Revisiting Mutual Information Analysis: Multidimensionality, Neural Estimation and Optimality Proofs

Anonymous Submission

Abstract. Recent works showed how Mutual Information Neural Estimation (MINE) could be applied to side-channel analysis in order to evaluate the amount of leakage of an electronic device. One of the main advantages of MINE over classical estimation techniques is to enable the computation between high dimensional traces and a secret, which is relevant for leakage assessment. However, optimally exploiting this information in an attack context in order to retrieve a secret remains a non-trivial task especially when a profiling phase of the target is not allowed.

Within this context, the purpose of this paper is to address this problem based on a simple idea: there are multiple leakage sources in side-channel traces and optimal attacks should necessarily exploit most/all of them. To this aim, a new mathematical framework, designed to bridge classical Mutual Information Analysis (MIA) and the multidimensional aspect of neural-based estimators, is proposed. One of the goals is to provide rigorous proofs consolidating the mathematical basis behind MIA, thus alleviating inconsistencies found in the state of the art.

This framework allows to derive a new attack called Neural Estimated Mutual Information Analysis (NEMIA). To the best of our knowledge, it is the first unsupervised attack able to benefit from both the power of deep learning techniques and the valuable theoretical properties of MI. From simulations and experiments conducted in this paper, it seems that NEMIA performs better than classical and more recent deep learning based unsupervised side-channel attacks, especially in low-information contexts.

Keywords: Side-channel analysis, Mutual information, Deep learning, Multidimensionality, MINE

1 Introduction

1.1 Context

Side-Channel Analysis (SCA) could be defined as the process of gaining information on a secret hold by a system through leakage that comes from its practical implementation. In the most famous examples, an adversary exploits physical leakages of an electronic device such as its power consumption [KJJ99] or Electromagnetic (EM) emanations [QS01] to recover a cryptographic key. Many other side-channels have been pointed out in the literature such as timing attacks [Koc96], cache monitoring [Per05] or even network packets length analysis [SSH$^+$14]. In any case, the problem can be reduced to the following form: an adversary is able to learn realizations of a leakage variable L, often called a trace, and aims at using it to infer information about another related secret variable S.

From an information theory point of view, the maximum amount of information one could extract from a side-channel trace is bounded by the Mutual Information $\mathcal{I}(S, L)$. This quantity is, indeed, central in the side-channel domain. The goals of the different actors could be summarized as follows:

- **Designers** aim at implementing countermeasures to decrease as much as possible $\mathcal{I}(S, L)$, under computational and efficiency constraints.
- **Evaluators** aim at estimating $\mathcal{I}(S, L)$ as closely as possible to assess leakages in a worst-case scenario.
- **Attackers** aim at developing strategies to partially or fully exploit $\mathcal{I}(S, L)$ in order to recover a secret.

The main problem with this paradigm is that $\mathcal{I}(S, L)$ is famously hard to estimate from drawn samples when the variables live in a high dimensional space, which is generally the case of L (*i.e.* power traces often consist of thousands of time samples). Classical MI estimators suffer from the so called "curse of dimensionality" and require an exponential (w.r.t. the dimension) amount of data to produce reliable results. This explains why, despite its valuable theoretical properties, $\mathcal{I}(S, L)$ is not directly used for side-channel analysis. Instead, one often compute $\max_i \mathcal{I}(S, L[i])$ where $L[i]$ stands for the i-th sample of the trace, but this may not represent the true available information when multiple samples leak or when there exist some dependencies between these samples.

However, in a recent work [CLM20], authors took advantage of a new deep learning technique called Mutual Information Neural Estimation (MINE) [BBR+18] to develop a side-channel tool able to reliably estimate the MI between the secret and full traces, drastically reducing the impact of high dimensionality on the estimation reliability. This tool allows one to get an absolute leakage quantification from raw traces which is helpful for designers or evaluators to perform leakage assessment. However, knowing the amount of potentially usable information is not the same as actually exploiting it to retrieve a secret, and authors left open questions regarding this tool from the attacker's point of view. Is an adversary also able to use the inherent multidimensional properties of MINE to exploit at the same time all the potential leakage sources ? And if so, what is the optimal way to do it ? This paper aims at answering these questions.

Side-channel attacks are mainly divided into two categories: supervised SCA, where the adversary can first perform a characterization of the target, and unsupervised SCA in which this profiling step is not possible. For profiled SCA, one is theoretically able to exploit all the information $\mathcal{I}(S, L)$ by perfectly learning the target's leakage model during the characterization phase. Deep learning attacks have been shown to effectively extract all the available information when using the negative log likelihood as loss function [MDP19]. Therefore the problem is closed, at least in theory, for profiled SCA.

However, this is not the case for unsupervised attacks, where the true leakage model of the target is unknown to the adversary. In this situation, only a fraction of $\mathcal{I}(S, L)$, which value depends on the correctness of one's *a priori* on the leakage model, can be exploited. For example, the Correlation Power Analysis (CPA) [BCO04] is efficient for linear dependencies between the leakage and a certain function of the intermediate variable (often being the Hamming weight function). The Linear Regression Analysis (LRA) [DPRS12] also assumes a linear dependency but can handle different weights for each bit of the intermediate variable.

Mutual Information Analysis (MIA), however, has been introduced as a generic strategy able to capture any kind of dependencies. Papers addressing the theoretical background behind MIA [GBTP08, PR09, VCS09, BGP+11] all present MIA as SCA distinguisher able to recover the correct key without any knowledge on the target nor on its leakage model. However, this leakage model free strategy only works to target non-bijective intermediate variables which makes it well suited for the DES (as the DES S-boxes are not bijectives) but less suited for more recent algorithms such as the AES. This explains why MIA has not often been used in practice.

A second version of the MIA allowing to target any intermediate variables (and is therefore applicable in many more contexts) has also been developed. These two versions are not separated in the literature but we decided to do so in this paper to clarify the relationship between MIA and leakage model *a priori*. Indeed, this second version is not leakage model free *i.e.* it requires an *a priori* on the leakage model to work. However, one of the main advantages of this attack is that it is not limited to linear leakage model and more generally, does not require any assumption on the leakage distribution (as long as the adversary's *a priori* is sufficiently correct). However, this gain in genericity comes at the cost of efficiency: CPA has almost always been proved to work better than MIA in classical attack scenarios since leakage models are often linear. Therefore MIA is more seen as a great tool in theory that does not offer much in practice.

However, one of the main advantages of MIA is that it generalizes well to higher dimension variables and offers a way to potentially use a bigger part of the information contained in a side-channel trace. This has not really been used in the literature (except to extend MIA for masked implementation [PR09,BGP+11]) due to MI estimators limitations. But recent breakthroughs regarding neural estimation encourages to revisit classical MIA in order to make it highly multidimensional, to get closer to an optimal attack regarding the amount of information being used from the traces.

Even if neural estimation techniques can be applied in the leakage model free version of the MIA, we are more interested in the second version of MIA since it does not impose restrictions on the targeted algorithm. However, we argue that the mathematical framework behind this version (developed in [GBTP08,PR09,VCS09,BGP+11]) is not complete or even wrong at some points and rely too much on intuition instead of proofs. As a result, it is difficult to derive the best way to use the new MI estimators, especially in the context of high dimensional variables, where intuition quickly falls short. That is why rebuilding a mathematical framework along with rigorous proofs on how to conduct an optimal multidimensional MIA is one of the contributions of this paper.

1.2 Contributions

1. Clarifying the State Of The Art (SOTA) around the MIA.

   We explicitly split MIA into two different versions (this is not explicitly done in the SOTA), to help understanding the need or not of an a priori on the leakage model (2.2). We then highlight inconsistencies with the second version mainly related to the fact that MIA relies on a distinguisher computing a score for each key hypotheses, but the wrong hypotheses scores are not taken into account in the analysis (2.3). This leads us to define a new generic version of MIA which objective is related to a maximization problem that includes the impact of the wrong hypotheses scores (2.4).

2. Providing rigorous proofs to analytically solve the mathematical problems emerging from our new version of MIA.

   One of the main contributions of this paper, given by theorem 1 (2.5), is to solve the optimization problem defined in (2.4). Then, theorem 2 provides an extension of the analysis in the context of masking (3). Both theorems are designed to take into account the potential multidimensionality of the leakage and therefore are suited to support the use of the new neural MI estimators.

3. Presenting a new unsupervised multidimensional attack: the Neural Estimated Mutual Information Analysis (NEMIA).

   Mathematical results are then combined with recent breakthroughs regarding neural MI estimation in high dimension. This allows to derive, to the best of our knowledge, the first unsupervised side-channel attack able to benefit from both deep learning techniques (highly multidimensional, no pre-processing of the data...) and the valuable theoretical properties of MI (4).

4. Providing Simulations and experiments to support the analysis.

   Simulations are provided both to empirically validate the mathematical analysis as well as to gain intuition about their meaning and about which situations are best suited for the use of NEMIA (5). Eventually, practical experiments on the ASCAD database (both on raw traces and on artificially noised traces) are conducted and show that this new attack seems to outperform classical SCA strategies in terms of number of traces needed and noise resiliency (6).

2 Mutual Information Analysis

2.1 Background

Notations. Random variables are represented as upper-case letters such as X. They take their values in the corresponding set $\mathcal{X}$ depicted with a calligraphic letter. Lower case letters such as x stand for elements of $\mathcal{X}$. Probability density function associated to the variable X is denoted by P_X (replaced by P when there is no ambiguity).

Information theory. The entropy $\mathcal{H}(X)$ [Sha48] of a random variable is a fundamental quantity in information theory which indicates how much information one would gain, in average, by learning a particular realization x of X. It is defined as the expectation of the self-information $log_2(1/p_X)$. In a discrete context:

$$\mathcal{H}(X) = \sum_{x \in \mathcal{X}} P_X(x) \cdot log_2(\frac{1}{P_X(x)}) \tag{1}$$

In a side-channel environment where L represents the acquired data, one is not interested in the absolute information provided by X but rather in the amount of information revealed about a second variable such as a secret S. This is exactly what is measured by the mutual information $\mathcal{I}(S, L)$. It is defined as:

$$\mathcal{I}(S, L) = \mathcal{H}(S) - \mathcal{H}(S \mid L) = \mathcal{H}(L) - \mathcal{H}(L \mid S) \tag{2}$$

where $\mathcal{H}(A \mid B)$ stands for the conditional entropy of A knowing B:

$$\mathcal{H}(A \mid B) = \sum_{b \in \mathcal{B}} P_B(b) \cdot \mathcal{H}(A \mid B = b) \tag{3}$$

Another useful way to characterise $\mathcal{I}(S, L)$ is to express it as the Kullback-Leibler (KL) divergence between the joint distribution and the product of the marginals:

$$\begin{aligned} \mathcal{I}(S, L) &= D_{KL}(P_{S,L} \mid\mid P_S \otimes p_L) \\ &= \sum_{s \in \mathcal{S}} \sum_{l \in \mathcal{L}} P(s, l) \cdot log(\frac{P(s, l)}{P(s) \cdot P(l)}) \end{aligned} \tag{4}$$

Unsupervised attacks. Suppose an adversary wants to recover the secret key used by the physical implementation of a cryptographic algorithm. He has access to a set of measurements (traces) $(L_i)_{1\leq i\leq n}$ labeled with the plaintext P_i used for the encryption. The general idea of an unsupervised side-channel attack is to make a series of hypotheses k_i, on a sub-part of the key and to use a distinguisher $\mathcal{D}(k)$ allowing to rank the different candidates. Distinguishers use statistical dependencies between traces and an intermediate variable $Z_{k^*} = g(P, k^*)$ that depends on the plaintext and the correct key k^* through a deterministic function $g : \mathcal{P} \times \mathcal{K} \to \mathcal{Z}$ related to the underlying algorithm. For simplicity, $g(P, k)$ is denoted $g_k(P)$ in the rest of the paper.

Common distinguishers such as Pearson's coefficient or coefficient of determination in a linear regression exploit some *a priori* on the leakage model. A common intuition about mutual information used as a distinguisher [GBTP08] is that it has been introduced precisely to reduce the need to have an *a priori*. It is often found in the literature (*e.g.* [BGP+11]) that it aims at generality, leading to successful attacks without requiring specific knowledge or assumptions on the target. While this is true in some sense, this assertion is mitigated hereafter.

2.2 State of the art

This section presents the state of the art of MIA [GBTP08, PR09, VCS09, BGP+11] and is organized to discuss and clarify the importance of the adversary's leakage model *a priori*.

MIA uses a distinguisher $\mathcal{D}$ which takes the following form[1]:

$$\mathcal{D}(k) = \mathcal{I}\big(f(Z_k), L\big) \tag{5}$$

with f being a function transforming the guessed intermediate variables Z_k. This function is one of the main concerns of this paper. It is often called the "model" of the adversary. The requirement of a model may seem contradictory with the claims of genericity of the MIA. Actually, this model can be replaced by the identity function making the MIA independent of any *a priori* on the leakage model. This version of the MIA is presented hereafter.

