

# Mutual Information Analysis: a Comprehensive Study <sup>★</sup>

Lejla Batina<sup>1,2</sup>, Benedikt Gierlichs<sup>1</sup>, Emmanuel Prouff<sup>3</sup>, Matthieu Rivain<sup>4</sup>,  
François-Xavier Standaert<sup>5\*\*</sup> and Nicolas Veyrat-Charvillon<sup>5</sup>

<sup>1</sup> K.U.Leuven, ESAT/SCD-COSIC and IBBT,  
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium.  
{lejla.batina, benedikt.gierlichs}@esat.kuleuven.be

<sup>2</sup> Radboud University Nijmegen, CS Dept./Digital Security group,  
Heyendaalseweg 135, 6525 AJ, Nijmegen, The Netherlands.

<sup>3</sup> Oberthur Technologies,  
71-73 rue des Hautes Pâtures, 92726 Nanterre Cedex, France.  
e.prouff@oberthur.com

<sup>4</sup> CryptoExperts, Paris, France.

matthieu.rivain@cryptoexperts.com

<sup>5</sup> Université catholique de Louvain, UCL Crypto Group,  
B-1348 Louvain-la-Neuve, Belgium.  
{fstandae, nicolas.veyrat}@uclouvain.be

**Abstract.** Mutual Information Analysis is a generic side-channel distinguisher that has been introduced at CHES 2008. It aims to allow successful attacks requiring minimum assumptions and knowledge of the target device by the adversary. In this paper, we compile recent contributions and applications of MIA in a comprehensive study. From a theoretical point of view, we carefully discuss its statistical properties and relationship with probability density estimation tools. From a practical point of view, we apply MIA in two of the most investigated contexts for side-channel attacks. Namely, we consider first order attacks against an unprotected implementation of the DES in a full custom IC and second order attacks against a masked implementation of the DES in an 8-bit microcontroller. These experiments allow to put forward the strengths and weaknesses of this new distinguisher and to compare it with standard power analysis attacks using the correlation coefficient.

**Keywords.** Side-Channel Analysis, Mutual Information Analysis, Masking Countermeasure, Higher-Order Attacks, Probability Density Estimation.

---

<sup>★</sup> Work supported in part by the IAP Programme P6/26 BCRYPT of the Belgian State, by FWO project G.0300.07, by the Walloon region through the project SCEPTIC, by the European Commission under grant agreement ICT-2007-216676 ECRYPT NoE phase II and by K.U. Leuven-BOF.

<sup>\*\*</sup> Associate researcher of the Belgian Fund for Scientific Research (F.R.S.-FNRS).

## 1 Introduction

Embedded devices such as smart cards, mobile phones, PDAs and more recently RFID tags or sensor networks are now closely integrated in our everyday lives. These devices typically operate in hostile environments and hence, the data they contain might be relatively easily compromised. For example, their physical accessibility sometimes allows a number of very powerful attacks against cryptographic implementations. Contrary to classical cryptanalyses that target the mathematical algorithms, such physical attacks take advantage of the peculiarities of the devices on which the algorithms are running. One of the most famous (and devastating) examples of physical attack is Differential Power Analysis (DPA), introduced by Kocher *et al.* in 1998 [15]. It demonstrates that by monitoring the power consumption of a smart card, the cryptographic keys can be rather efficiently extracted if no special countermeasures are taken. In the last decade, many other side-channels were exhibited, including timing [14] and electromagnetic radiation [7, 23]. Both the theory and practice have been improved, leading to advanced attacks such as correlation attacks [2], template attacks [3] and higher-order attacks [18]. In addition, various types of countermeasures, such as masking [10], or hiding [34], as well as better tools to analyze and evaluate these attacks and countermeasures [28], have been proposed. A state-of-the-art view of power analysis attacks can be found in [17].

The core idea of differential side-channel attacks is to compare some key-dependent predictions of the physical leakages with actual measurements, in order to identify which prediction (or key) is the most likely to have given rise to the measurements. In practice, it requires both to be able to model the leakages with a sufficient precision, in order to build the predictions, and to have a good comparison tool (also called distinguisher) to efficiently extract the keys. At CHES 2008, a side-channel distinguisher called Mutual Information Analysis (MIA) was introduced [8]. This distinguisher aims at generality in the sense that it is expected to lead to successful attacks without requiring specific knowledge of, or restrictive assumptions about the device it targets. In other words, it can cope with less precise leakage predictions than other types of side-channel attacks. This generality comes at the price of a limited decrease of the attack efficiency (*i.e.* an increase in the number of measurements required to perform a successful key recovery) when the leakage model fits well enough to the physics. For example, standard attacks using a correlation coefficient may work better if the physical leakages linearly depend on the Hamming weight of the data processed in a device, in the presence of a close to Gaussian noise distribution.

From a theoretical point of view, MIA can be seen as the non-profiled (or unsupervised) counterpart of the information theoretic metric that has been established in [28] as a measure of side-channel leakage. Hence, its main advantage is that it can detect any (*e.g.* not only linear or monotonic) kind of data dependency in the physical measurements. As a consequence, MIA is a useful tool when evaluating the security of an implementation, in order to demonstrate its side-channel attack resistance. By contrast, a less general distinguisher may give

a false sense of security, just because it cannot capture the data dependencies at hand. In general, MIA is well suited for “difficult” attack scenarios where standard assumptions about the leakage behavior of a device may not hold.

Following the original work of Gierlichs *et al.* [8], various recent publications investigated theoretical and practical issues related to MIA. For example, [19, 20, 33] discuss the statistical properties of the original distinguisher; [9, 20, 22, 30] consider its application to implementations protected by masking or other countermeasures; and [29] performs exhaustive empirical comparisons of various side-channel distinguishers, including MIA. In this paper, we compile these recent results into a single comprehensive treatment. The rest of the paper is organized as follows. In Section 2, we recall the diverse information theoretic definitions that are the theoretical background of MIA. We also describe our model for side-channel attacks, inspired by [8, 28]. In Section 3, we carefully investigate the properties of MIA when applied in a univariate attack scenario. In particular, we detail the impact of a good probability density estimation when performing the attacks. Section 4 addresses the advanced context of implementations protected with a masking countermeasure. Finally, Section 5 gives our conclusions and lists some open problems. All our analyses are backed up with experimental results. This allows us to put forward the interesting features of MIA compared to other techniques used to attack leaking devices.

## 2 Preliminaries

### 2.1 Information theoretic definitions

**Entropy.** Let  $X$  be a random variable on a (discrete) space  $\mathcal{X}$ , and  $x$  an element from  $\mathcal{X}$ . For every positive integer  $d$ , we denote by  $\mathbf{X}$  a  $d$ -dimensional random vector  $(X_1, \dots, X_d) \in \mathcal{X}^d$ , and by the letter  $\mathbf{x}$  an element from  $\mathcal{X}^d$ .

