first kind derives from topological security, with distances defined between secret-dependent observations. Our novel *Average-Case Discernible Leakage* (ACDL) metric quantifies how difficult it is for the attacker to find a relevant point of interest, while our novel *Worst-Case Discernible Leakage* (WCDL) metric quantifies the worst possible discernibility when the corresponding point of interest has been found by the attacker.

The second kind derives from information-theoretic security, applying Shannon's entropy on the considered channels of information. Our novel *Indiscernible Information Ratio* (IIR) metric quantifies the proportion of what remains of the exploration space for the secret after observation by the attacker. It exploits any statistical correlation, unlike common security metrics detecting only linear correlations.

We demonstrated the concrete applicability of our methodology by providing experimental results based on measurements for both time and power leakages. These results show that our novel ACDL, WCDL and IIR security metrics perform according to expectations, and can be used for system development and to help prevent side-channel attacks.

Future works include: analysing our metrics behavior in the presence of noisy measurements; extending our methodology to the analysis of joint side-channels (when several channels providing information leakage are simultaneously exploited); analyzing the behavior of our metrics for non-uniform distributions of inputs (depending on code executed before the studied code); and using a path analysis to identify from our security metrics the most leaking part of the code, to provide the developper with insights on potential security patches.

Ackowledgments

The authors want to thank Nicolas Kiss for the timing and power measurements, Céline Minh for the early prototyping of our security metrics, Bruno Lebon for developing the tool computing these metrics, and Tania Richmond for her help on the Barrett's reduction. This work was supported by EU Horizon 2020 project *TeamPlay*[13], grant number 779882.

References

[APS19] Melissa Azouaoui, Romain Poussier, and François-Xavier Standaert. Fast side-channel security evaluation of ecc implementations. In Ilia Polian and Marc Stöttinger, editors, *Constructive Side-Channel Analysis and Secure Design*, pages 25–42, Cham, 2019. Springer International Publishing.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *International workshop on cryptographic hardware and embedded systems*, pages 16–29. Springer, 2004.

[BGP$^+$11] Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual information analysis: a comprehensive study. *Journal of Cryptology*, 24(2):269–291, Apr 2011.

[13] https://www.teamplay-h2020.eu

[BSS19] Olivier Bronchain, Tobias Schneider, and François-Xavier Standaert. Multi-tuple leakage detection and the dependent signal issue. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(2):318–345, Feb. 2019.

[CD13] Jeremy Cooper and E. F. J. Demulder. Test vector leakage assessment (tvla) methodology in practice (extended abstract). In *ICMC 2013*, 2013.

[CK14] Omar Choudary and Markus G. Kuhn. Efficient template attacks. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications*, pages 253–270, Cham, 2014. Springer International Publishing.

[CRR03] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski, çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 13–28, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[DMWS12] John Demme, Robert Martin, Adam Waksman, and Simha Sethumadhavan. Side-channel vulnerability factor: A metric for measuring information leakage. In *Computer Architecture (ISCA), 2012 39th Annual International Symposium on*, pages 106–117. IEEE, 2012.

[GBTP08] Benedikt Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual information analysis a generic side-channel distinguisher. *Lecture Notes in Computer Science*, 5154:426–442, 2008.

[GJJR11] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for side-channel resistance validation, niat, 2011.

[Hot31] Harold Hotelling. The generalization of student's ratio. *Ann. Math. Statist.*, 2(3):360–378, 08 1931.

[KB07] Boris Köpf and David Basin. An information-theoretic model for adaptive side-channel attacks. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 286–296. ACM, 2007.

[KBBS21] Dina Kamel, Davide Bellizia, Olivier Bronchain, and François-Xavier Standaert. Side-channel analysis of a learning parity with physical noise processor. *Journal of Cryptographic Engineering*, 11:1–9, 06 2021.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, pages 388–397, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[Koc96] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Neal Koblitz, editor, *Advances in Cryptology — CRYPTO '96*, pages 104–113, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[KSAG03] Alexander Kraskov, Harald Stögbauer, Ralph Andrzejak, and Peter Grassberger. Hierarchical clustering based on mutual information. *Computing Research Repository*, 12 2003.

[LM87] Peter L. Montgomery. Montgomery, P.L.: Speeding the Pollard and Elliptic Curve Methods of Factorization. Math. Comp. 48, 243-264. *Mathematics of Computation - Math. Comput.*, 48:243–243, 01 1987.

[LMMS17] Joseph Lange, Clément Massart, André Mouraux, and Francois-Xavier Standaert. Side-channel attacks against the human brain: The pin code case study. In Sylvain Guilley, editor, *Constructive Side-Channel Analysis and Secure Design*, pages 171–189, Cham, 2017. Springer International Publishing.

[Mah36] P. C. Mahalanobis. On the generalized distance in statistics. *Proc. Natl. Inst. Sci. India*, 2:49–55, 1936.

[MOP07] S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks – Revealing the Secrets of Smart Cards.* Springer, 2007.

[MvOV96] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.

[PR09] Emmanuel Prouff and Matthieu Rivain. Theoretical and practical aspects of mutual information based side channel analysis. In Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud, editors, *Applied Cryptography and Network Security*, pages 499–518, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[RSVC$^+$11] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, pages 109–128, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[SMY09] François-Xavier Standaert, Tal G. Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, pages 443–461, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[VCS09] Nicolas Veyrat-Charvillon and François-Xavier Standaert. Mutual information analysis: How, when and why? In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009*, pages 429–443, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[Wel47] B. L. Welch. The generalization of 'student's' problem when several different population varlances are involved. In *Biometrika*, volume 34, pages 28–35, 1947.

[ZLCL13] Tianwei Zhang, Fangfei Liu, Si Chen, and Ruby B Lee. Side channel vulnerability metrics: the promise and the pitfalls. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, page 2. ACM, 2013.