MIA version 1. (Leakage model free) In its most basic form, MIA uses $\mathcal{I}(Z_k = g_k(P), L)$ as a distinguisher, making hypotheses on k. With $\varphi : \mathcal{Z} \to \mathbb{R}^n$ representing the leakage model of the target, L can be written as $L = \varphi(Z_{k^*}) + N$, with N being a random variable independent of Z_k for all k, and representing the noise. With these notations, the distinguisher becomes:

$$\mathcal{D}(k) = \mathcal{I}\big(Z_k, \varphi(Z_{k^*}) + N\big) \tag{6}$$

Proposition 1. *This distinguisher is maximized for the correct key hypothesis k^*.*

Proof. Using equation 2, for any $k \in \mathcal{K}$:

$$\begin{aligned} \mathcal{D}(k^*) - \mathcal{D}(k) &= \mathcal{H}(L) - \mathcal{H}(L \mid Z_{k^*}) - \big[\mathcal{H}(L) - \mathcal{H}(L \mid Z_k)\big] \\ &= \mathcal{H}\big(\varphi(Z_{k^*}) + N \mid Z_k\big) - \mathcal{H}\big(\varphi(Z_{k^*}) + N \mid Z_{k^*}\big) \end{aligned} \tag{7}$$

Since adding knowledge can only decrease entropy:

$$\mathcal{D}(k^*) - \mathcal{D}(k) \geq \mathcal{H}\big(\varphi(Z_{k^*}) + N \mid Z_k, Z_{k^*}\big) - \mathcal{H}\big(\varphi(Z_{k^*}) + N \mid Z_{k^*}\big) \tag{8}$$

Now using the independence of N and the fact that $\varphi(Z_{k^*})$ is entirely determined by Z_{k^*}:

$$\begin{aligned} \mathcal{D}(k^*) - \mathcal{D}(k) &\geq \mathcal{H}(N) - \mathcal{H}(N) \\ &\geq 0 \end{aligned} \tag{9}$$

which concludes the proof. □

[1] Due to MI estimator limitations, $\mathcal{D}(k)$ is often replaced in practice by $\max_i \mathcal{I}(f(Z_k), L[i])$, where $L[i]$ represents the i-th sample of the trace. This does not affect the theory described in this section so we decided to keep it as described in eq. 5 for the sake of simplicity. More details are provided in section 4.2.

This strategy does not require any assumption on the leakage model of the target. However, it only works if the correct key hypothesis is distinguishable from the other ones, or, in other words, if $\mathcal{D}(k) < \mathcal{D}(k^*), \forall k \neq k^*$, which is not guaranteed by proposition 1. An important property of the MI is that it is preserved by injective transformations of the input variables. So if different key hypotheses yield Z_k variables differing from each other only by a permutation (for example if the g_k functions are bijective), $\mathcal{I}(Z_k, L)$ would be constant for all k and the distinguisher could not discriminate key candidates. Therefore, g_k has to be non-injective. For example, one could target the output of the first round DES S-box.

While this form of MIA is effectively leakage model free, it comes with a huge constraint since in many interesting cases g_k is bijective. In the AES case, this means that one cannot target the output of the first S-box since $Sbox[k^* \oplus P]$ is bijective with P. In [PR09] and [RGV14a], authors suggest to target the bitwise addition between two S-box outputs during the *MixColumns* operation. This requires making hypotheses on 16 bits of the key (leading to 2^{16} MI computations). Moreover, it is only feasible if this operation leaks enough information which may not be the case in practice. Indeed, for hardware implementations, this step is usually fully combinatorial and does not use any register. This explains why most of the MIA experiments in the literature have been performed on the DES.

MIA version 2. (Leakage model dependent) It is still possible to target $Z_{k^*} = g_{k^*}(P)$ for bijective g_k functions. The idea is to apply a non-injective function f to Z_k and use $\mathcal{I}(f(Z_k), L)$ as distinguisher. The application of f create a partition of $\mathcal{Z}$ so f will be called the "partition function" in the rest of this paper. Since no data transformation can create information (this is the so called data processing inequality [BR12]), the application of f can only decrease the initial information: $\forall f, \forall k,\ \mathcal{I}(f(Z_k), L) \leq \mathcal{I}(Z_k, L)$. The goal is then to find a function that decreases more $\mathcal{I}(Z_k, L)$ than $\mathcal{I}(Z_{k^*}, L)$ and therefore, enhance the discriminating power of the analysis.

For example, assuming that bits leak independently, [GBTP08] proposes to drop one bit of Z. This is equivalent to redefine the intermediate variable as a restrictive number of bits of $g_{k^*}(P)$, and apply MIA version 1 with no partition function. Another idea is to use a guessed version $\bar{\varphi}$ of the leakage model φ. Indeed, $\mathcal{I}\big(\varphi(Z_k), \varphi(Z_{k^*}) + N\big)$ is clearly maximized for $k = k^*$. Therefore, if $\bar{\varphi}$ is not too far from φ, $\mathcal{I}\big(\bar{\varphi}(Z_k), \varphi(Z_{k^*}) + N\big)$ may still be maximized for $k = k^*$. It is shown in [VCS09] that error in the approximation of φ may be less penalizing than for other attacks.

In addition, MIA is more flexible in the sense that it is not limited to exploit linear dependencies and gives an option to mount a successful attack with any leakage model. However, it should be emphasized that, for this version, the adversary must have a good enough *a priori* on the leakage, otherwise, the attack is unsuccessful. A suitable choice for the partition function necessarily uses assumptions on φ.

While we think this point needed to be clarified, we do not see this as a criticism of MIA. As stated in [WOS14], hopes of finding a leakage model free strategy able to target a bijective intermediate variable are vain, even outside the context of MIA. We present hereafter a synthetic proof of the main result of [WOS14].

Proposition 2. *Let g_k be a bijective map for all k. For any strategy $\mathfrak{S}$ which takes as input a set of traces $\vec{L} = \big(\varphi(g_{k^*}(P_i))\big)_{1 \leq i \leq n}$ and outputs a ranking of the different key hypotheses, there exists a leakage model $\tilde{\varphi}$ that would rank k^* in the last position such that the attack completely fails.*

Proof. First, apply $\mathfrak{S}$ on traces obtained through any leakage model φ_0 and denote by $\bar{k}$ the last key returned by $\mathfrak{S}$. Now, for all P, define $\tilde{\varphi}_0(g_{k^*}(P)) = \varphi_0(g_{\bar{k}}(P))$, which completely defines $\tilde{\varphi}_0$ as $g_{\bar{k}}$ is bijective. Applying $\mathfrak{S}$ on traces obtained through $\tilde{\varphi}_0$ would now rank k^* in the last position. □

This proposition shows that there does not exist any generic distinguisher, that would both:

1) Exploit statistical dependencies between traces and an intermediate variable bijectively related to the plaintext.

2) Work whatever the leakage model of the target.

Since MIA version 2, with a fixed partition function, verifies 1), it necessarily fails for some leakage models or, in other words, has to use an assumption on the leakage model to succeed. Even though it requires an analysis on what partition function should be used, the rest of this paper is more focused on MIA version 2 since it is more generic in the sense that it can be applied in many more situations.

2.3 About the distinguishability

As stated in [WO11], even if $\mathcal{D}(k)$ is maximized for $k = k^*$, it is not enough to guarantee a successful attack in practice, when noise comes into play. What is really important is the difference between $\mathcal{D}(k^*)$ and the others, or in other words, the distinguishability of the correct hypothesis through the distinguisher $\mathcal{D}$. The idea found in the literature is that for a wrong key hypothesis:

«false predictions will form a partition corresponding to a random sampling of [L] and therefore simply give scaled images of the global side-channel probability density function. Hence, the estimated mutual information will be equal (or close) to zero in this case.» [BGP+11].

We do not agree with this fact since the wrong hypotheses scores totally depend on the partition function f and on g_k. As explained in the previous section, if the g_k's are bijective, all the scores would be equal if f is also bijective. This fact is well noted in all the papers about MIA but we would like to emphasize that even for non-bijective f the wrong hypotheses score depends on the "degree of bijectiveness" of f. Intuitively, the more compact f is (in the sense of more collisions through f) the smaller the wrong hypotheses scores would be. But the same is true for the correct score which means that there is a trade-off between how much one wants to decrease $\mathcal{I}\big(f(Z_k), L\big)$ for the wrong k and keep $\mathcal{I}\big(f(Z_{k^*}), L\big)$ high, to enhance the distinguishability.

2.4 Towards an optimal partition function *f*

In the SOTA, the partition function is not seen as a parameter on which a maximization research could be done. Therefore, no research on finding the optimal function f has been conducted. It is generally fixed to one or two constant choices, except in [PR09] where authors proposed that f could be a generic function. However, it is stated that the adversary:

«does not need a good linear approximation of φ but only a function [f] such that the mutual information $\mathcal{I}\big(f(Z_{k^*}), \varphi(Z_{k^*})\big)$ is non-negligible » [PR09].

Again, this condition is necessary but not sufficient. Even if bijective functions are excluded one can create the following f_0 function such that:

$$f_0(x) = \begin{cases} 0, \text{ if } x \in \{0, 1\} \\ x, \text{ else} \end{cases} \tag{10}$$

Being almost the identity function, f_0 is such that $\mathcal{I}\big(f_0(Z_{k^*}), \varphi(Z_{k^*})\big)$ is high but would have a very low discriminating power. This shows that the wrong hypotheses scores can not be left out of the analysis. One typically wants to find the f function maximizing the distinguishability of the correct hypothesis. Several criterion has been studied in the literature [WO11, RGV14b]. In this paper we chose to use the nearest rival criterion[2]. Therefore, let us define the optimal set of functions $\mathcal{F}_{opt}$ as:

$$\mathcal{F}_{opt} = \underset{f\colon \mathcal{Z}\to\mathbb{R}^n}{\arg\max} \left\{ \mathcal{I}\big(f(Z_{k^*}), L\big) - \max_{k\neq k^*} \left[\mathcal{I}\big(f(Z_k), L\big)\right] \right\} \tag{11}$$

$\mathcal{F}_{opt}$ is a set since the maximum is reached by an infinite amount of functions. Indeed, if $f_{opt} \in \mathcal{F}_{opt}$, for any bijection b, $b \circ f_{opt}$ is also in $\mathcal{F}_{opt}$ since bijections do not affect MI. Note that f is not restricted to be one-dimensional. Its domain is set to be $\mathbb{R}^n$ where n can be any positive integer.

2.5 Analytical resolution

Being consistent with proposition 2, $\mathcal{F}_{opt}$ depends on L and therefore on the leakage model. Since knowledge on φ is required anyway, this section assumes a full knowledge on φ in order to conduct an analytical analysis to find the optimal f function. Traces are also supposed to be acquired in an ideal set-up, without noise, so that, at least for a significant sub-part of the trace, $L = \varphi(Z_{k^*})$. Bounds taking into account imperfect knowledge on φ as well as noise will be given in section 2.7.

A natural choice for the partition function would be to take $f = \varphi$ because it maximizes the left term in (11): $\mathcal{I}\big(f(Z_{k^*}), \varphi(Z_{k^*})\big)$. But it may be possible to find a function that would maximize the global objective without maximizing the left term of (11) (we emphasize that f and φ can be multi-dimensional which make the intuition harder to have). Th. 1 actually proves that it is not possible and that whatever the leakage model, $\varphi \in \mathcal{F}_{opt}$. The main demonstration requires the use of a helper which is introduced in the form of a lemma hereafter.

Lemma 1. *Let $f\colon \mathcal{Z} \to \mathbb{R}^n$ be any function. For any leakage model $\varphi\colon \mathcal{Z} \to \mathbb{R}^n$ there exists a decomposition of f into $f = f_2 \circ f_1$, with $f_1 : \mathcal{Z} \to \mathbb{N}$, $f_2 : \mathbb{N} \to \mathbb{R}^n$, satisfying the two following properties:*

1) $\exists\, f_3 : \operatorname{Im} f_1 \to \mathbb{R}^n$ such that $f_3 \circ f_1 = \varphi$

2) $\forall z \in \mathcal{Z}, f_2|_{f_1\left(\varphi^{-1}(\{\varphi(z)\})\right)}$ is bijective of reciprocal $f_2^{-1}|_{f_2\circ f_1\left(\varphi^{-1}(\{\varphi(z)\})\right)}$

Proof. The proof is given in appendix A. □

Theorem 1. *Let P follow a uniform distribution. Let Z_k represent the hypothetical intermediate variables such that $Z_k = g_k(P)$ with bijective g_k's. Let $\varphi\colon \mathcal{Z} \to \mathbb{R}^n$ be the leakage model of the target so that $L = \varphi(Z_{k^*})$. Then, $\varphi \in \mathcal{F}_{opt}$.*

Proof. Let $\mathcal{S}_f = \mathcal{I}\big(f(Z_{k^*}), L\big) - \max_{k\neq k^*}\left[\mathcal{I}\big(f(Z_k), L\big)\right]$ represent the distinguishability score for a given function f such that:

$$\mathcal{F}_{opt} = \underset{f\colon \mathcal{Z}\to\mathbb{R}^n}{\arg\max}\{\mathcal{S}_f\}$$

Since all the Z_k follow a uniform distribution (P follows a uniform distribution and the g_k functions are bijective), the entropy $H(f(Z_k))$ is equal for all k. Then using $\mathcal{I}(A, B) = H(A) - H(A \mid B)$:

$$\mathcal{S}_f = -H\big(f(Z_{k^*}) \mid L\big) + \min_{k\neq k^*}\left[H\big(f(Z_k) \mid L\big)\right] \tag{12}$$

[2] Note that other criterion such as the distance with the mean of the wrong hypotheses could also have been used without modifying the analysis as discussed in remark 1.