The (Shannon) entropy [4] of a random variable  $X$  on a discrete space  $\mathcal{X}$  is a measure of its uncertainty during an experiment. It is defined as:

$$H[X] = - \sum_{x \in \mathcal{X}} \Pr[X = x] \cdot \log(\Pr[X = x]).$$

The joint entropy of a pair of random variables  $(X, Y)$  expresses the uncertainty one has about the combination of these variables:

$$H[X, Y] = - \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \Pr[X = x, Y = y] \cdot \log(\Pr[X = x, Y = y]).$$

The joint entropy is always greater than or equal to that of either variable, with equality if and only if (iff)  $Y$  is a deterministic function of  $X$ . It is also sub-additive, and equality occurs iff the two variables are independent:

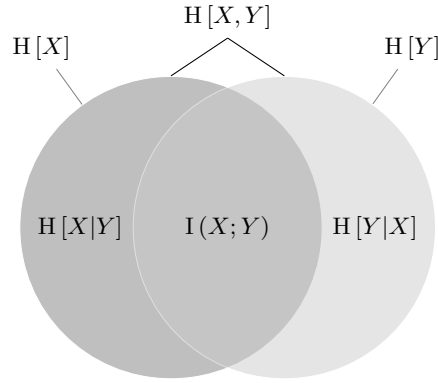
$$\max(H[X], H[Y]) \leq H[X, Y] \leq H[X] + H[Y].$$

Finally, the conditional entropy of a random variable  $X$  given another variable  $Y$  expresses the uncertainty on  $X$  which remains once  $Y$  is known:

$$H[X|Y] = - \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \Pr[X = x, Y = y] \cdot \log(\Pr[X = x|Y = y]).$$

The conditional entropy is always greater than or equal to zero, with equality iff  $X$  is a deterministic function of  $Y$ . It is also at most equal to the entropy of  $X$ , and equality occurs iff the two variables are independent:

$$0 \leq H[X|Y] \leq H[X].$$



**Fig. 1.** Information diagram.

All these relations are depicted in Figure 1. They can be straightforwardly extended to continuous spaces by turning the previous sums into integrals. For example, in this case the differential entropy is defined as:

$$H[X] = - \int_{\mathcal{X}} \Pr[X = x] \cdot \log(\Pr[X = x]) dx.$$

The differential entropy can be negative, contrary to the discrete one. In order to easily deal with hybrid situations combining discrete and continuous variables, we denote by  $\Pr[X = x]$  the value in  $x$  of the probability density function (pdf for short) of the continuous variable  $X$  (generally denoted as  $f_X(x)$ ).

**Mutual information.** The mutual information is a general measure of the dependence between two random variables. It expresses the quantity of information one has obtained on  $X$  by observing  $Y$ . On a discrete domain, the mutual information of two random variables  $X$  and  $Y$  is defined as:

$$I(X; Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \Pr[X = x, Y = y] \cdot \log \left( \frac{\Pr[X = x, Y = y]}{\Pr[X = x] \cdot \Pr[Y = y]} \right).$$

It can be seen as the Kullback-Leibler divergence [4] between the joint distribution  $\Pr[X = x, Y = y]$  and the product distribution  $\Pr[X = x] \cdot \Pr[Y = y]$ . The mutual information can similarly be expressed as the expected value over  $\mathcal{X}$  of the divergence between the conditional probability  $\Pr[Y = y|X = x]$  and the marginal probability  $\Pr[Y = y]$ :

$$I(X; Y) = \sum_{x \in \mathcal{X}} \Pr[X = x] \sum_{y \in \mathcal{Y}} \Pr[Y = y|X = x] \cdot \log \left( \frac{\Pr[Y = y|X = x]}{\Pr[Y = y]} \right).$$

It is directly related to Shannon's entropy through the following equations:

$$\begin{aligned} I(X; Y) &= H[X] - H[X|Y], \\ &= H[X] + H[Y] - H[X, Y], \\ &= H[X, Y] - H[X|Y] - H[Y|X]. \end{aligned}$$

The mutual information is always greater than or equal to zero, with equality iff  $X$  and  $Y$  are independent. It is lower than the entropy of either variable, and equality only occurs iff one variable is a deterministic function of the other. The higher the mutual information, the stronger the dependency between  $X$  and  $Y$ :

$$0 \leq I(X; Y) \leq \min(H[X], H[Y]).$$

It can again be straightforwardly extended to the continuous case:

$$I(X; Y) = \int_{\mathcal{X}} \int_{\mathcal{Y}} \Pr[X = x, Y = y] \cdot \log \left( \frac{\Pr[X = x, Y = y]}{\Pr[X = x] \cdot \Pr[Y = y]} \right) dx dy.$$

Eventually, the mutual information between a discrete random variable  $X$  and a continuous random variable  $Y$  is defined as:

$$I(X; Y) = \sum_{x \in \mathcal{X}} \Pr[X = x] \int_{\mathcal{Y}} \Pr[Y = y|X = x] \cdot \log \left( \frac{\Pr[Y = y|X = x]}{\Pr[Y = y]} \right) dy,$$

or equivalently:

$$I(X; Y) = \sum_{x \in \mathcal{X}} \int_{\mathcal{Y}} \Pr[X = x, Y = y] \cdot \log \left( \frac{\Pr[X = x, Y = y]}{\Pr[X = x] \cdot \Pr[Y = y]} \right) dy.$$

## 2.2 Pearson’s correlation coefficient

Pearson’s correlation coefficient is a simple measure of dependence between two random variables  $X$  and  $Y$ . Computing it does not require to know the probability density functions of  $X$  and  $Y$ , but it can express only the linear dependence between these variables (whereas mutual information is able to detect any kind of dependence). It is defined as follows:

$$\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \cdot \sigma_Y} = \frac{E[XY] - E[X] \cdot E[Y]}{\sigma_X \cdot \sigma_Y}. \quad (1)$$

In this equation,  $\text{cov}(X, Y)$  is the covariance between  $X$  and  $Y$ ,  $E[X]$  denotes the expected value of  $X$  and  $\sigma_X$  the standard deviation of  $X$ . The correlation coefficient satisfies the following inequality:

$$0 \leq |\rho(X, Y)| \leq 1,$$

with the upper bound achieved iff  $Y$  is an affine function of  $X$ . The lower bound is achieved if  $X$  and  $Y$  are independent but the opposite does not hold:  $X$  and  $Y$  can be dependent and have their correlation equal to zero.

## 2.3 Side-channel cryptanalysis

In a side-channel attack, an adversary tries to recover secret information from a leaking implementation, *e.g.* a software program or an IC computing a cryptographic algorithm. As most cryptanalytic techniques, side-channel attacks are based on a divide-and-conquer strategy. For example, in the context of a block cipher implementation, one typically targets small pieces of the master key or a round key - called subkeys in the following - one by one. The core idea is to compare subkey-dependent models of the leakages with actual measurements. That is, for each subkey candidate, the adversary builds models that correspond to the leakage generated by the encryption of different plaintexts. Then, he evaluates which model (*i.e.* which subkey) gives rise to the best prediction of the physical leakages, measured for the same set of plaintexts. As a matter of fact and assuming that the models can be represented by a random variable  $X$  and the leakages can be represented by a random variable  $Y$ , side-channel analysis can be seen as the problem of detecting a dependence between these two variables.