Symmetrically, using $\mathcal{I}(A,B) = H(B) - H(B \mid A)$:

$$\mathcal{S}_f = -H\big(L \mid f(Z_{k^*})\big) + \min_{k \neq k^*} \big[H\big(L \mid f(Z_k)\big)\big] \tag{13}$$

Let $f\colon \mathcal{Z} \to \mathbb{R}^n$ be any function. Applying lemma 1, there exist f_1 and f_2 satisfying the two properties given in lemma 1, such that $f = f_2 \circ f_1$. The goal is to show that $\mathcal{S}_f \leq \mathcal{S}_\varphi$. The proof is divided into two phases: first show that $\mathcal{S}_f \leq \mathcal{S}_{f_1}$ using (12), then show that $\mathcal{S}_{f_1} \leq \mathcal{S}_\varphi$ using (13). Let us start with (12):

$$\begin{aligned} \mathcal{S}_f &= -H\big(f_2 \circ f_1(Z_{k^*}) \mid L\big) + \min_{k \neq k^*} \big[H\big(f_2 \circ f_1(Z_k) \mid L\big)\big] \\ &\leq -H\big(f_2 \circ f_1(Z_{k^*}) \mid L\big) + \min_{k \neq k^*} \big[H\big(f_1(Z_k) \mid L\big)\big] \end{aligned} \tag{14}$$

since applying f_2 in the second term can only decrease entropy (see lemma 2). The goal is now to remove f_2 in the first term:

$$-H\big(f_2 \circ f_1(Z_{k^*}) \mid L\big) = \sum_{\substack{l \in \mathcal{L} \\ \bar{f}_2 \in \operatorname{Im} f_2}} P(l) \cdot P(\bar{f}_2 \mid l) \cdot log(P(\bar{f}_2 \mid l)) \tag{15}$$

$$\begin{aligned} P(\bar{f}_2 \mid l) &= P\big(f_2 \circ f_1(Z_{k^*}) = \bar{f}_2 \mid \varphi(Z_{k^*}) = l\big) \\ &= P\big(f_1(Z_{k^*}) \in f_2^{-1}(\bar{f}_2) \mid \varphi(Z_{k^*}) = l\big) \end{aligned} \tag{16}$$

Knowing that $\varphi(Z_{k^*}) = l$ implies that $Z_{k^*} \in \varphi^{-1}(\{l\})$ and also that $f_1(Z_{k^*}) \in f_1(\varphi^{-1}(\{l\}))$. Let $\mathcal{A}_l$ denotes $f_1(\varphi^{-1}(\{l\}))$ to avoid heavy notations. Then:

$$\begin{aligned} \varphi(Z_{k^*}) = l &\implies f_1(Z_{k^*}) \in \mathcal{A}_l \\ &\implies f_1(Z_{k^*}) \in f_2^{-1}(f_2(\mathcal{A}_l)) \end{aligned} \tag{17}$$

which means that:

$$P(\bar{f}_2 \mid l) = \begin{cases} P\big(f_1(Z_{k^*}) \in f_2^{-1}|_{f_2(\mathcal{A}_l)}(\bar{f}_2) \mid l\big) & \text{if } \bar{f}_2 \in f_2(\mathcal{A}_l) \\ 0 & \text{else} \end{cases} \tag{18}$$

Lemma 1 states that $f_2|_{\mathcal{A}_l}$ is bijective of reciprocal $f_2^{-1}|_{f_2(\mathcal{A}_l)}$, so if $\bar{f}_2 \in f_2(\mathcal{A}_l)$:

$$P(\bar{f}_2 \mid l) = P\big(f_1(Z_{k^*}) = f_2^{-1}|_{f_2(\mathcal{A}_l)}(\bar{f}_2) \mid l\big) \tag{19}$$

Let us plug this result back into (15):

$$\begin{aligned} -H\big(f_2 \circ f_1(Z_{k^*}) \mid L\big) = \sum_{l \in \mathcal{L}} \sum_{\bar{f}_2 \in f_2(\mathcal{A}_l)} & P(l) \cdot P\big(f_1(Z_{k^*}) = f_2^{-1}|_{f_2(\mathcal{A}_l)}(\bar{f}_2) \mid l\big) \cdot \\ & log\Big(P\big(f_1(Z_{k^*}) = f_2^{-1}|_{f_2(\mathcal{A}_l)}(\bar{f}_2) \mid l\big)\Big) \end{aligned} \tag{20}$$

Now, one can apply the following change of variable in the second sum: $\bar{f}_1 = f_2^{-1}|_{f_2(\mathcal{A}_l)}(\bar{f}_2)$:

$$\begin{aligned} -H\big(f_2 \circ f_1(Z_{k^*}) \mid L\big) = \sum_{l \in \mathcal{L}} \sum_{\bar{f}_1 \in \mathcal{A}_l} & P(l) \cdot P\big(f_1(Z_{k^*}) = \bar{f}_1) \mid l\big) \cdot \\ & log\Big(P\big(f_1(Z_{k^*}) = \bar{f}_1) \mid l\big)\Big) \end{aligned} \tag{21}$$

Finally, since $P\big(f_1(Z_{k^*}) = \bar{f}_1) \mid l\big) = 0$ when $\bar{f}_1 \in \operatorname{Im} f_1 \setminus \mathcal{A}_l$, one can artificially add some terms equal to 0 in the second sum:

$$\begin{aligned} -H\big(f_2 \circ f_1(Z_{k^*}) \mid L\big) &= \sum_{l \in \mathcal{L}} \sum_{\bar{f}_1 \in \operatorname{Im} f_1} P(l) \cdot P(\bar{f}_1 \mid l) \cdot log\big(P(\bar{f}_1 \mid l)\big) \\ &= -H\big(f_1(Z_{k^*}) \mid L\big) \end{aligned} \tag{22}$$

Applying this result to (14) gives:

$$\begin{aligned} \mathcal{S}_f &\leq -H\big(f_1(Z_{k^*}) \mid L\big) + \min_{k \neq k^*} \big[H\big(f_1(Z_k) \mid L\big)\big] \\ \mathcal{S}_f &\leq \mathcal{S}_{f_1} \end{aligned} \tag{23}$$

which concludes the first step of the demonstration.

Now the goal is to show that $\mathcal{S}_{f_1} \leq \mathcal{S}_{\varphi}$. Lemma 1 guarantees that there exists f_3 such that $f_3 \circ f_1 = \varphi$. Let us use this in (13):

$$\begin{aligned} \mathcal{S}_{f_1} &= -H\big(L \mid f_1(Z_{k^*})\big) + \min_{k \neq k^*} \big[H\big(L \mid f_1(Z_k)\big)\big] \\ &\leq -H\big(L \mid f_1(Z_{k^*})\big) + \min_{k \neq k^*} \big[H\big(L \mid \underbrace{f_3 \circ f_1}_{\varphi}(Z_k)\big)\big] \end{aligned} \tag{24}$$

since applying f_3 to the known variable can only increase the global entropy (see lemma 3). Now using $L = \varphi(Z_{k^*})$:

$$\begin{aligned} &-H\big(L \mid f_1(Z_{k^*})\big) \leq 0 \\ &-H\big(L \mid f_1(Z_{k^*})\big) \leq -H\big(\varphi(Z_{k^*}) | \varphi(Z_{k^*})\big) = 0 \end{aligned} \tag{25}$$

Therefore:

$$\begin{aligned} \mathcal{S}_{f_1} &\leq -H\big(L \mid \varphi(Z_{k^*})\big) + \min_{k \neq k^*} \big[H\big(L \mid \varphi(Z_k)\big)\big] \\ \mathcal{S}_{f_1} &\leq \mathcal{S}_{\varphi} \end{aligned} \tag{26}$$

Finally, using both part of the demonstration:

$$\mathcal{S}_f \leq \mathcal{S}_{f_1} \leq \mathcal{S}_{\varphi} \tag{27}$$

which ensures that φ is better or equal to any other functions and so that $\varphi \in \mathcal{F}_{opt}$. □

Remark 1. Demonstration of Th. 1 would have worked exactly the same if one had first fixed a particular hypothesis k, and tried to maximize $\mathcal{S}_{f,k} = \mathcal{I}\big(f(Z_{k^*}), L\big) - \mathcal{I}\big(f(Z_k), L\big)$. Therefore, for each k, φ maximizes the distance between the score of k^* and k which is an even stronger version of the theorem. One could not be sure that such a function would exist *a priori*, that is why $\mathcal{F}_{opt}$ has not been defined with this criterion. However, this shows *a posteriori* that Th 1 is still valid even if one decides to redefine $\mathcal{F}_{opt}$, for example using the distance with the mean (instead of the maximum) of the wrong hypotheses scores.

Interpretation. This theorem tells that to conduct an optimal MIA, one has to transform the targeted variable Z_k by applying the leakage model φ (or any bijection of φ) and use $\mathcal{I}(\varphi(Z_k), L)$ as a distinguisher. Note the multidimensional aspect of this theorem since both $\varphi(Z_k)$ and L can live in high dimensional space. This is a key point in this paper that will be discussed in detail in section 4.2 which bridges this theorem with newest multidimensional MI estimators in order to derive a new attack. Note that this theorem also implies that if the leakage model is itself bijective, MIA is not a valid strategy since the distinguishability score would be bounded by 0.

2.6 Selecting leakage model *a priori*

In a real-life experiment, one might not perfectly know the leakage model φ but only an estimation $\bar{\varphi}$. This is especially true when working in an unsupervised context. This section provides a procedure to evaluate the correctness of $\bar{\varphi}$, helping to choose from multiple guesses $\bar{\varphi}_1, \ldots, \bar{\varphi}_n$. This test relies on the following observation:

Proposition 3. *Let $L = \varphi(Z_{k^*}) + N$, with N an independent random variable representing the noise. Then:* $\varphi \in \arg\max_f [\mathcal{I}\big(f(Z_{k^*}), L\big)]$

Proof. On one hand:

$$\begin{aligned}\mathcal{I}\big(f(Z_{k^*}), L\big) &= \mathcal{H}(L) - \mathcal{H}\big(L \mid f(Z_{k^*})\big) \\ &\leq \mathcal{H}(L) - \mathcal{H}\big(\varphi(Z_{k^*}) + N \mid f(Z_{k^*}), \varphi(Z_{k^*})\big) \\ &\leq \mathcal{H}(L) - \mathcal{H}(N)\end{aligned} \tag{28}$$

and on the other hand:

$$\begin{aligned}\mathcal{I}\big(\varphi(Z_{k^*}), L\big) &= \mathcal{H}(L) - \mathcal{H}\big(L \mid \varphi(Z_{k^*})\big) \\ &= \mathcal{H}(L) - \mathcal{H}\big(\varphi(Z_{k^*}) + N \mid \varphi(Z_{k^*})\big) \\ &= \mathcal{H}(L) - \mathcal{H}(N)\end{aligned} \tag{29}$$

Then:

$$\mathcal{I}\big(\varphi(Z_{k^*}), L\big) \geq \mathcal{I}\big(f(Z_{k^*}), L\big) \tag{30}$$

which concludes the proof. □

The identity function obviously also maximizes: $\mathcal{I}\big(f(Z_{k^*}), L\big)$ so combining this with proposition 3:

$$\mathcal{I}\big(Z_{k^*}, L\big) = \mathcal{I}\big(\varphi(Z_{k^*}), L\big) \tag{31}$$

or,

$$\mathcal{I}\big(Z_{k^*}, L\big) = \max_k [\mathcal{I}\big(\varphi(Z_k), L\big)] \tag{32}$$

Then, if k^* is known (for example in an evaluation setup) one can use equation 31 and estimate $\mathcal{I}\big(Z_{k^*}, L\big)$ and $\mathcal{I}\big(\bar{\varphi}(Z_{k^*}), L\big)$ and compare them. If $\bar{\varphi}$ is a good approximation of the true underlying leakage model, one should have $\mathcal{I}\big(Z_{k^*}, L\big) \approx \mathcal{I}\big(\bar{\varphi}(Z_{k^*}), L\big)$. If k^* is unknown, the adversary can still use equation 32 estimating $\mathcal{I}\big(\bar{\varphi}(Z_k), L\big)$ for all k, and comparing the maximum with $\mathcal{I}\big(Z_{k_0}, L\big)$ (k_0 can be chosen randomly since all the Z_k variables are just permutation of each other which does not affect MI). Note that this test is only a rejection test since passing the test does not guarantee a good estimation of φ: for example, the identity function always passes the test.

2.7 Leakage model uncertainty and noise

Let assume that the adversary has chosen a given estimation $\bar{\varphi}$ of φ. Let also assume that the ideal data $L = \varphi(Z_{k^*})$, used in theorem 1, are now noisy so that the acquired data takes the following form: $\bar{L} = \varphi(Z_{k*}) + N$, with N an independent random variable. This section aims at complementing theorem 1 by lower bounding the distinguishably score $\bar{\mathcal{S}}_{\bar{\varphi}}$ that one would get in practice in such a context:

$$\bar{\mathcal{S}}_{\bar{\varphi}} = \mathcal{I}\big(\bar{\varphi}(Z_{k^*}), \bar{L}\big) - \max_{k \neq k^*} \left[\mathcal{I}\big(\bar{\varphi}(Z_k), \bar{L}\big)\right] \tag{33}$$

Our goal is to compare $\bar{\mathcal{S}}_{\bar{\varphi}}$ with the optimal score $\mathcal{S}_{\varphi}$ (from theorem 1) that one would get with the perfect knowledge of φ and un-noised data such that:

$$\mathcal{S}_{\varphi} = \mathcal{I}\big(\varphi(Z_{k^*}), L\big) - \max_{k \neq k^*} \left[\mathcal{I}\big(\varphi(Z_k), L\big)\right] \tag{34}$$

Proposition 4. $\bar{\mathcal{S}}_{\bar{\varphi}}$ *is lower-bounded by the following inequality:*

$$\bar{\mathcal{S}}_{\bar{\varphi}} \geq \mathcal{S}_{\varphi} - H(N) - H\big(\varphi(Z_{k^*}) \mid \bar{\varphi}(Z_{k^*})\big) - \max_{k \neq k^*} \left[H\big(\bar{\varphi}(Z_k) \mid \varphi(Z_k)\big)\right] \tag{35}$$

Proof. Using the same argument as in (13) one has:

$$\bar{\mathcal{S}}_{\bar{\varphi}} = -H\big(\varphi(Z_{k^*}) + N \mid \bar{\varphi}(Z_{k^*})\big) + \min_{k \neq k^*} \left[H\big(\varphi(Z_{k*}) + N \mid \bar{\varphi}(Z_k)\big)\right] \tag{36}$$

Since removing noise on the right term can only decrease entropy:

$$\bar{\mathcal{S}}_{\bar{\varphi}} \geq -H\big(\varphi(Z_{k^*}) + N \mid \bar{\varphi}(Z_{k^*})\big) + \min_{k \neq k^*} \left[H\big(\varphi(Z_{k^*}) \mid \bar{\varphi}(Z_k)\big)\right] \tag{37}$$

Now since $H(A+B) \leq H(A) + H(B)$ and using the independence of N:

$$\bar{\mathcal{S}}_{\bar{\varphi}} \geq -H(N) - H\big(\varphi(Z_{k^*}) \mid \bar{\varphi}(Z_{k^*})\big) + \min_{k \neq k^*} \left[H\big(\varphi(Z_{k^*}) \mid \bar{\varphi}(Z_k)\big)\right] \tag{38}$$

Using $H(A \mid B) \geq H(A \mid C) - H(B \mid C)$ which can be shown through information Venn diagram:

$$\begin{aligned} \bar{\mathcal{S}}_{\bar{\varphi}} &\geq -H(N) - H\big(\varphi(Z_{k^*}) \mid \bar{\varphi}(Z_{k^*})\big) + \min_{k \neq k^*} \left[H\big(\varphi(Z_{k^*}) \mid \varphi(Z_k)\big) - H\big(\bar{\varphi}(Z_k) \mid \varphi(Z_k)\big)\right] \\ &\geq -H(N) - H\big(\varphi(Z_{k^*}) \mid \bar{\varphi}(Z_{k^*})\big) + \min_{k \neq k^*} \left[H\big(\varphi(Z_{k^*}) \mid \varphi(Z_k)\big)\right] \\ &\quad - \max_{k \neq k^*} \left[H\big(\bar{\varphi}(Z_k) \mid \varphi(Z_k)\big)\right] \end{aligned} \tag{39}$$

Now let $\mathcal{S}_\varphi$ appear in the equation:

$$\begin{aligned} \min_{k \neq k^*} \left[H\big(\varphi(Z_{k^*}) \mid \varphi(Z_k)\big)\right] &= \min_{k \neq k^*} \left[H\big(\varphi(Z_{k^*}) \mid \varphi(Z_k)\big)\right] - \overbrace{H\big(\varphi(Z_{k^*}) \mid \varphi(Z_{k^*})\big)}^{0} \\ &= \mathcal{S}_\varphi \end{aligned} \tag{40}$$

So:

$$\bar{\mathcal{S}}_{\bar{\varphi}} \geq \mathcal{S}_\varphi - H(N) - H\big(\varphi(Z_{k^*}) \mid \bar{\varphi}(Z_{k^*})\big) - \max_{k \neq k^*} \left[H\big(\bar{\varphi}(Z_k) \mid \varphi(Z_k)\big)\right] \tag{41}$$

which concludes the proof. □

This proposition describes the impact of the noise and leakage model approximation in a quantitative way. Its qualitative interpretation is fairly intuitive. It clearly shows that one has two strategies to get closer to the optimal score: reducing the noise entropy or improving his guess on $\bar{\varphi}$. When $H(N)$ tends towards 0 and $\bar{\varphi}$ gets closer to φ , $\bar{\mathcal{S}}_{\bar{\varphi}}$ tends towards the optimal score $\mathcal{S}_\varphi$. It also captures the fact that bijective errors do not impact the outcome of the attack since if there exists a bijection between $\bar{\varphi}(Z_k)$ and $\varphi(Z_k)$, both terms $H\big(\varphi(Z_{k^*}) \mid \bar{\varphi}(Z_{k^*})\big)$ and $\max_{k \neq k^*}[H\big(\bar{\varphi}(Z_k) \mid \varphi(Z_k)\big)]$ would be equal to 0.

It should be noted that the given bound may not be tight especially in the context of high noise where the right term could become negative. In such a context, finding inequalities, able to control or give useful insights on $\bar{\mathcal{S}}_{\bar{\varphi}}$ is an interesting problem for further works.

3 MIA against masked implementations

Masking is one of the most widely used countermeasures to protect implementations of block ciphers against side-channel analysis [CJRR99]. The idea is to split each sensitive intermediate value Z, into d shares $(Z_i)_{1 \leq i \leq d}$. The $d-1$ shares $Z_2, ..., Z_d$ are randomly chosen and the last one, Z_1 is processed such that::

$$Z_1 = Z * Z_2 * \cdots * Z_d \tag{42}$$

for a group operation $*$. Assuming the masks are uniformly distributed, the knowledge of $d-1$ shares does not tell anything about Z. However, partial knowledge on the d shares can be exploited to retrieve information on Z. That is why, to defeat masking, one should use a distinguisher able to combine the leakage of at least d samples of the traces (assuming masks do not leak at the same time). Higher-order correlation attacks [Mes00] exploit a combining function, $C : \mathbb{R}^d \rightarrow \mathbb{R}$, which transforms a multidimensional leakage into a single value such that the output of C correlates with Z. The optimal combining function is unknown but, the centered product between the shares [PRB09] is a popular choice.

3.1 MIA, a natural choice against masking

Although higher-order CPA attacks lead to successful key recoveries, they are not optimal from an information-theoretic point of view. Indeed, by the data processing inequality [BR12], the application of the combining function leads to an information loss. Opposed to Pearson's correlation, mutual information can deal with dependencies of multidimensional variables. Therefore, no combining function is required which makes MIA a very natural strategy against masked implementations. An extension of MIA in the context of masking has been proposed in [PR09] and [BGP$^+$11]. The idea is very similar to the non-masked case. Concepts of MIA versions 1 and 2 still apply and one can use $\mathcal{I}(f(Z_k), L)$ as a distinguisher.

3.2 About the partition function in the presence of masking

Using $\mathcal{I}(f(Z_k), L)$ as distinguisher still raises the question of the optimal f function. Th. 1 cannot be applied straightforwardly since, for masked implementation, the leakage cannot be expressed as a deterministic function $\varphi(Z_{k^*})$ modulo some noise. Instead, with Z_i representing the shares, one now has:

$$L = \sum_i \varphi_i(Z_i) \tag{43}$$

for some functions $\varphi_i : \mathcal{Z} \rightarrow \mathbb{R}^n$. Note that, as for the unmasked case, a noise-free version of the leakage is first considered to simplify the analysis. Noise will be added in section 3.3. Most of the time, the φ_i supports can be supposed disjoint (*i.e.* leakages of the shares do not overlap). In that case, the leakage vector could be summarized as:

$$L = [\varphi_1(Z_1), \dots, \varphi_d(Z_d)] \tag{44}$$

with φ_i taking its values in a subspace of $\mathbb{R}^n$. Even with this simplification, we could not solve analytically the problem of finding an optimal partition function, or, in other words, a function $f \in \mathcal{F}_{opt}$ as defined in (11). However, we still give some useful insights in the common case of Boolean masking on a device leaking the Hamming weight (or Hamming distance with a known value) of the shares.

For this specific case, [BGP$^+$11] tried to use the Hamming weight as well as the identity function for f (they were attacking the output of a DES S-box, therefore a non-injective intermediate variable). The Hamming weight produced better results. Their justification is that the Hamming weight is closer to the underlying leakage model of the circuit. We do not find this justification straightforward especially in a multivariate context since even in the ideal case where the leakage could be expressed as:

$$L = [\mathrm{HW}(Z_{k^*} \oplus M), \mathrm{HW}(M)] \tag{45}$$

$\mathrm{HW}(Z_{k^*})$ is not directly related to any physical leakage. More generally, there is no proof that if all shares leak with the same leakage model φ, taking $f = \varphi$ is the optimal (or even a good) option. However, in the specific case of a Hamming weight leakage model, [PRB09] has shown that their exists a linear correlation between $\mathrm{HW}(Z_{k^*})$ and the covariance: $\mathrm{cov}\big(\mathrm{HW}(Z_{k^*} \oplus M), \mathrm{HW}(M)\big)$ which is a clue that there exists a non-negligible mutual information between $\mathrm{HW}(Z_{k^*})$ and L. However, we go further in this paper by showing in Theorem 2 that there is actually no loss of information when applying the Hamming weight function to the Z_{k^*} variable. This result can then be used to give a formal justification for using $f = \mathrm{HW}$, as done hereafter.

Let us introduce $\mathcal{F}_{Left}$ as the left part of equation 11:

$$\mathcal{F}_{Left} = \underset{f\colon \mathcal{Z}\to\mathbb{R}^n}{\arg\max} \Big\{ \mathcal{I}\big(f(Z_{k^*}), L\big) \Big\} \tag{46}$$

This set does not consider the wrong hypotheses. Therefore it is not hard to find a function $f \in \mathcal{F}_{Left}$: the identity or any bijective function works. The problem is that with a bijective map, $\mathcal{I}\big(f(Z_{k^*}), L\big) = \mathcal{I}\big(f(Z_k), L\big)$ for any k. However, a non-injective function f such that $f \in \mathcal{F}_{Left}$ would naturally decrease $\mathcal{I}\big(f(Z_k), L\big)$ and create some distinguishability. Such a function is not *a priori* likely to exist. But the following theorem shows that, while being highly non-injective, $\mathrm{HW} \in \mathcal{F}_{Left}$.

Theorem 2. *Let L represent the leakage of a masked variable Z_{k^*} with a mask M. Let both shares follow any bijection b_1 and b_2 of a Hamming weight leakage model so that:*

$$L = \Big[b_1\big(HW(Z_{k^*} \oplus M)\big), b_2\big(HW(M)\big)\Big] \tag{47}$$

Then, $HW \in \mathcal{F}_{Left}$ or in other words: $\mathcal{I}\big(HW(Z_{k^*}), L\big) = \mathcal{I}\big(Z_{k^*}, L\big)$.

The following proof may be generalizable to higher-order (see section 5.3 for an empirical validation), but for simplicity, only first-order masking is considered here.

Proof. Since bijective transformations do not impact mutual information, one can consider without loss of generality that:

$$L = \big[\mathrm{HW}(Z_{k^*} \oplus M), \mathrm{HW}(M)\big] \tag{48}$$

Now let us evaluate $\mathcal{I}\big(f(Z_{k^*}), L\big)$ using equation 4:

$$\mathcal{I}\big(f(Z_{k^*}), L\big) = \sum_{\bar{f}\in f(\mathcal{Z})} \sum_{l\in\mathcal{L}} P(\bar{f}, l) \cdot log\left(\frac{P(\bar{f}, l)}{P(\bar{f}) \cdot P(l)}\right) \tag{49}$$

One can split the first sum by summing on z instead of $\bar{f}$:

$$\begin{aligned} \mathcal{I}\big(f(Z_{k^*}), L\big) &= \sum_{z\in\mathcal{Z}} \sum_{l\in\mathcal{L}} P(z, l) \cdot log\left(\frac{P(l \mid f(z))}{P(l)}\right) \\ &= \sum_{z\in\mathcal{Z}} \sum_{l\in\mathcal{L}} P(z) \cdot P(l \mid z) \cdot log\left(\frac{P(l \mid f(z))}{P(l)}\right) \end{aligned} \tag{50}$$

Since the identity function is bijective and maximizes this quantity, it would be enough to show that $P(l \mid \mathrm{HW}(z)) = P(l \mid z)$ for any given z and a given $l = [\mathrm{HW}(z \oplus m), \mathrm{HW}(m)]$ for a fixed m. Let us start by the latter term:

$$P(l \mid z) = P(\mathrm{HW}(m)) \cdot P(\mathrm{HW}(z \oplus m) \mid z,\, \mathrm{HW}(m)) \tag{51}$$

To compute the right term one can evaluate the cardinal of the set $\mathfrak{M}$ of all the masks m' satisfying the following conditions:

1) $\mathrm{HW}(m') = \mathrm{HW}(m)$

2) $\mathrm{HW}(z \oplus m') = \mathrm{HW}(z \oplus m)$

and divide by the number of byte with a Hamming Weight of $\mathrm{HW}(m)$ which is $\binom{8}{\mathrm{HW}(m)}$.

To evaluate this cardinal, we first show an invariance property. For any $m' \in \mathfrak{M}$, let $n_{m'}$ denotes the number of bits set to 1 in m' such that there is also a bit set to 1 at the same position (0 to 7) in z. Then:

$$\begin{aligned} \mathrm{HW}(z \oplus m') &= \mathrm{HW}(m') + \mathrm{HW}(z) - 2 \cdot n_{m'} \iff \\ n_{m'} &= \frac{\mathrm{HW}(m') + \mathrm{HW}(z) - \mathrm{HW}(z \oplus m')}{2} \end{aligned} \tag{52}$$

Now since m' satisfies the above two conditions:

$$n_{m'} = \frac{\mathrm{HW}(m) + \mathrm{HW}(z) - \mathrm{HW}(z \oplus m)}{2} \tag{53}$$

which does not depend on m' anymore. As $n_{m'}$ has to be a positive integer, the above equation shows that:

$$\mathrm{HW}(m) + \mathrm{HW}(z) - \mathrm{HW}(z \oplus m) \notin 2\mathbb{N} \implies \mathfrak{M} = \emptyset \tag{54}$$

This allows us to define a generic n as:

$$n = \begin{cases} \frac{\mathrm{HW}(m)+\mathrm{HW}(z)-\mathrm{HW}(z\oplus m)}{2}, \text{ if } \mathrm{HW}(m) + \mathrm{HW}(z) - \mathrm{HW}(z \oplus m) \in 2\mathbb{N} \\ -1, \text{ otherwise} \end{cases} \tag{55}$$

so that $\forall m' \in \mathfrak{M}, n_{m'} = n$.