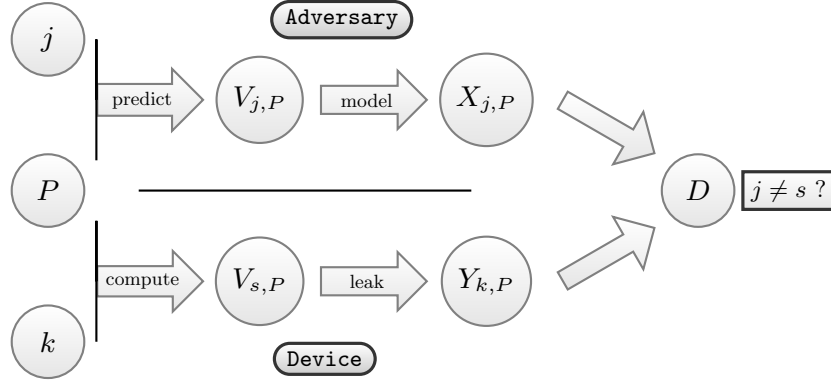
In the rest of this paper, we consider that  $X$  is a discrete random variable and that  $Y$  is a continuous random variable that is sampled with a sufficient resolution (*e.g.* using an oscilloscope). A consequence is that we also considered pdf estimation techniques designed for continuous distributions (in Section 3.2, 4.4).

The next sections of the paper analyze the attack depicted in Figure 2, following the models in [8] and [28]. That is, we consider a device performing several cryptographic computations  $E_k(p)$  on different plaintexts  $p$  drawn uniformly from the text space  $\mathcal{P}$ , using some fixed key  $k$  drawn from the key space  $\mathcal{K}$ . While computing  $E_k(P)$  (where  $P$  is a random variable over  $\mathcal{P}$ ), the device

will handle some intermediate values (defined as sensitive variables in [24]) that depend on the known input  $P$  and the unknown key  $k$ . In practice, the interesting sensitive variables in a DPA attack are the ones that only depend on an enumerable subkey  $s$ : we denote them as  $V_{s,P}$ . Anytime such a sensitive intermediate value is computed, the device generates some physical leakage, denoted as  $Y_{k,P}$  (that potentially depends on all the key  $k$ , including the subkey  $s$ ).

In order to perform a key recovery, an adversary first has to select a sensitive value. Given that this value only depends on a subkey  $s$ , he can then evaluate its result for the same plaintexts that have been used to generate  $Y_{k,P}$  and all the possible subkey candidates  $j \in \mathcal{S}$ . It gives rise to different hypothetical values  $V_{j,P}$ . Afterwards, the adversary uses a leakage model to map these values from their original space  $\mathcal{V}$  towards a hypothetical leakage space  $\mathcal{X}$ . For example, a usual model (that has been experimentally confirmed in numerous works *e.g.* see [17]) is to take the Hamming weight of the values  $V_{j,P}$ . As a result, he obtains  $|\mathcal{S}|$  different models denoted as  $X_{j,P}$ , again corresponding to the different subkey candidates. Eventually, he uses a distinguisher  $D$  to compare the different models  $X_{j,P}$  with the actual leakages  $Y_{k,P}$ . If the attack is successful, the best comparison result (*i.e.* the highest value of the distinguisher) should be obtained for the correct subkey candidate  $j = s$ . This procedure can then be repeated for different subkeys in order to eventually recover the full key.

We mention that [8] uses the terms hypothetical leakages for  $X_{j,P}$  and observations for  $Y_{k,P}$  while [28] uses the terms models for  $X_{j,P}$  and leakages for  $Y_{k,P}$ . We use the latter terminology in the following, but both are equivalent.



**Fig. 2.** Schematic illustration of a side-channel key recovery attack.

### 3 Univariate MIA

#### 3.1 Basic principle

Following the previous informal description, the goal of a distinguisher is to detect the dependencies between two random variables. For example, in the case of a correlation attack [2], one simply needs to compute:

$$d_j = \hat{\rho}(X_{j,P}, Y_{k,P}),$$

where the hat sign indicates that we use an estimator. In practice, a usual choice is the Pearson coefficient where the expected values and standard deviations of Equation (1) are replaced by sample means and standard deviations. Similarly, the distinguisher used in a mutual information analysis can be written as:

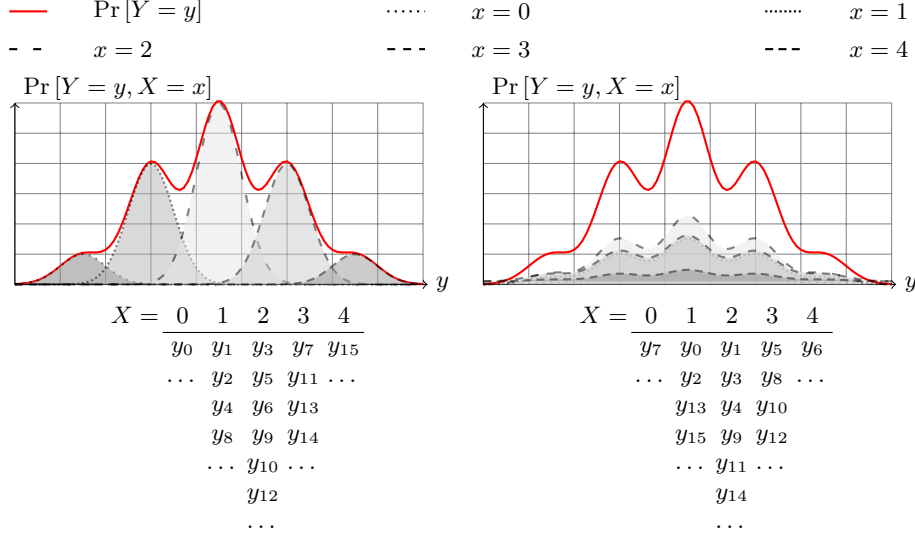
$$d_j = \hat{I}(X_{j,P}; Y_{k,P}).$$

For simplicity, we will omit the different subscripts of  $X$  and  $Y$  in the remainder of the paper. The idea behind this latter procedure is that a meaningful partition of  $Y$ , where each subset corresponds to a particular model value, will relate to a side-channel sample distribution  $\Pr[Y|X = x]$  distinguishable from the global distribution of  $\Pr[Y]$ . The estimated mutual information will then be larger than zero. By contrast, if the key guess is incorrect, the false predictions will form a partition corresponding to a random sampling of  $Y$  and therefore simply give scaled images of the global side-channel pdf. Hence, the estimated mutual information will be equal (or close) to zero in this case.

**Example.** Let us consider a target implementation in which the adversary receives leakages of the form  $y = H_w(S(p \oplus s)) + n$  where  $H_w$  is the Hamming weight function,  $S$  the 4-bit S-box of the block cipher Serpent,  $p$  a known plaintext,  $s$  the target subkey of the attack and  $n$  is a Gaussian noise. Let us also assume that the model  $X$  corresponds to  $H_w(S(p \oplus j))$ . Figure 3 illustrates what happens asymptotically for the correct and a wrong subkey hypothesis in the case of this attack. It shows the higher dependence for the correct subkey (*i.e.* in the left part of the figure) than for an incorrect one, as expected in a successful attack.