Reciprocally, one can see that each byte m' such that $\mathrm{HW}(m') = \mathrm{HW}(m)$ and $n_{m'} = n$ is in $\mathfrak{M}$. So to form a valid $m' \in \mathfrak{M}$ one has to choose first the position of the n '1s' superposing with the '1s' in z, which lead to $\binom{\mathrm{HW}(z)}{n}$ possibilities. Then, choose the positions of the remaining '1s', which lead to $\binom{8-\mathrm{HW}(z)}{\mathrm{HW}(m)-n}$ possibilities. Therefore, with the convention $\binom{l}{k} = 0$ when k is strictly negative:

$$P(\mathrm{HW}(z \oplus m) \mid z \text{ and } \mathrm{HW}(m)) = \binom{\mathrm{HW}(z)}{n} \cdot \binom{8 - \mathrm{HW}(z)}{\mathrm{HW}(m) - n} \cdot \frac{1}{\binom{8}{\mathrm{HW}(m)}} \tag{56}$$

Injecting this into (51) gives:

$$\begin{aligned} P(l \mid z) &= \frac{\binom{8}{\mathrm{HW}(m)}}{2^8} \cdot \binom{\mathrm{HW}(z)}{n} \cdot \binom{8 - \mathrm{HW}(z)}{\mathrm{HW}(m) - n} \cdot \frac{1}{\binom{8}{\mathrm{HW}(m)}} \\ &= \frac{1}{2^8} \cdot \binom{\mathrm{HW}(z)}{n} \cdot \binom{8 - \mathrm{HW}(z)}{\mathrm{HW}(m) - n} \end{aligned} \tag{57}$$

Now let us evaluate $P(l \mid \mathrm{HW}(z))$:

$$P(l \mid \mathrm{HW}(z)) = P(\mathrm{HW}(m)) \cdot \overbrace{P(\mathrm{HW}(z \oplus m) \mid \mathrm{HW}(z) \text{ and } \mathrm{HW}(m))}^{A} \tag{58}$$

And,

$$A = \sum_{\substack{z' \text{ s.t.} \\ \mathrm{HW}(z')=\mathrm{HW}(z)}} P(z' \mid \mathrm{HW}(z)) \cdot P(\mathrm{HW}(z' \oplus m) \mid z' \text{ and } \mathrm{HW}(m)) \tag{59}$$

Now using result from (56):

$$A = \sum_{\substack{z' \text{ s.t.} \\ \mathrm{HW}(z')=\mathrm{HW}(z)}} \frac{1}{\binom{8}{\mathrm{HW}(z)}} \cdot \binom{\mathrm{HW}(z')}{n} \cdot \binom{8-\mathrm{HW}(z')}{\mathrm{HW}(m)-n} \cdot \frac{1}{\binom{8}{\mathrm{HW}(m)}}$$
$$= \binom{\mathrm{HW}(z)}{n} \cdot \binom{8-\mathrm{HW}(z)}{\mathrm{HW}(m)-n} \cdot \frac{1}{\binom{8}{\mathrm{HW}(m)}} \tag{60}$$

since all the terms are constant in the sum and there are exactly $\binom{8}{\mathrm{HW}(z)}$ of them. Now plugging this into (58) gives:

$$P(l \mid \mathrm{HW}(z)) = \frac{1}{2^8} \cdot \binom{\mathrm{HW}(z)}{n} \cdot \binom{8-\mathrm{HW}(z)}{\mathrm{HW}(m)-n} = P(l \mid z) \tag{61}$$

Thus,

$$\mathcal{I}\big(\mathrm{HW}(Z_{k^*}), L\big) = \mathcal{I}\big(Z_{k^*}, L\big) \tag{62}$$

which ensures that HW $\in \mathcal{F}_{left}$ and concludes the proof. □

Interpretation. This theorem shows that when the shares leak in Hamming weight, it is sound to use f = HW in practice because it creates some distinguishability by decreasing the information only for the wrong hypotheses. Since the Hamming distance with a computable value can be rewritten as a Hamming weight, it also works in that case. However, Th. 2 is not generalizable to any leakage model φ (for example on 3 bits words, $\varphi = 2b_1 + b_2 + b_3$ gives a counter-example). Knowing if there exists a generic strategy against masking (depending on φ but working for any φ) or if one will always be condemned to work on a case-by-case basis is an interesting question and may be handled in future works.

Remark 2. Note that since $\mathcal{I}\big(Z_{k^*}, L\big) = \mathcal{I}\big(\mathrm{HW}(Z_{k^*}), L\big) = \max_k[\mathcal{I}\big(\mathrm{HW}(Z_k), L\big)]$, the procedure described in section 2.6 can also be applied on a masked implementation, to test the validity of the Hamming weight leakage model hypothesis. If the Hamming weight is too far from the true model, a practical alternative is to use only specific bits of the unmasked variable as partition function. An example of this is given in section 6.

Considering the distinguishability score:

$$\mathcal{S}_f = \mathcal{I}\big(f(Z_{k^*}), L\big) - \max_{k \neq k^*}\left[\mathcal{I}\big(f(Z_k), L\big)\right] \tag{63}$$

HW has not been shown to be optimal. However, a partial result can be given introducing the concept of "wider" function.

Definition 1. A function f is said wider than g if there exists another function h such that: $h \circ f = g$.

Corollary 1. *Let L be defined as in (47). Then, for any function $\bar{h}$ wider than HW,* $\mathcal{S}_{HW} \geq \mathcal{S}_{\bar{h}}$.

Proof. The proof is given in appendix B. □

Even though we do not conjecture so, a function doing with a better distinguishability than the HW may exist. But a straightforward consequence of Th. 2, given by corollary 1, is that HW has a better or equal distinguishability score than any other wider function.

3.3 Noise and multidimensionality

The advantage of MINE is to be able to exploit the information contained in multiple samples at the same time. In a Hamming weight leakage scenario, the Hamming weight of a variable is probably not going to leak perfectly on a single sample. Instead, multiple samples may leak a noisy version of it. To ensure that it is sound to use MINE and its multidimensional capabilities to mount an attack in the case of masking, one would need a multidimensional version of Th.2. This is exactly the purpose of corollary 2, in which the noise is directly included.

In the context of masking the actual useful part of the leakage could be expressed as:

$$L = \big[b_1\big(\mathrm{HW}(Z_{k^*} \oplus M)\big) + N_1, \dots, b_{m_1}\big(\mathrm{HW}(Z_{k^*} \oplus M)\big) + N_{m_1}, \\ b'_1\big(\mathrm{HW}(M)\big) + N'_1, \dots, b'_{m_2}\big(\mathrm{HW}(M)\big) + N'_{m_2}\big] \tag{64}$$

with b_i and b'_j being bijective maps, and N_i and N'_j being discrete noise variables independent of the shares. The following corollary shows that Th 2 is still valid in that case.

Corollary 2. *Let L be defined as in (64). Then, one still has $HW \in \mathcal{F}_{Left}$ as defined in (46).*

Proof. As for Th. 2, one can drop, without loss of generality, the bijections in L as they do not affect the MI. Let N be the noise vector $[N_1, \dots, N_{m_1}, \bar{N}_1, \dots, \bar{N}_{m_2}]$ and $\bar{L}$ the noise-free version of the leakage so that $L = \bar{L} + N$. As for Th. 2, it is enough to show that $P(l \mid \mathrm{HW}(z)) = P(l \mid z)$ for any given l and z. Decomposing on all the possible values of the noise one has:

$$\begin{aligned} P(l \mid z) &= \sum_{n \in \mathcal{N}} P(n) \cdot P(L = l \mid z \text{ and } n) \\ &= \sum_{n \in \mathcal{N}} P(n) \cdot P(\bar{L} = l - n \mid z) \end{aligned} \tag{65}$$

Since $\bar{L}$ is noise free, it consists of the repetition of the same two variables: $\mathrm{HW}(Z_{k^*} \oplus M)$ (m_1 times) and $\mathrm{HW}(M)$ (m_2 times). So for the probability $P(\bar{L} = l - n \mid z)$ to be non-zero, the vector $l - n$ should be constant on its first m_1 coordinates, and constant on its m_2 last one. Let $\mathcal{N}_c$ be the subset of $\mathcal{N}$ verifying the precedent property. If $n \notin \mathcal{N}_c$, then:

$$P(\bar{L} = l - n \mid z) = P(\bar{L} = l - n \mid \mathrm{HW}(z)) = 0 \tag{66}$$

Else, if $n \in \mathcal{N}_c$, then, with $a_n = (l - n)[1]$, $b_n = (l - n)[m_1 + m_2]$ and $\tilde{L} = [\mathrm{HW}(Z_{k^*} \oplus M), \mathrm{HW}(M)]$:

$$P(\bar{L} = l - n \mid z) = P(\tilde{L} = [a_n, b_n] \mid z) \tag{67}$$

So (65) can be rewritten as:

$$P(l \mid z) = \sum_{n \in \mathcal{N}_c} P(n) \cdot P(\tilde{L} = [a_n, b_n] \mid z) \tag{68}$$

Since, Th. 2 tells that $P(\tilde{L} = [a_n, b_n] \mid z) = P(\tilde{L} = [a_n, b_n] \mid \mathrm{HW}(z))$:

$$\begin{aligned} P(l \mid z) &= \sum_{n \in \mathcal{N}_c} P(n) \cdot P(\tilde{L} = [a_n, b_n] \mid \mathrm{HW}(z)) \\ P(l \mid z) &= \sum_{n \in \mathcal{N}_c} P(n) \cdot P(\bar{L} = l - n \mid \mathrm{HW}(z)) \\ P(l \mid z) &= P(l \mid \mathrm{HW}(z)) \end{aligned} \tag{69}$$

which concludes the proof. □

This corollary shows that it is sound to use $\mathcal{I}(\mathrm{HW}(Z_k), L)$ as distinguisher even when considering a noisy multidimensional leakage vector. Th. 2 still applies and MINE may benefit from the different leakage sources resulting in an attack (presented in the next section) exploiting more of the available information.

4 Neural Estimated Mutual Information Analysis (NEMIA)

This section aims at formally describing the new attack proposed in this paper. Note that throughout this work, a tool able to compute $\mathcal{I}(Z, L)$ with high dimensional variables has been assumed to exist. This research has been driven by recent progress regarding neural estimation techniques. However, this work is not absolutely related to MINE. It would stay sound with any MI estimator able to work in high dimension. In particular, any progress in the field, which is likely to happen since it is a very active domain, would instantly impact the attack efficiency. In this work, the most basic version of MINE is used. It should be seen as a proof of concept with almost no hyper-parameters tuning and without considering recent optimizations nor improvements in the technique (non-exhaustively: [CL20, LSN+19, CABH+19]). A study focused on deep learning optimizations would be interesting but is out of the scope of this paper. Basic principles of MINE are recalled hereafter.

4.1 Mutual Information Neural Estimation

Technical details about the utilization of MINE in a side-channel context can be found in [CLM20]. However, a high-level picture is still given in this section. The general idea is to express $\mathcal{I}(Z, L)$ as the Kullback-Leibler divergence between the joint distribution and the product of the marginals: $\mathcal{I}(Z, L) = D_{KL}(p_{Z,L} \,||\, p_Z \otimes p_L)$. Then, to exploit the Donkser-Varadhan variational formulation of the KL-divergence that states that if p and q are two densities defined over a compact set $\Omega \in \mathbb{R}^d$:

$$D_{KL}(p \,||\, q) = \sup_{T:\,\Omega \to \mathbb{R}} [\mathbb{E}_p[T] - log(\mathbb{E}_q[e^T])] \tag{70}$$

This allows to express MI as a supremum. Then, the following loss function can be defined:

$$\mathcal{L}(\theta) = \mathbb{E}_{p_{Z,L}}[T_\theta] - log(\mathbb{E}_{p_Z \otimes p_L)}[e^{T_\theta}]) \tag{71}$$

and deep learning techniques can be applied to maximize this loss over all the functions T_θ parametrized by a neural network with parameters $\theta \in \Theta$. The objective function should converge towards the supremum so that its final value constitutes the MI estimation. Formally:

Definition 2. (MINE) Let $\mathcal{A} = \{(z_1, l_1), \dots, (z_n, l_n)\}$ and $\mathcal{B} = \{(\widetilde{z}_1, \widetilde{l}_1), \dots, (\widetilde{z}_n, \widetilde{l}_n)\}$ be two sets of n empirical samples respectively from $p_{Z,L}$ and $p_Z \otimes p_L$. Let $\mathcal{F} = \{T_\theta\}_{\theta \in \Theta}$ be the set of functions parametrized by a neural network. MINE is defined as follows:

$$\widehat{\mathcal{I}(S, X)}_n = \sup_{T \in \mathcal{F}} \overline{\mathbb{E}_{\mathcal{A}}[T]} - log(\overline{\mathbb{E}_{\mathcal{B}}[e^T]}) \tag{72}$$

where $\overline{\mathbb{E}_{\mathcal{X}}[\cdot]}$ stands for the expectation empirically estimated over the set $\mathcal{X}$.

In practice one only has samples from the joint distribution: $\mathcal{A} = \{(z_1, l_1), \dots (z_n, l_n)\}$ of the labeled traces. Samples from the product of the marginals can be artificially generated by shuffling the variable L using a random permutation ρ: $\mathcal{B} = \{(z_1, l_{\rho(1)}), \dots, (z_n, l_{\rho(n)})\}$.

Validation loss function. One of the main problems of MINE pointed out in [CLM20] is the overfitting. Indeed, the loss function may overestimate the true MI. Therefore, one can introduce a validation loss function to detect overfitting and to produce a more reliable estimation. The idea is to split $\mathcal{A}$ and $\mathcal{B}$ into training datasets $\mathcal{A}_t$ and $\mathcal{B}_t$ and validation datasets $\mathcal{A}_v$ and $\mathcal{B}_v$. Then, only the training datasets are used for back-propagation so that the loss function evaluated on the validation datasets cannot overestimate the MI. That is why only validation loss functions are considered/plotted in this paper. For robustness, the MI estimation is not set to be the supremum of the validation loss, but instead, the supremum of a moving average along the epochs with a window size of w which depends on the variability between epochs ($w = 10$ in this paper).

Architecture. The network's input layer consists of a concatenation of both Z and L variables. Authors in [CLM20] have shown that the representation of Z is important and that one should use the One-Hot Encoding (OHE) or a binary encoding of Z (unless otherwise specified we used the OHE in this paper). The output layer is a single neuron as the function T output has to be a real value. Other layers are not specified and should be adapted to the underlying problem (*e.g.* convolutional layers to counter jitter or traces misalignment).

For our experiments, we used a Convolutional Neural Network (CNN) where a batch normalization layer is added after the first layer and dropout layers are inserted after each hidden layer in order to mitigate overfitting. The activation function is set to the Exponential Linear Unit (ELU) and the batch size to 1000. The precise architecture is depicted in Appendix D. The validation dataset represents 20 percent of the full dataset.