In theory, MIA tests a null hypothesis stating that the predicted leakages and the measured ones are independent if the subkey hypothesis is false. When this hypothesis is not verified, the adversary assumes that he found the correct subkey. However, in practice there may exist certain dependencies between a wrong subkey candidate and the actual leakages (*e.g.* the ghost peaks as defined in [2]). Hence, the adversary generally selects the subkey that leads to the highest distinguisher value. The efficiency of a distinguisher can be measured with a success rate. As discussed in [28], a success can be strictly defined as a situation in which the distinguisher reaches its maximum for the correct subkey (as we will consider in the following), or softly defined as a situation in which the correct subkey is highly rated by the distinguisher. Alternative metrics like the guessing entropy can also be used to quantify how much a side-channel attack reduces the average workload required to complete a key recovery.





**Fig. 3.** Joint probability densities  $\Pr[Y = y, X = x]$  for different model values  $X = x$  and marginal leakage probability  $\Pr[Y = y]$  densities for the correct (left) and a wrong (right) subkey hypothesis in the case of a 4-bit DPA attack.

### 3.2 PDF estimation tools

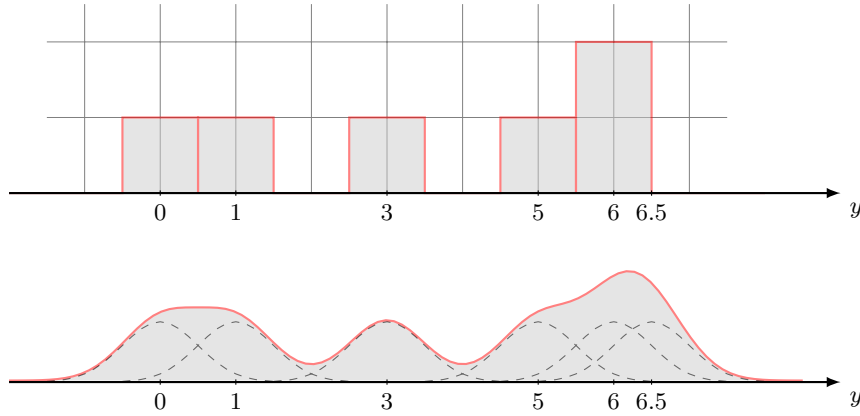
In order to perform a mutual information analysis, one first has to estimate the probability density function of the joint distribution  $\Pr[X = x, Y = y]$  (or alternatively, the conditional distribution  $\Pr[Y = y|X = x]$  and the marginal distribution  $\Pr[Y = y]$ ) from a limited number of samples. In other words, one needs to estimate the distribution of the leakages  $Y$  for different model values  $X = x$ . The problem of modeling a probability density function from random samples is a well studied problem in statistics, referred to as density estimation. Several solutions exist, ranging from simple histograms to kernel density estimation, data clustering [35] and vector quantization [31]. The authors of [8] used histograms for density estimation as a proof of concept for MIA. But in certain contexts, an attack's efficiency (regarding the number of traces needed to recover the key) can be improved by using more advanced techniques, possibly at the cost of a higher computation load and memory requirement.

For example, some density estimation tools have been initially suggested in [37] as relevant to side-channel attacks and then applied to MIA in [20, 33]. In this section, we present two common ways of approximating densities, namely histograms and kernels. We mention that these are two non-parametric methods. Since one interesting feature of MIA is that it does not rely on particular assumptions on the leakage distributions, it seems a reasonable starting point. However, parametric tools making more specific assumptions (*e.g.* that the pdf of the leakages is a Gaussian mixture) could improve the efficiency of the attacks in certain practically meaningful implementation contexts [16, 20].

**Histograms.** Histogram estimation performs a partition of the samples by grouping them into bins, as illustrated in the upper part of Figure 4. More precisely, each bin contains the samples of which the value falls into a certain range. The respective ranges of the bins have equal width and form a partition of the range between the extreme values of the samples. Using this method, one approximates a probability by dividing the number of samples that fall within a bin by the total number of samples. For  $n$  bins denoted as  $b(i)$ , the probability is estimated as:

$$\hat{\Pr}[y \in b(i)] = \frac{\#b(i)}{q},$$

where  $\#b(i)$  is the number of samples in bin  $b(i)$  and  $q = \sum_{i=1}^n \#b(i)$  is the total number of samples. The optimal choice for the bin width  $h$  is an issue in statistical theory, as different bin sizes can have great impact on the estimation. For simple Gaussian distributions, reasonable choices are Scott's rule [25] ( $h = 3.49 \times \hat{\sigma}(Y) \times q^{-1/3}$ ) and Freedman-Diaconis rule [6] ( $h = 2 \times \text{IQR}(Y) \times q^{-1/3}$ ,  $\text{IQR} = \text{interquartile range}$ ). In side-channel attacks, and in particular for wrong key hypotheses, one has to estimate leakage distributions which comprise multiple source distributions (or components), *e.g.* Gaussian mixtures. While the above mentioned methods can yield acceptable results in practice, their theoretical foundation is not necessarily provided [13], in particular since in general, the best pdf estimation does not necessarily give rise to the best subkey discrimination. In [8] Gierlichs *et al.* suggest a different and simpler rule: namely to choose the number of bins equal to the number of expected components in the distribution, which is equal to the number of distinct model values.



**Fig. 4.** Histogram (top) and kernel-based (bottom) density estimations (thick line) resulting from sample set  $\{0, 1, 3, 5, 6, 6.5\}$ , using bin width  $h = 1$  or gaussian kernels (dashed lines) with bandwidth  $h = 0.5$ , respectively.

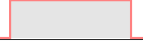

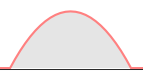





**Kernels.** Kernel density estimation is a generalization of histograms. Instead of bundling samples together in bins, it adds (for each observed sample) a small kernel centered on the value of the leakage to the estimated pdf, as illustrated in the lower part of Figure 4. The resulting estimation is a sum of small “bumps” that is much smoother than the corresponding histogram, which can be desirable when estimating a continuous distribution. In such cases it usually provides faster convergence towards the true distribution. Note that although this solution requires to select a kernel and a bandwidth, it does not assume anything more about the estimated pdf than histograms. The probability is estimated as:

$$\hat{\Pr}[Y = y] = \frac{1}{qh} \sum_{i=1}^q K\left(\frac{y - y^i}{h}\right),$$

where  $y^i$  denote the leakage samples and the kernel function  $K$  is a real-valued integrable function satisfying  $\int_{-\infty}^{\infty} K(u) du = 1$  and  $K(u) = -K(u)$  for all  $u$ .