4.2 Multidimensional paradigm

MINE is by essence a tool that estimates MI in a multidimensional way, enabling to compute the MI between $f(Z_k)$ and significant part of the traces. This was not possible with classical MI estimators which do not scale with high dimensional variables. Until now, MIA was only performed with the following distinguisher:

$$\mathcal{D}_{old}(k) = \max_{i} \mathcal{I}(f(Z_k), L[i]) \tag{73}$$

where $L[i]$ represents the i-th sample of the trace. This way, trace dimension is kept low, allowing methods such as the histogram or the kernel density estimation [PR09] to produce reliable results. However, this comes at the cost of sacrificing some, and maybe a large part, of the available information. MINE allows to directly use:

$$\mathcal{D}_{new}(k) = \mathcal{I}(f(Z_k), L) \tag{74}$$

as a distinguisher. This comes with two main advantages:

- Intermediate variables often leak at multiple instants in the trace. MINE allows to exploit all these leakage sources at the same time.
- Other intermediate variables, statistically dependent from the first one, can also leak information. For example, there could be some useful information about an AES key, before and after the application of the first S-box. In this context, MINE could exploit leakage from both intermediate variables at the same time, without any assumption related to the kind of link between these variables.

Theorem 1 states that the optimal distinguisher is $\mathcal{I}(\varphi(Z_k), L)$ with φ being the leakage model. It is important to note that $\varphi(Z_k)$ itself can be multidimensional. Therefore, an optimal MI attack should exploit this multidimensionality of the leakage model to increase the distinguishability of the correct hypothesis. However, it is frequent that multiple samples leak with the same underlying model: for example, a noisy version of the Hamming weight of $Sbox[k^* \oplus P]$ can leak multiple times in the trace. In such a context, the deterministic parts of the leakage of all these samples are all bijectively related. As adding bijection of the same variables multiple times would not change the MI, one can keep only one version of each different sub-leakage model. For example, if the target leaks (maybe multiple times) the Hamming weight of the first S-box of an AES and the Hamming distance between the S-box and $k \oplus P$, Z_k could be defined as $k \oplus P$ and one could replace $\varphi(Z_k)$ by the two-dimensional vector:

$$\left[\mathrm{HW}\big(Sbox[Z_k]\big), \mathrm{HW}\big(Sbox[Z_k] \oplus Z_k\big)\right] \tag{75}$$

Remark 3. In practice, one may deliberately drop some intermediates variables for not being enough discriminating for wrong key candidates making them less tolerant regarding errors in the estimation of φ. For example, it is theoretically possible to use leakage on a xor: $\mathrm{HW}(k \oplus P)$ (assuming a Hamming weight *a priori*) but it is preferable to use intermediate variables where each bit depends on multiple bits of k such as the output of an S-box. Indeed, these variables are more discriminating since single bit errors on k are diffused to the whole variable which prevents from rewarding wrong hypotheses with several correct bits.

Scalability with masking order. In the context of masking, another advantage of multidimensionality emerges. In a classical d-order attack one often does not know the exact leakage time of each share, and therefore, has to compute the value of the distinguisher for each possible tuple $(i_1, \ldots, i_d)$ and select the maximum. In the case of MIA the old distinguisher takes the following form:

$$\mathcal{D}_{old}(k) = \max_{i_1,\ldots,i_d} \{\mathcal{I}(f(Z_k), L[i_1, \ldots, i_d])\} \tag{76}$$

For long traces, this can become a huge constraint since the total number of tuples grows exponentially with the masking order. Our version of the MIA which uses $\mathcal{I}(f(Z_k), L)$ as distinguisher, does not suffer from this since it does not require any kind of recombination between time samples. Note that it does not mean that masking is useless: it still decreases exponentially the information contains in side-channel traces [PR13] and an attack may require exponentially more traces to succeed.

For a fixed number of traces, the number of network trainings to mount a NEMIA is constant with respect to the masking order. However, each training may require more epochs to succeed when dealing with higher order masking schemes, in order to escape from the so called *plateau* effect described in [MCLS22]. The computational complexity required by gradient descent-based algorithms to escape from such a *plateau* (and start being better than random models) is an open problem but figure 9-b of [MCLS22] suggests that the number of epochs compared to the masking order is sub-exponential.

4.3 Attack description

A step-by-step description of the NEMIA is given hereafter. It takes as input a set of traces and outputs a ranking of the key hypotheses.

1. Define an *a priori* $\bar{\varphi}$ on the leakage model. It can be multidimensional if multiple intermediate variables related to the key leak information. Also, a single intermediate variable can have different leakage models at different times. The test described in section 2.6 can be used to detect wrong *a priori*. Even if MIA is tolerant regarding estimation errors on φ, better *a priori* lead to more efficient attacks.

2. Compute, for all k, the hypothesis vectors: $H_k = \bar{\varphi}(Z_k)$.

3. Compute $\mathcal{I}(H_k, L)$, for all k, with MINE. This implies to run a neural network trainings for each key hypothesis. Each estimation is the supremum of a moving average along the epochs of the validation loss function.

4. Rank the key hypotheses.

For masked implementation, the only step that changes is the construction of H_k. If the shares have a Hamming weight leakage model, Th.2 proves that it is sound to use the Hamming weight of the corresponding unmasked intermediate variable in H_k (one may do this for multiple intermediate variables). For a generic leakage model of the shares, the best strategy to adopt remains an open question. It appears that, in some cases, it is efficient to keep a restrictive number of bits of the unmasked variable as partition function, for example in a situation where some bits of the shares leak much more information than the others (an example of this is given in section 6).

5 Simulation experiments

In order to gain confidence in the mathematical results presented in this paper, as well as to gain intuition about their implications, this section presents experiments on synthetic data.

5.1 The importance of the *a priori*

The main message of Th.1 is that, to maximize the distinguishability of the correct hypothesis, one should use the leakage model φ to create the hypothesis vectors H_k. In a classical side-channel scenario, with no other specific information, one may often guess a Hamming weight leakage of the intermediate variables. This is justified by electronic arguments. However, it has been shown that bits may have different leakage behaviours, such as leakage weighting or even sign inversions [CLH19]. To illustrate Th.1, 10k synthetic traces leaking a slightly modified version φ_0 of the Hamming weight have been generated. They consist of a single sample leaking the Hamming weight of $Z_{k^*} = Sbox(k^* \oplus P)$ but with a flipped sign for bit 0 so that:

$$\varphi_0(z) = -z_0 + \sum_{i=1}^{7} z_i \tag{77}$$

with z_i representing the i-th bit of z. Some Gaussian noise has been added to the traces so that $L = \varphi_0(Z) + \mathcal{N}(0, 1)$. Fig.1 shows the results of a NEMIA with $k^* = 0$, both with HW and φ_0 as partition function. As predicted by Th.1, the distinguishability score:

$$\mathcal{S}_f = \mathcal{I}\big(f(Z_{k^*}), L\big) - \max_{k \neq k^*} \left[\mathcal{I}\big(f(Z_k), L\big)\right] \tag{78}$$

is higher for $f = \varphi_0$ than for $f =$ HW. Obviously, an attacker may not know φ_0 and an attack with the Hamming weight still succeeds in that case. However, this shows that, if by any means, an adversary knows the particularity of bit 0 of such a target, he can perform more efficient attacks.

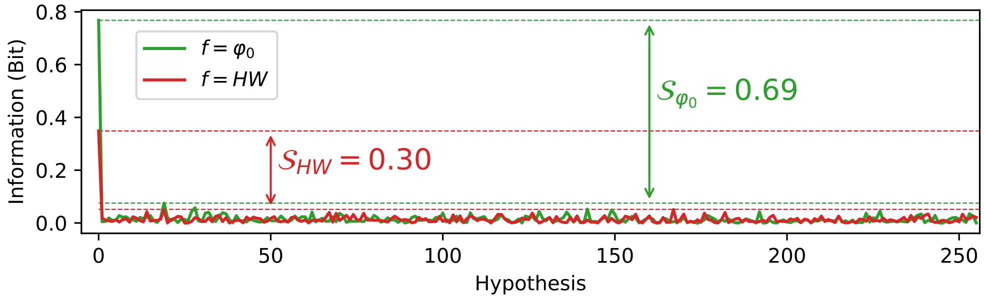


Figure 1: $\mathcal{I}\big(f(Z_k), L\big)$ in terms of k, with $k^* = 0$

Semi-supervised attacks. This opens the idea of semi-supervised attacks. One of the main problems of profiling attacks is the portability [EG12]. Indeed, during the characterization phase, the adversary learns a perfect representation of the leakage model which may overfit on the particular target which is profiled. It has been shown that portability to other targets is not trivial. Therefore NEMIA could be turned into a semi-supervised attack where the purpose of the characterization phase is only to learn general leakage characteristics, such as the sign or weighting of each bit, and use them as an improved *a priori* for a NEMIA. Since NEMIA is agnostic towards bijective errors in the leakage model estimation, it has a better chance of being portable on many other targets similar to the one used for profiling.

5.2 The potential of multidimensionality

One of the main advantages of NEMIA is its potential to exploit at the same time, multiple leakage sources. It is possible that multiple intermediate variables leak information on the key and each particular variable may leak multiple times in the traces. This section aims at showing how NEMIA could exploit all these leakage sources as well as to compare it with other state of the art attacks.

Traces Generation. To this aim, a dataset of 100k synthetic traces have again been generated. These traces represent the leakage of an AES that both leaks $A_{k^*} = \mathrm{HW}(Sbox[k^* \oplus P])$ and $B_{k^*} = \mathrm{HW}(Sbox[k^* \oplus P] \oplus (k^* \oplus P))$. One could imagine that the bus leaks the Hamming weight of the S-box data and that the update of the state register leaks the Hamming distance with its precedent value (*e.g.* [MEP+08]).

One of the strength of using deep learning in an unsupervised attack is the absence of need for preprocessing techniques. To highlight this fact we also added 90 % of uninformative samples as well as some misalignment in the traces following the shifting deformation procedure introduced in [CDP17] which simulates a random delay effect of maximal amplitude T by shifting each trace by a random number uniformly drawn between 0 and T. The procedure for the trace generation is depicted in Algorithm 1.

Algorithm 1 Generate Traces

Output: L**, a (100k, 1010) array**
Output: P**, a (100k) array**

$P \leftarrow$ Draw 100k plaintexts uniformly from $[\![0, 255]\!]$
$A \leftarrow \mathrm{HW}(\mathtt{Sbox}[P \oplus k^*])$
$B \leftarrow \mathrm{HW}(\mathtt{Sbox}[k^* \oplus P] \oplus (k^* \oplus P))$
$S \leftarrow$ Draw 1010 samples from a Gaussian $\mathcal{N}(0, 10^2)$ ▷ Generate a baseline shape
$L \leftarrow$ Repeat S 100k times to form a (100k, 1010) array
for $1 \leq i \leq$ 100k **do**
  for $1 \leq j \leq 50$ **do** ▷ Add leakage one every 10 samples
    $L[i, 10 * j] \leftarrow L[i, 10 * j] + A[i]$
    $L[i, 10 * j + 500] \leftarrow L[i, 10 * j + 500] + B[i]$
  end for
end for
$R \leftarrow$ Draw an array (100k, 1010) of random number from a Gaussian $\mathcal{N}(0, 20^2)$
$L \leftarrow L + R$ ▷ Add some noise
for $1 \leq i \leq$ 100k **do**
  $sh \leftarrow$ Draw a random integer uniformly from $[\![0, 10]\!]$
  $L[i] \leftarrow \mathrm{Roll}(L[i], sh)$ ▷ Apply the jitter (Roll shift the array by sh)
end for
return L, P

Compared strategies. We used the generated dataset to compute and compare guessing entropies for the following attack strategies:

1. A classical CPA [BCO04] with a Hamming weight model. The score for each hypothesis is defined as the maximum score along the sample axis.

2. A classical univariate MIA with a Hamming weight model computing the MI with the histogram method described in [BGP+11] with 9 bins. Again, the score for each hypothesis is defined as the maximum score along the sample axis.

3. NEMIA$_{Partial}$, only considering the Hamming weight leakage (A_k) to construct the hypothesis vectors $H_k = A_k$:

4. NEMIA$_{Full}$, considering both leakages (A_k and B_k) to construct the hypothesis vectors $H_k = [A_k, B_k]$.

5. The Differential Deep Learning Analysis (DDLA) introduced in [Tim19]. It is sound to compare NEMIA to DDLA since both methods use deep learning with an unsupervised approach. It builds 256 classifiers, one for each key hypothesis, and uses a metric (we used the accuracy as suggested in [Tim19]) as a distinguisher. Note that a partition function also has to be applied to the intermediate variables but its optimal choice has not been discussed in [Tim19]. We use the Hamming weight function in this experiment.

6. A classical deep learning supervised attack [MPP16], denoted DL-supervised, where a network is train to classify amoung the 256 classes. The total number of traces is divided into 80% for training and 20% for the actual attack. The architechture of the network is depicted in Appendix D.

7. The same deep learning attack but in a non-limited setup regarding the number of traces during profiling. In practice we have trained the network using another dataset of 100k traces generated with Algorithm 1. This attack is denoted DL-supervised$_\infty$.

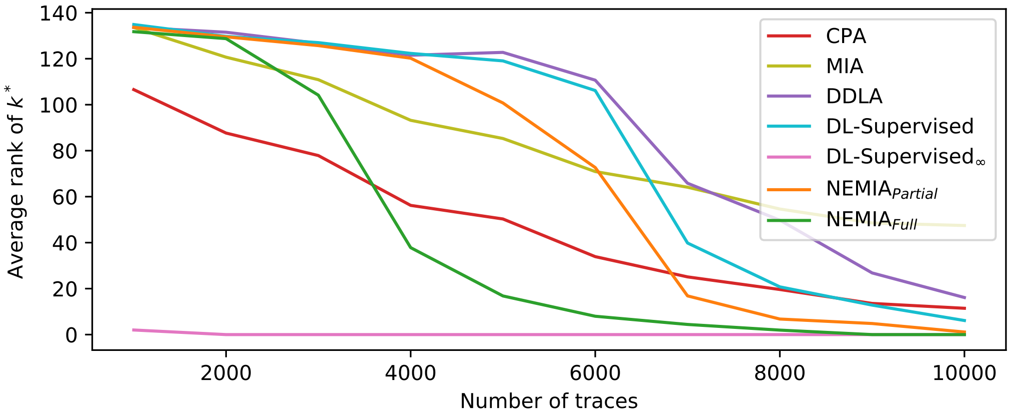


Figure 2: Guessing entropies for the considered attacks

Figure 2 shows the evolution of the average rank of k^* for each attack. Each point represents the average over 100 attacks computed with traces randomly drawn from the 100k traces dataset. It appears that for low numbers of traces, CPA performs the best amoung the unsupervised attacks but this is not very meaningful since attacks with such guessing entropies (greater than 20 on a single key byte) are not really exploitable for a full key recovery. Deep learning attacks behave more like if they had a threshold: after a certain number of traces, one can observe a quick drop in their guessing entropies.