Some kernel functions are represented in Table 1. Similarly to histograms, the most important parameter is the bandwidth  $h$ . Its optimal value is the one minimizing the AMISE (Asymptotic Mean Integrated Squared Error), which itself usually depends on the true density. A number of approximation methods have been developed, see [32] for an extensive review. In our case, we used the modified rule of thumb estimator [12, 27]:

$$h = 1.06 \times \min\left(\hat{\sigma}(Y), \frac{\text{IQR}(Y)}{1.34}\right) \times q^{-\frac{1}{5}}.$$

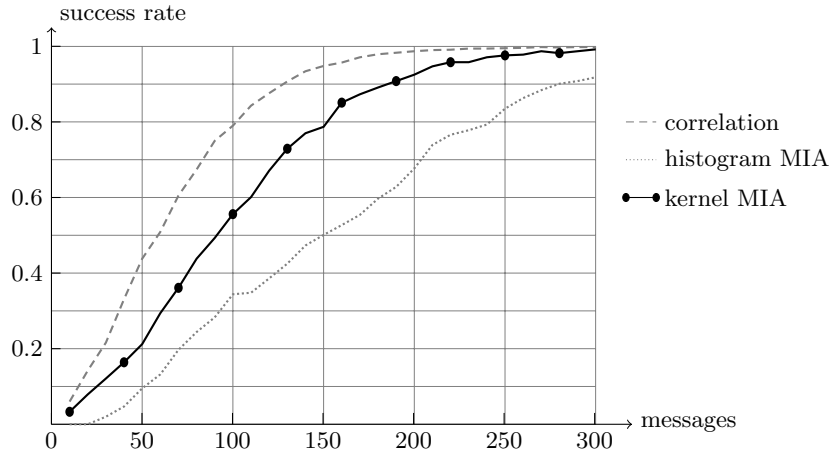
kernel	$K(u)$	kernel	$K(u)$
Uniform	$\frac{1}{2}i(u)$ 	Triangle	$(1 -  u )i(u)$ 
Epanechnikov	$\frac{3}{4}(1 - u^2)i(u)$ 	Quartic	$\frac{15}{16}(1 - u^2)^2i(u)$ 
Triweight	$\frac{35}{32}(1 - u^2)^3i(u)$ 	Tricube	$\frac{70}{81}(1 -  u ^3)^3i(u)$ 
Gaussian	$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right)$ 	Cosinus	$\frac{\pi}{4} \cos\left(\frac{\pi}{2}u\right) i(u)$ 

**Table 1.** Some kernel functions.  $i$  is defined as:  $i(u) = 1$  if  $|u| \leq 1$ , 0 otherwise.

### 3.3 Experiments

The previous subsections discussed the theoretical ideas behind univariate MIA. Quite naturally, it is also interesting to evaluate the extent up to which different pdf estimations affect the efficiency of the distinguisher in practice, and how this distinguisher relates to standard attacks using the correlation coefficient. For this purpose, we carried out attacks based on the traces that are publicly available in the DPA Contest [5]. We computed the first-order success rate as function of the number of traces available to the adversary (*i.e.* encrypted messages), over 1000 independent experiments, using a Hamming distance leakage model.

The results of these experiments are represented in Figure 5, from which we can extract two main observations. First, classical attacks using the correlation coefficient are the most efficient in this simple context, where the models closely fit to the physics. Second, the choice of a pdf estimation tool has a significant impact on the efficiency of MIA. In particular, the kernel-based MIA seems to perform better than its counterpart using histograms (we used 5-bin histograms, following [8]). This can be explained by the large amount of algorithmic noise that is present in the DPA contest measurements (*i.e.* the targeted architecture contains a 64-bit register for the state, while the attack merely targets four of them). More examples of univariate MIA experiments can be found in [19, 29].



**Fig. 5.** Success rate of different attacks against the first DES S-box in the DPA Contest.

Note that when applying a kernel-based MIA, it is only the distribution of the continuous random variable  $Y$  that is approximated with kernels. For the discrete random variable  $X$ , we directly estimated the probability mass function. In other words, we estimated one conditional distribution of  $Y$  per model value  $X = x$ .

### 3.4 Discussion

**Comparison with correlation attacks.** As originally advertised by Gierlichs *et al.* [8], MIA is a generic distinguisher in the sense that it can capture any type of dependency between an adversary’s models and actual physical leakages. For example, a successful correlation attack requires that  $\hat{\rho}(X_s, Y) > \hat{\rho}(X_j, Y)$ , for all subkey candidates  $j \in \mathcal{S}$  and  $j \neq s$ . A successful MIA rather requires that  $\hat{I}(X_s; Y) > \hat{I}(X_j; Y)$ . Hence, there are situations where correlation attacks are unable to exploit the leakage dependencies while MIA can still succeed. However, the fact that MIA can exploit more general dependencies is not directly related to the efficiency of the distinguisher (*i.e.* its speed to discriminate the correct key candidate). As illustrated in Figure 5, if the models and the leakages are reasonably related through a linear relation, then the correlation coefficient can do an excellent job in characterizing this relation quite fast.

In other words, there is a tradeoff between the efficiency of an attack and the amount of assumptions required to mount it. In this respect, even the application of MIA (that is clearly designed to work with little assumptions) can take advantage of carefully selected parameters, as we now detail.

**Choice of the model.** As for any DPA attack, the good selection of a leakage model highly influences the efficiency of a distinguisher. The better a model relates to the actual physics, the easier their relation will be observed through the distinguisher. As detailed in [33], MIA better resists to model inaccuracies than, *e.g.* correlation attacks. But a completely wrong model will not allow any key recovery at all, for any attack. In fact, the problem of finding a good leakage model is similar for all distinguishers and mainly relates to the engineering intuition about the target device. In the following, we simply ensure that different attacks are fed with the same models, when comparing them.

Note that regardless their connection to the physics, there are certain models that will not be useful to the MIA distinguisher. For example, as detailed in [8, 20], attacking bijective S-boxes such as the AES Rijndael ones, with the identity leakage model  $X = V$ , will not lead to successful attacks. This is because different subkey candidates merely lead to different permutations of a certain partition in this context. And this feature has no effect on the conditional entropy and the corresponding mutual information. This is in contrast with correlation attacks, which may still be able to detect a (weak) linear dependence if it exists, but not specific to MIA: most attacks based on leakage partitions suffer from the same limitation [29, 33, 35]. We note that if an adversary aims to be perfectly generic and to use an identity leakage model, he can always target intermediate variables that do not bijectively depend on the key for a given plaintext. This is possible either due to the S-box properties, as in the DES, or because he decides to leave out some bits of the target values, *e.g.* predicting 6 bits out of 8 of the AES S-box output, at the cost of a slightly increased algorithmic noise.

**Choice of the number of bins / bandwidth.** In the same line, pdf estimation tools also require to fix, *e.g.* the number of bins or the kernel bandwidth. It implies a similar tradeoff between efficiency and flexibility. The more bins (or the smaller the bandwidth), the more precise the pdf estimation and the more dependencies can be estimated. But on the other hand, adding bins or reducing the bandwidth also implies the need of more samples to reach a proper estimate of the leakage pdf, which generally slows down the key extraction.

Summarizing, one of the main interests of MIA is that it allows running differential side-channel attacks with minimum assumptions on the underlying hardware. This can already be an advantage in certain univariate attacks (*e.g.* if very little is known about the leakage model of an implementation protected with a dual rail logic style such as [34], or if the exploitation of the leakage is non trivial [22]). As the next section will underline, MIA can also be straightforwardly extended towards second- and higher-order side-channel attacks.

## 4 Multivariate MIA against masked implementations

Masking is one of the most widely used countermeasures to protect implementations of block ciphers against side-channel analysis (see, *e.g.* [1, 10]). Efficient side-channel key recovery in the presence of masking is therefore an important issue for the security of embedded cryptography. In this section, we explain how MIA can be generalized to break masked implementations. For this purpose, we briefly recall the principles of masking and higher-order side-channel attacks first. Then, the rest of the section exactly follows the structure of the univariate case (*i.e.* we present the basic principle of the attack, describe two pdf estimation tools, provide experiments and discuss our results in Sections 4.1 to 4.6).