As predicted by the theory, $NEMIA_{Full}$ converges faster towards a ranking of 0 than $NEMIA_{Partial}$, and both converge faster than CPA. $NEMIA_{Partial}$ outperforms DDLA and also the supervised DL attack with a restricted number of traces for profiling. This may seem counter-intuitive but in this case we argue that the learning problem is simpler for NEMIA since it has to deal with 9 different classes instead of 256 for the DL model. This may result into succesfull profiling with less traces. In this case, the application of the partition function is only beneficial and does not induce information loss since the true leakage model is known.

To the best of our knowledge, classical MI-based attacks always performed worse than CPA in the literature, when considering the Hamming weight model, which is again confirmed by our results. This experiment shows that in a low-information scenario (noisy traces with jitter), NEMIA may be worth considering among the other unsupervised attacks.

5.3 Empirical validation of theorem 2

Th.2 may seem very counterintuitive since it basically says that: when shares of a Boolean masking leak in a Hamming weight model, one has:

$$\mathcal{I}\big(\mathrm{HW}(Z_{k^*}), L\big) = \mathcal{I}\big(Z_{k^*}, L\big) \tag{79}$$

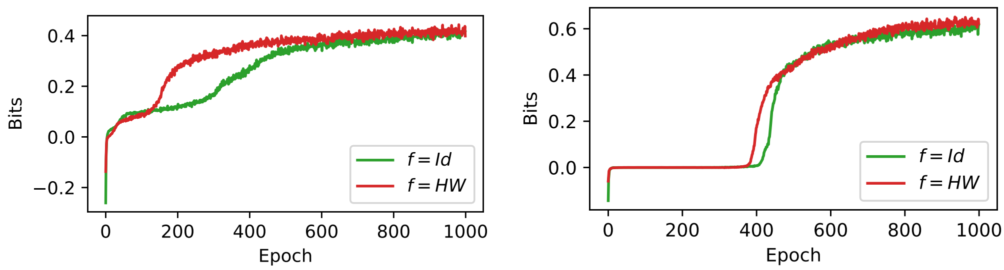
which is surprising since HW is highly non-injective and should at first glance, decrease the information. Corollary 2 says that this is even true when multiple samples leak a noised version of the Hamming weight of the shares. To verify this claim, 100k synthetic traces have been generated considering the following leakage:

$$\begin{aligned} L = \big[&\mathrm{HW}(Z_{k^*} \oplus M) + N_1, \ldots, \mathrm{HW}(Z_{k^*} \oplus M) + N_{10}, \\ &\mathrm{HW}(M) + N_{11}, \ldots, \mathrm{HW}(M) + N_{20}\big] \end{aligned} \tag{80}$$

with $Z_{k*} = Sbox(k^* \oplus P)$, $N_i = \mathcal{N}(0,1)$ and M being uniformly distributed in $\mathbb{Z}/256\mathbb{Z}$.

Fig. 3a shows the evolution of the loss function for both the HW and the identity function for the correct key hypothesis. As predicted, both converge towards the same value which confirms experimentally that the application of the HW does not alter information. The HW function is even doing a little better which can be explained with practical machine learning considerations. Indeed, the information being constant, it is easier for the network to learn with a 9-classes variable than with a 256 classes variable (note that in this experiment, $id(Z_{k^*})$ has been encoded in binary rather than in OHE, because it produced slightly better results). Also, since overfitting was not really a problem in this experiment, the dropout parameter has been set to $p = 0.1$.

Fig. 3b shows the result of the same experiment performed on a second-order masking, with three shares and 10 leakage samples for each. Noise has been a bit decreased ($\sigma = 0.5$ instead of 1) to keep comparable level of information. The result sustains that Th.2 may be generalized to higher-order and that MINE is able to extract information even with a second-order masking.



(a) First order masking ($\sigma = 1$) (b) Second order masking ($\sigma = 0.5$)

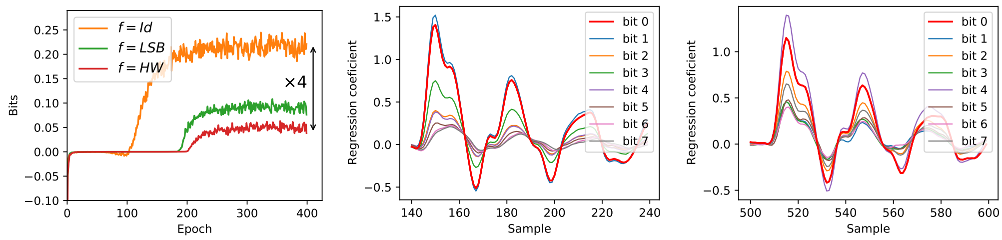
Figure 3: Comparison of $\mathcal{I}\big(Z_{k^*}, L\big)$ and $\mathcal{I}\big(\mathrm{HW}(Z_{k^*}), L\big)$ on masked synthetic traces

6 A practical case: attack on ASCAD

This section provides a real case experiment on the public dataset of ASCAD [BPS+18]. We only considered the training dataset composed of 50k traces composed of 700 samples focusing on the processing of the third byte (the first two are not masked) of the masked state $Sbox(k^*[3] \oplus P[3]) \oplus r[3]$, with r being the mask variable and with a fixed key $k^*[3]$.

Since it is a masked implementation, the test described in remark 2 has first been conducted. Results are presented in Fig. 4a. $\mathcal{I}\big(Z_{k^*}, L\big)$ is more than four times greater than $\mathcal{I}\big(\mathrm{HW}(Z_{k^*}), L\big)$ which indicates that the underlying leakage of the shares is far from a pure Hamming weight model. In parallel to this, authors in [Tim19] applied the DDLA strategy which also requires a partition function and they reported that, for the ASCAD database, only keeping the value of the Least Significant Bit (LSB) produced better results than the Hamming weight without giving further explanations.

In a real attack scenario, an adversary mounting a NEMIA could obviously try to use every single bit of the unmasked variable as partition function. But in order to gain some intuition, and since the masks values are given in the database, we first performed a linear regression on both shares, assuming bits leak independently so that the actual leakage of share s is: $\sum_{i=0}^{7} \alpha_i s_i + \beta$. Figs. 4b and 4c show the evolution of the α_i coefficients, on a leakage window for both shares. Since the implementation is protected by a Boolean masking, a mono-bit leakage is exploitable only if it is present on the same bit of both shares. Out of the 8 bits, bit 0 (LSB) is clearly the one that leaks the most information since its coefficients are among the greatest ones in both shares. Thus, we computed with MINE $\mathcal{I}(Z_{k^*}[0], L)$ where $Z_{k^*}[0]$ represents the LSB of $Sbox(k^*[3] \oplus P[3])$. It returned 0.09 bit, which is two times more than the information left with the Hamming weight (see Fig. 4a). This indicates that the LSB may be a good partition function since it is highly non-injective and still keep a decent amount of information for the correct hypothesis. We also tried with other bits but the information, while being non-zero, was significantly lower.



(a) $\mathcal{I}(f(Z_{k^*}), L)$ (b) $r[3]$ (c) $Sbox(k^*[3] \oplus P[3]) \oplus r[3]$

Figure 4: Analysis of the ASCAD leakage model:
a) Test from remark 2 - b) & c) Coeficients of a linear regression on the given variable

Even though attacks with the Hamming weight were successful, we decided to use the LSB as partition function for the rest of our analysis. The attacks presented in this section uses the whole 700 samples as input. We compared the following attacks:

1. A classical second-order CPA [PRB09] with a Hamming weight model. For each key hypothesis, the CPA is performed on each possible combination of two samples, the maximum being retained as the score.
2. A second-order MIA with a LSB model computing the MI with the histogram method described in [BGP+11] with 9 bins. Again, for each key hypothesis, the MIA is performed on each possible combination of two samples, the maximum being retained as the score.
3. NEMIA with LSB as a partition function. The architechture of the network is depicted in Appendix D.
4. The Differential Deep Learning Analysis (DDLA) using the accuracy as distinguisher and with LSB as partition function. The architechture of the network is depicted in Appendix D.
5. A deep learning supervised attack [MPP16], denoted DL-supervised, where a network is train to classify among the 256 classes (we do not apply any partition functions because it is not required in a supervised context). The total number of traces is divided into 80% for training and 20% for the actual attack. The architechture of the network is depicted in Appendix D.

Results. In order to evaluate the potential of NEMIA to exploit leakage even in very low information context, the dataset has been artificially degraded adding Gaussian noise $\mathcal{N}(0, \sigma^2)$ to each sample. All the attacks have been performed with σ going from 0 to 20, using the whole 50k traces. For each level of noise, the attacks have been repeated 10 times (with different random sampling of the noise) in order to compute the average rank of the correct hypothesis. Results are presented in Figure 5. They confirm that NEMIA is able to succeed in situations where the considered state of the art attacks would not.

As for the experiment in Subsection 5.2, the DL-Supervised attack performs worse than the unsupervised attack which is non-intuitive. However, an adversary performing a supervised attack would likely have an unlimited amount of traces for profiling which will give rise to the best attack in terms of attack traces. We lack traces to compute the equivalent of DL-Supervised$_\infty$ for such noise level. It appears that the application of the partition function (the LSB which only has two classes) makes the training easier for the networks which explain why a DL model, with a restricted number of traces for profiling, underperforms compared to the supervised attacks. Obvioulsy the partition function could be applied even in the supervised case (*i.e.* building a two classes classifier) but one would then loose the interest of being in a supervised context where no assumption has to be done on the leakage model.

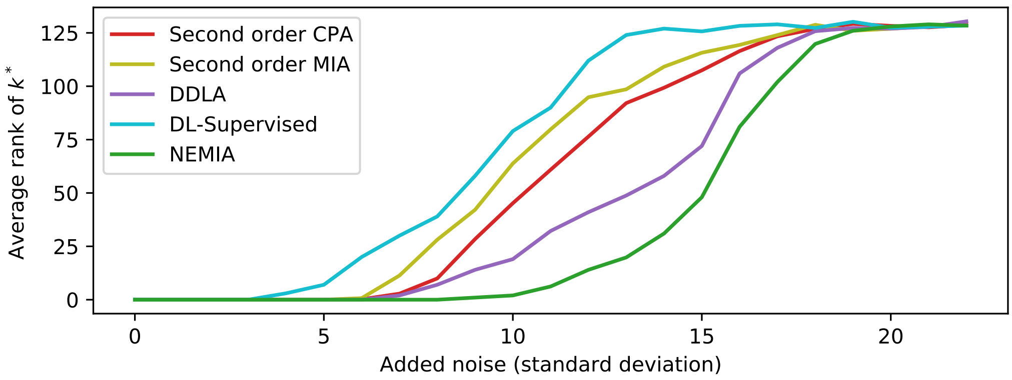


Figure 5: Guessing entropies for the considered attacks on ASCAD with added noise

6.1 Complexity

One of the limitations of non-profiled deep learning based attack is that they require to train a neural network for each key hypotheses. That is why 256 trainings were necessary to run NEMIA and DDLA. Both attacks had almost the same time complexity since we used essentially the same network architecture (Appendix D) and stopped training after 50 epochs in both cases, for all values of added noise. To give an order of magnitude, running the full NEMIA (or DDLA) on the ASCAD dataset ($50k$ traces, 700 samples) required approximately 2 hours and a half on a personal computer with 128 GB or RAM, a Tesla V100 GPU, and 2 Intel Xeon gold 5218R 2.1GHz with 20 cores each. In lower information context, requiring to train networks with much more traces, the complexity of such attacks may become a serious limitations. However, in such cases, the recombination of samples required by more conventional higher-order attacks may also be overwhelming. If the leakage area of the shares can be reduced to small part of the traces there may be a trade-off between the required number of traces and the time complexity of using NEMIA compared to a classical higher-order attack. Such a trade-off would depend on the the nature of the leakage and especially on its multivariate aspect.

7 Conclusion and perspectives

This paper first proposes a clarification of the state of the art around the MIA. It provides rigorous proofs whose goal is to derive the optimal MI-based attack working with high-dimensional traces. Combined with recent breakthroughs on neural MI estimation techniques, this allows to mount a new attack: the NEMIA, which benefits from both the strength of deep learning and information theory. Being able to exploit at the same time multiple leakage sources, it pushes the amount of effectively used information (depending on the strength of the attacker *a priori*) closer to the actual existing information between traces and secret. Simulations and real case experiments are presented to support the mathematical theory developed in this paper. They also show that NEMIA outperforms classical uni/bi-variate side-channel attacks and that this strategy may be worth to consider in low-information/high-noise situations, where all (or a large part of) the available information contained in traces need to be used to mount a successful attack.

Several lines of research emerge from this paper. The mathematical analysis could be further extended, especially in the context of masking, in order to develop strategies for generic leakage model of the shares or for other masking schemes such as arithmetic masking. On the practical side, integrating the latest optimization on neural estimation techniques, as well as deep learning research on optimal networks architecture and hyper-parameters would allow to mount more efficient attacks, taking as input larger portion of the traces, leading to better/easier attacks.

References

[BBR+18] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: Mutual information neural estimation, 2018.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[BGP+11] Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual information analysis: A comprehensive study. *J. Cryptol.*, 24(2):269–291, April 2011.

[BPS+18] Ryad Benadjila, Emmanuel Prouff, Rémi Strullu, Eleonora Cagli, and Cécile Dumas. Study of deep learning techniques for side-channel analysis and introduction to ascad database. *ANSSI, France & CEA, LETI, MINATEC Campus, France.*, 2018.