### 4.1 The masking countermeasure

The basic principle of masking can be explained as follows: every sensitive variable  $v$  occurring during the computation is randomly split into  $d$  shares  $v_1, \dots, v_d$  in such a way that the following relation is satisfied for a group operation  $\star$ :

$$v_1 \star v_2 \star \dots \star v_d = v . \quad (2)$$

Typically, one can use the bitwise XOR or a modular addition as group operation. The  $d - 1$  shares  $v_2, \dots, v_d$  (called the masks) are randomly chosen and the last one,  $v_1$  (called the masked variable) is processed such that it satisfies (2).

Assuming that the masks are uniformly distributed, masking renders every single intermediate value during a cryptographic computation non-sensitive. As a result, the univariate side-channel attacks of the previous section are not possible anymore. But the vector of leakages  $y_1, \dots, y_d$  resulting from the observation of the  $d$  shares is still dependent on a sensitive variable. Consequently, masking can be overcome by higher-order side-channel attacks that jointly exploit the

leakages of several intermediate variables. The goal of such attacks is to exhibit a dependency between the  $d$ -dimensional random vector  $\mathbf{Y}$  associated to the shares' leakages and the random variable  $X_j$  associated to the attacker's models for a key guess  $j \in \mathcal{S}$ . In the following, we first limit ourselves to second-order attacks and denote the random vector  $\mathbf{Y}$  by the couple of random variables  $(Y_1, Y_2)$ . The generalization to higher-orders will be discussed in Section 4.6.

## 4.2 Second-order differential power analysis

Second-order DPA has been initially introduced by Messerges in [18] to defeat masking in the case  $d = 2$ , using the difference-of-means test as distinguisher. It has since been improved by using Pearson's correlation coefficient. In order to apply such a distinguisher, second-order DPA first applies a combining function  $C$  to the pair of leakages  $(Y_1, Y_2)$ . As a result, a univariate signal that can be correlated with the adversary's models is obtained. The attack is then similar to the first-order case and just consists in estimating the coefficient:

$$d_j = \hat{\rho}(X, C(Y_1, Y_2)).$$

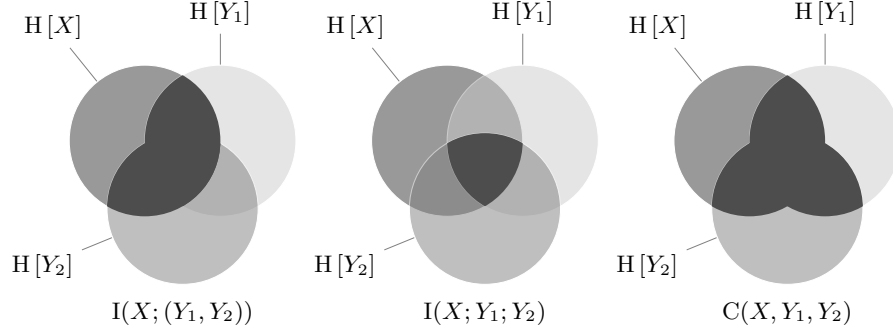
If the combining function  $C$  and the model  $X$  are well-chosen (which is briefly discussed in Section 4.6), then the maximum correlation should again be observed for the correct subkey candidate  $j = s$ . Several combining functions have been proposed in the literature. Two of them are commonly used: the product combining [1] which consists in multiplying the two signals and the absolute difference combining [18] which computes the absolute value of the difference between two signals. [21] confirmed the hint of [1] that centering the leakages before combining them by product yields a better combining function in the context of a second-order DPA with leakages closely following a Hamming weight model. The resulting normalized product combining is defined as:

$$C(Y_1, Y_2) = (Y_1 - \hat{E}[Y_1]) \cdot (Y_2 - \hat{E}[Y_2]).$$

Although second-order attacks using the correlation coefficient do lead to successful key recoveries, they are not optimal from an information theoretic point of view. Indeed, the application of a combining function to the leakages inevitably leads to a loss of information [1]. This motivates the investigation of alternative distinguishers in this context. The next subsection details how MIA can be extended to higher-order attacks, as independently described in [9, 20].

## 4.3 Multivariate MIA: basic principle

In general, the mutual information is a multivariate operator that can easily deal with the dependencies of multiple variables. Taking the example of two-dimensional leakages  $\mathbf{Y} = (Y_1, Y_2)$ , the information diagram of Figure 6 directly suggests that there are different ways to detect such dependencies, including:



**Fig. 6.** Information diagram for second-order attacks.

1. One can simply consider the mutual information between the models  $X$  and the random vector  $\mathbf{Y} = (Y_1, Y_2)$ , just as in Section 2.1, *i.e.* compute:

$$I(X; \mathbf{Y}) = \sum_{x \in \mathcal{X}} \Pr[X = x] \int_{\mathbf{y}^2} \Pr[\mathbf{Y} = \mathbf{y} | X = x] \cdot \log \left( \frac{\Pr[\mathbf{Y} = \mathbf{y} | X = x]}{\Pr[\mathbf{Y} = \mathbf{y}]} \right) d\mathbf{y}.$$

2. Another solution is to estimate the *multivariate mutual information* which is an attempt to extend the definition of mutual information to more than two variables. It is defined as  $I(X; Y_1; Y_2) = I(Y_1; Y_2) - I(Y_1; Y_2 | X)$ , where:

$$I(Y_1; Y_2 | X) = \sum_{x \in \mathcal{X}} \Pr[X = x] \left[ I(Y_1; Y_2 | X = x) \right].$$

3. Eventually, it is possible to use the *total correlation* [36]  $C(X, Y_1, Y_2) = H[X] + H[Y_1] + H[Y_2] - H[X, Y_1, Y_2]$  that can also be written as:

$$C(X, Y_1, Y_2) = \sum_{x \in \mathcal{X}} \int_{\mathbf{y}^2} \Pr[x, y_1, y_2] \cdot \log \left( \frac{\Pr[x, y_1, y_2]}{\Pr[x] \cdot \Pr[y_1] \cdot \Pr[y_2]} \right) d\mathbf{y}.$$

These definitions can be related by standard information theoretic relations:

$$\begin{aligned} I(X; (Y_1, Y_2)) &= I(X; Y_1) + I(X; Y_2) - I(X; Y_1; Y_2), \\ C(X, Y_1, Y_2) &= I(X; Y_1) + I(X; Y_2) + I(Y_1; Y_2) - 2 \cdot I(X; Y_1; Y_2). \end{aligned}$$

Depending on the applications, one or another definition will be preferable. For example, if a good masking scheme is used, the variables  $X$  and  $Y_1$  are independent (and so are  $X$  and  $Y_2$ ). Hence, the contribution of the terms  $I(X; Y_1)$  and  $I(X; Y_2)$  will not be useful in this context. As for the univariate case, these equations all lead to an asymptotically successful key recovery. But the convergence towards the correct subkey may differ in practice, in function of the physical



leakages and distinguisher selected by the adversary. In every case, detecting the multivariate dependencies requires to estimate one multivariate probability density function for each modeled value. The next section briefly discusses the adaptation of the histogram and kernel methods for this purpose.

#### 4.4 Multivariate PDF estimation tools

**Histograms.** The histogram method is straightforwardly extended by partitioning the  $d$ -dimensional sample space into bins of equal width along a given coordinate (or orthotopes). For each bin denoted as  $b(i_1, \dots, i_d)$ , the probability is estimated as:

$$\hat{\text{Pr}}[\mathbf{y} \in b(i_1, \dots, i_d)] = \frac{\#b(i_1, \dots, i_d)}{q},$$

where  $(i_1, \dots, i_d)$  denotes the index of a bin,  $\#b(i_1, \dots, i_d)$  the number of samples in bin  $b(i_1, \dots, i_d)$  and  $q$  the total number of samples available.

**Kernels.** The kernel density estimation method cannot be directly extended to the multivariate case in general. But different assumptions allow overcoming this issue. For example, one can use Gaussian kernels and apply the formula:

$$\hat{\text{Pr}}[\mathbf{Y} = \mathbf{y}] = \frac{1}{q(2\pi)^{d/2}|\Sigma_{\mathbf{Y}\mathbf{Y}}|^{1/2}} \sum_{i=1}^q \exp\left(-\frac{1}{2h^2}(\mathbf{y} - \mathbf{y}^i)' \Sigma_{\mathbf{Y}\mathbf{Y}}^{-1}(\mathbf{y} - \mathbf{y}^i)\right),$$

where  $\Sigma_{\mathbf{Y}\mathbf{Y}}$  is the leakage covariance matrix. Alternatively, if we additionally make the hypothesis that the coordinates of the multivariate distribution are pairwise independent (which is the case for masking schemes), it is then possible to use the product of any kernel defined as:

$$\hat{\text{Pr}}[\mathbf{Y} = \mathbf{y}] = \frac{1}{q} \sum_{i=1}^q \left( \prod_{j=1}^d K\left(\frac{\mathbf{y}_j - \mathbf{y}_j^i}{h_j}\right) \right).$$

Assuming a normal kernel and a normal distribution for the leakages with  $\Sigma_{\mathbf{Y}\mathbf{Y}}$  diagonal, the rule-of-thumb for the bandwidth of Section 3.2 becomes (see [26]):

$$h_j^* = \hat{\sigma}_j q^{-1/(d+4)},$$

with  $\hat{\sigma}_j$  denoting the diagonal elements of the leakage covariance matrix. Alternative bandwidth selection rules include the one of Hall *et al.* [11].

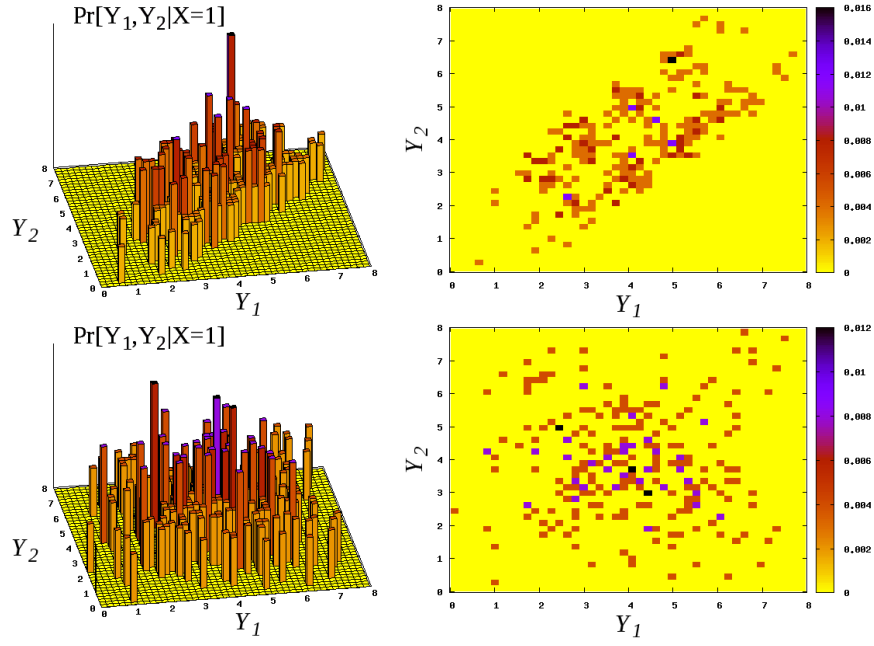
**Example.** In order to illustrate the estimation methods described above in the context of MIA against masked implementations, we applied them on simulated leakage traces corresponding to a masked AES S-box. We generated the distribution of the pairs  $(y_1, y_2)$  such that  $y_1 = H_w(\mathbf{S}(p \oplus k) \oplus m) + n_1$  and  $y_2 = H_w(m) + n_2$  for  $\mathbf{S}$  being the AES S-box,  $m$  being a random mask and

the  $n_i$ 's being (independent) random Gaussian noises with standard deviation 0.3. We then applied both histogram and kernel density methods to estimate the probability density functions  $\Pr[\mathbf{Y}|X=x]$  for the Hamming weight model  $X = H_w(S(P \oplus j))$  and pairs  $(x, j) \in \mathcal{X} \times \mathcal{K}$ . Figures 7 and 8 show the obtained pdfs when  $X = 1$  for the correct subkey guess (upper part of the figure) and for a wrong subkey guess (lower part of the figure). As expected, we observe that the densities obtained for the correct key guess are less dissipated than for the wrong key guess, which seem to randomly sample the leakage space.