[BR12] Normand J. Beaudry and Renato Renner. An intuitive proof of the data processing inequality, 2012.

[CABH+19] Chung Chan, Ali Al-Bashabsheh, Hing Pang Huang, Michael Lim, Da Sun Handason Tam, and Chao Zhao. Neural entropic estimation: A faster path to mutual information estimation, 2019.

[CDP17] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems – CHES 2017*, pages 45–68, Cham, 2017. Springer International Publishing.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, pages 398–412, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[CL20] Kwanghee Choi and Siyeong Lee. Regularized mutual information neural estimation, 2020.

[CLH19] Valence Cristiani, Maxime Lecomte, and Thomas Hiscock. A Bit-Level Approach to Side Channel Based Disassembling. In *CARDIS 2019*, Prague, Czech Republic, November 2019.

[CLM20] Valence Cristiani, Maxime Lecomte, and Philippe Maurine. Leakage Assessment through Neural Estimation of the Mutual Information. In *International Conference on Applied Cryptography and Network Security (ACNS)*, volume 12418 of *Lecture Notes in Computer Science*, pages 144–162, Rome, Italy, October 2020.

[DPRS12] Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. Univariate side channel attacks and leakage modeling. *Journal of Cryptographic Engineering*, 1:123–144, 04 2012.

[EG12] M. Abdelaziz Elaabid and Sylvain Guilley. Portability of templates. *Journal of Cryptographic Engineering*, 2:63–74, 2012.

[GBTP08] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2008*. Springer Berlin Heidelberg, 2008.

[KB14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Annual International Cryptology Conference*, 1999.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[LSN+19] Xiao Lin, Indranil Sur, Samuel A. Nastase, Ajay Divakaran, Uri Hasson, and Mohamed R. Amer. Data-efficient mutual information neural estimator, 2019.

[MCLS22] Loïc Masure, Valence Cristiani, Maxime Lecomte, and François-Xavier Standaert. Don't learn what you already know: Scheme-aware modeling for profiling side-channel analysis against masking. Cryptology ePrint Archive, Paper 2022/493, 2022. https://eprint.iacr.org/2022/493.

[MDP19] Loïc Masure, Cécile Dumas, and Emmanuel Prouff. A comprehensive study of deep learning for side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020, 2019.

[MEP+08] Amir Moradi, Thomas Eisenbarth, Axel Poschmann, Carsten Rolfes, Christof Paar, M.T. Manzuri, and Mahmoud Salmasizadeh. Information leakage of flip-flops in dpa-resistant logic styles. *IACR Cryptology ePrint Archive*, 2008:188, 01 2008.

[Mes00] Thomas S. Messerges. Using second-order power analysis to attack dpa resistant software. In Çetin K. Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, pages 238–251, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[MPP16] Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. Breaking cryptographic implementations using deep learning techniques. In Claude Carlet, M. Anwar Hasan, and Vishal Saraswat, editors, *Security, Privacy, and Applied Cryptography Engineering*, pages 3–26, Cham, 2016. Springer International Publishing.

[Per05] Colin Percival. Cache missing for fun and profit. In *In Proc. of BSDCan 2005*, 2005.

[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.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, pages 142–159, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[PRB09] E. Prouff, M. Rivain, and R. Bevan. Statistical analysis of second order differential power analysis. *IEEE Transactions on Computers*, 58(6):799–811, 2009.

[QS01] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In Isabelle Attali and Thomas Jensen, editors, *Smart Card Programming and Security*, pages 200–210, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[RGV14a] Oscar Reparaz, Benedikt Gierlichs, and Ingrid Verbauwhede. Generic dpa attacks: Curse or blessing? In Emmanuel Prouff, editor, *Constructive Side-Channel Analysis and Secure Design*, pages 98–111, Cham, 2014. Springer International Publishing.

[RGV14b] Oscar Reparaz, Benedikt Gierlichs, and Ingrid Verbauwhede. A note on the use of margins to compare distinguishers. In Emmanuel Prouff, editor, *Constructive Side-Channel Analysis and Secure Design*, pages 1–8, Cham, 2014. Springer International Publishing.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[SSH$^+$14] Alexander Schaub, Emmanuel Schneider, Alexandros Hollender, Vinicius Calasans, Laurent Jolie, Robin Touillon, Annelie Heuser, Sylvain Guilley, and Olivier Rioul. Attacking suggest boxes in web applications over https using side-channel stochastic algorithms. volume 8924, pages 116–130, 08 2014.

[Tim19] Benjamin Timon. Non-profiled deep learning-based side-channel attacks with sensitivity analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(2):107–131, Feb. 2019.

[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.

[WO11] Carolyn Whitnall and Elisabeth Oswald. A comprehensive evaluation of mutual information analysis using a fair evaluation framework. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, pages 316–334, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[WOS14] Carolyn Whitnall, Elisabeth Oswald, and François-Xavier Standaert. The myth of generic dpa. . . and the magic of learning. In Josh Benaloh, editor, *Topics in Cryptology – CT-RSA 2014*, pages 183–205, Cham, 2014. Springer International Publishing.

A Proof of lemma 1

Lemma 1. *Let $f\colon \mathcal{Z} \to \mathbb{R}^n$ be any function. For any leakage model $\varphi\colon \mathcal{Z} \to \mathbb{R}^n$ there exists a decomposition of f into $f = f_2 \circ f_1$, with $f_1 : \mathcal{Z} \to \mathbb{N}$, $f_2 : \mathbb{N} \to \mathbb{R}^n$, satisfying the two following properties:*

1) $\exists\, f_3 : \operatorname{Im} f_1 \to \mathbb{R}^n$ *such that* $f_3 \circ f_1 = \varphi$

2) $\forall z \in \mathcal{Z}, f_2|_{f_1\left(\varphi^{-1}(\{\varphi(z)\})\right)}$ *is bijective of reciprocal* $f_2^{-1}|_{f_2\circ f_1\left(\varphi^{-1}(\{\varphi(z)\})\right)}$

Proof. Let us create a partition of $\mathcal{Z} = \sqcup_{i=1}^{n} P_i$ where two elements $z_1, z_2 \in \mathcal{Z}$ are in the same P_i if and only if:

- $\varphi(z_1) = \varphi(z_2)$
- $f(z_1) = f(z_2)$

Then, one may define f_1 as $f_1(z) = i, \forall z \in P_i$. Since f_1 only collides for z that already collides through φ, there exists f_3 such that $f_3 \circ f_1 = \varphi$. As f is constant on P_i, let us denote by v_i its output on elements of P_i. Then f_2 can be defined as $f_2(i) = v_i$ so that $f_2 \circ f_1 = f$. Now let us prove 2). Let $z \in \mathcal{Z}$ and $a, b \in f_1(\varphi^{-1}(\{\varphi(z)\}))$ such that $f_2(a) = f_2(b)$. There exists z_a and z_b such that $a = f_1(z_a)$ and $b = f_1(z_b)$ with $\varphi(z_a) = \varphi(z_b) = \varphi(z)$. So:

- $\varphi(z_a) = \varphi(z_b)$
- $f_2(f_1(z_a)) = f_2(f_1(z_b)) \iff f(z_a) = f(z_b)$

which means that z_a and z_b are in the same P_i and thus collides through f_1. So $a = b$ which proves that $f_2|_{f_1(\varphi^{-1}(\{\varphi(z)\}))}$ is injective. Then, considering its set of destination being its image, one can say that this function is bijective with reciprocal function: $f_2^{-1}|_{f_2\circ f_1(\varphi^{-1}(\{\varphi(z)\}))}$. □

B Proof of corollary 1

Definition 1. A function f is said wider- than g if there exists another function h such that: $h \circ f = g$.

Corollary 1. *Let L be defined as in (47). Then, for any function $\bar{h}$ wider than HW, $\mathcal{S}_{HW} \geq \mathcal{S}_{\bar{h}}$.*

Proof. There exists h such that $h \circ \bar{h} = \mathrm{HW}$. So:

$$\begin{aligned}\mathcal{S}_{\mathrm{HW}} &= \mathcal{I}\big(\mathrm{HW}(Z_{k^*}), L\big) - \max_{k\neq k^*}\big[\mathcal{I}\big(\mathrm{HW}(Z_k), L\big)\big] \\ &= \mathcal{I}\big(h\circ\bar{h}(Z_{k^*}), L\big) - \max_{k\neq k^*}\big[\mathcal{I}\big(h\circ\bar{h}(Z_k), L\big)\big]\end{aligned} \tag{81}$$

Since removing h in the second term can only increase the information:

$$\mathcal{S}_{\mathrm{HW}} \geq \mathcal{I}\big(h\circ\bar{h}(Z_{k^*}), L\big) - \max_{k\neq k^*}\big[\mathcal{I}\big(\bar{h}(Z_k), L\big)\big] \tag{82}$$

By Th.2, HW maximizes over g the quantity: $\mathcal{I}\big(g(Z_{k^*}), L\big)$, so removing h in the first term cannot increase the information:

$$\begin{aligned}&\mathcal{S}_{\mathrm{HW}} \geq \mathcal{I}\big(\bar{h}(Z_{k^*}), L\big) - \max_{k\neq k^*}\big[\mathcal{I}\big(\bar{h}(Z_k), L\big)\big] \\ &\mathcal{S}_{\mathrm{HW}} \geq \mathcal{S}_{\bar{h}}\end{aligned} \tag{83}$$

□

C Complementary material on the entropy

Lemma 2. *Let A and B be a two discrete random variables. Let $f\colon \mathcal{A} \to \mathbb{R}^n$ be any function. Then:*

$$\mathcal{H}\big(f(A) \mid B\big) \le \mathcal{H}(A \mid B) \tag{84}$$

Proof. The data processing inequality [BR12] ensures that applying f to any variables can not increase its mutual information with another variable so:

$$\begin{aligned} \mathcal{I}\big(f(A), f(A) \mid B\big) &\le \mathcal{I}(A, A \mid B) \\ \mathcal{H}\big(f(A) \mid B\big) &\le \mathcal{H}(A \mid B) \end{aligned} \tag{85}$$

□

Lemma 3. *Let A and B be a two discrete random variables. Let $f\colon \mathcal{A} \to \mathbb{R}^n$ be any function. Then:*

$$\mathcal{H}\big(A \mid f(B)\big) \ge \mathcal{H}(A \mid B) \tag{86}$$

Proof. Again, using the data processing inequality [BR12]:

$$\begin{aligned} \mathcal{I}\big(A, f(B)\big) &\le \mathcal{I}(A, B) \\ \mathcal{H}(A) - \mathcal{H}\big(A \mid f(B)\big) &\le \mathcal{H}(A) - \mathcal{H}(A \mid B) \\ \mathcal{H}\big(A \mid f(B)\big) &\ge \mathcal{H}(A \mid B) \end{aligned} \tag{87}$$

□

D Network architectures

Figure 6 and Figure 7 show the network architectures used for the experiments performed respectfully with MINE and classifiers (supervised and DDLA). For fairness, we tried to keep the two architectures as close as possible. The optimizer used in both cases is Adam [KB14] with default parameters. The loss function used for the classifiers is the categorical cross-entropy. Note that when using convolutional layers with MINE, the convolutional layers should only be applied to the trace variable and not to $f(Z_k)$ which would not make sense.

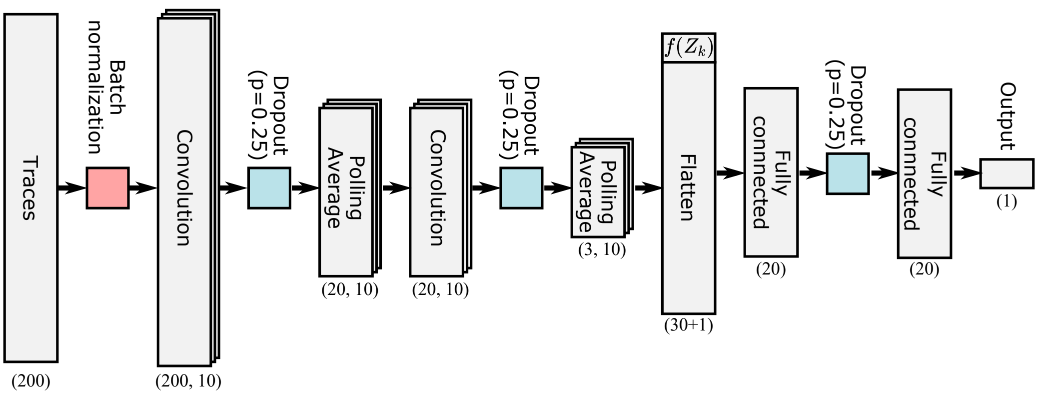


Figure 6: Network architecture for MINE

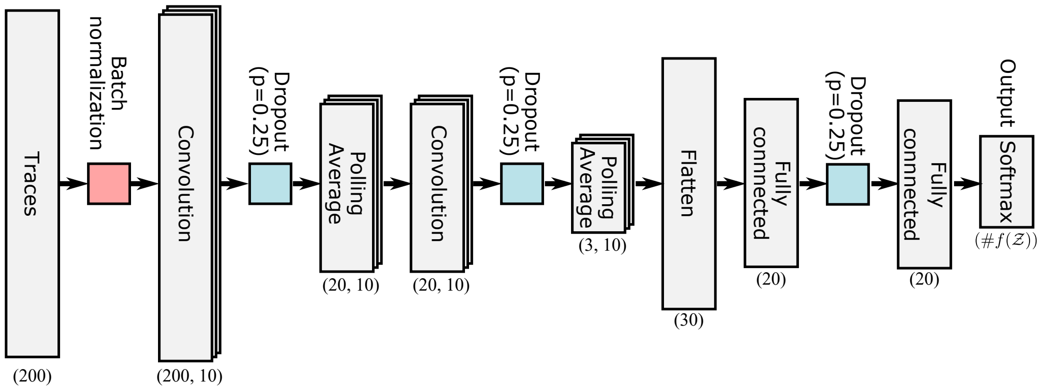


Figure 7: Network architecture for the classifiers (Supervised and DDLA)