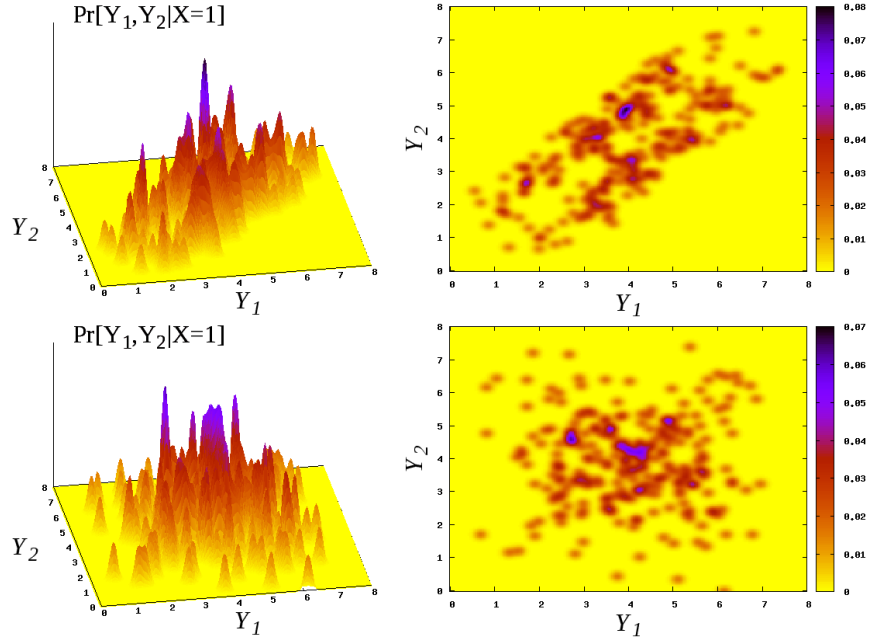
#### 4.5 Experiments

In order to confirm the previous theory and compare the various methods to implement a second-order MIA, we performed different experiments against the Boolean masking scheme of [10] implemented for the DES. For simplicity, our descriptions focus on a representative step of the encryption that consists of a single masked S-box lookup (we targeted the first DES S-box  $S_1$ ). As for Figures 7 and 8, each measurement trace is composed of (*i.e.* reduced to) two leakages samples:  $Y_2$  which is generated by the random mask, and  $Y_1$  which is generated by the masked output of the S-box. This masking scheme ensures that the circuit never processes unmasked intermediate values. Eventually, our experimental setup monitors a smart card embedding an 8-bit RISC microcontroller of which the bus was reset to zero before and after each memory access. The power measurements represent the voltage drop across a  $10\Omega$  resistor inserted in the circuit ground. We used this setup to analyze the following scenarios.

1. **Different distinguishers.** We applied the two first multivariate MIA described in Section 4.3. In other words, we computed the mutual information  $\hat{I}(X; \mathbf{Y})$  and the multivariate mutual information  $\hat{I}(X; Y_1; Y_2)$ . In addition and for the reference, we also applied a second-order attack using the normalized product combining function described in Section 4.2.
2. **Different leakage models.** We considered both a Hamming weight leakage model  $X = H_w(S_1(P' \oplus j))$  and an identity leakage model making no assumptions at all on the leakages  $X = S_1(P' \oplus j)$ , where  $P'$  are the six known plaintext bits entering  $S_1$  in the DES implementation.
3. **Different pdf estimation tools.** As for the univariate case, we directly estimated the probability mass function of the discrete predictions  $X$ . And the pdfs of the leakages  $\mathbf{Y}$  were estimated both with histograms and with kernels. When using histograms, the number of bins is chosen according to the size of the model space  $\mathcal{X}$ . More precisely, we used five bins when assuming a Hamming weight leakage model and sixteen bins for the identity leakage model. In the case of kernel density estimation, a Gaussian kernel is used and the bandwidth is selected according to Hall's rule.

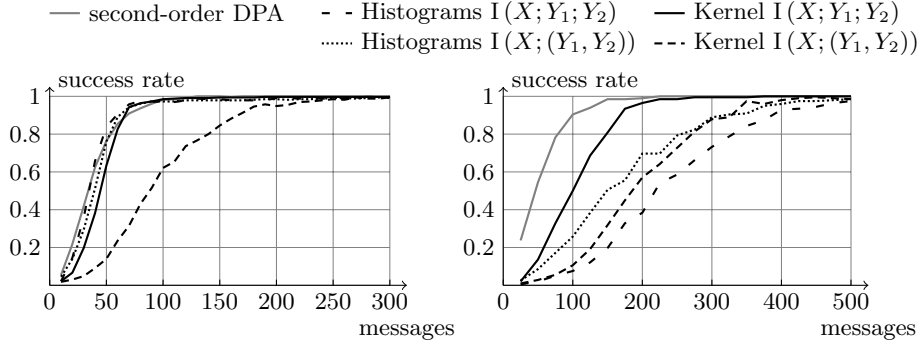


**Fig. 7.** Histogram method in the context of a second-order attack against an 8-bit masked S-box, for the correct (top) and a wrong (bottom) subkey hypothesis.



**Fig. 8.** Kernel method in the context of a second-order attack against an 8-bit masked S-box, for the correct (top) and a wrong (bottom) subkey hypothesis.

4. **Different noise levels.** Eventually, we considered two noise levels. In the first one, the device simply computes the masked 4-bit S-box outputs, leaving the 4 remaining bits in the bus datapath stuck to zero. This scenario gives rise to measurements with very little noise, as acknowledged by a correlation coefficient of approximately 0.99 when attacking an unprotected S-box. In the second scenario, we randomly flip the remaining bits on the bus, giving rise to 4 bits of independent, so-called algorithmic, noise on  $Y_1$  and  $Y_2$ .



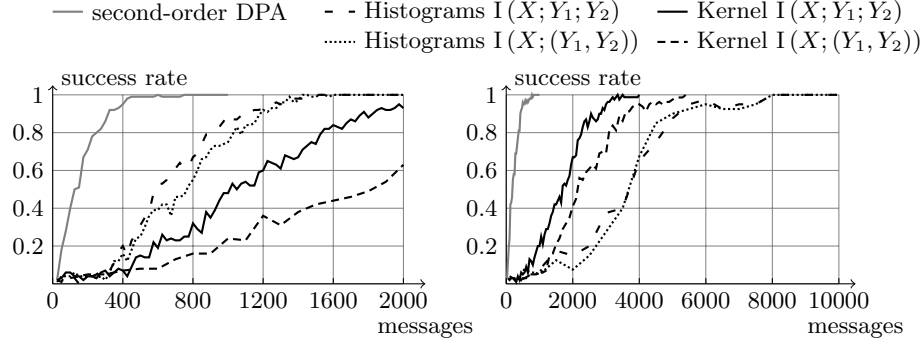
**Fig. 9.** Low noise scenario: success rate of attacks against a masked implementation of the first DES S-box, for Hamming weight (left) and identity (right) leakage models.

The results of our experiments for these different contexts are shown in Figures 9 and 10 from which we can extract the following observations.

First and quite naturally, a good leakage model helps all the attacks under investigation. In particular, since the target device of our experiments closely follows a Hamming weight leakage model, this assumption always improves the success rate when compared to the identity leakage model. Note that the impact of this good leakage model is weaker in the case of a correlation attack, since the correlation between 4-bit values and their Hamming weights is high ( $\approx 0.81$ ).

Second, the pdf estimation tools have a strong impact on MIA’s efficiency, confirming the univariate experiments of Section 3.3. Interestingly, the amount of noise in the leakages can significantly influence which estimation tool is best. For example, histograms perform well in the low noise scenario with a Hamming weight leakage model. The reason is that the leakage probability densities then behave like a Dirac comb that is well approximated by a 5-bin histogram. When moving to an identity leakage model and increasing the noise, this advantage vanishes, as witnessed by the better efficiency of the kernel estimation method in this case. In other words, kernel-MIA gains interest compared to histogram-MIA, when the amount of noise in the physical leakages increases.

In our experiments, the second-order DPA using the correlation coefficient and a normalized product combining function is the most efficient. This observation can again be explained by the fact that the investigated leakages have



**Fig. 10.** High noise scenario: success rate of attacks against a masked implementation of the first DES S-box, for Hamming weight (left) and identity (right) leakage models.

strong linear dependencies with the Hamming weights of the target intermediated values. It is also well in line with the corrected simulated experiments in [20] that we give in Appendix A. Note that the efficiency of MIA compared to 2nd-order DPA gets worse when increasing the noise in our case.

Third, the best methods for applying MIA to two-dimensional leakages depends on the physical leakages, measurement noise, models and pdf estimation method used by the adversary. But they do not exhibit strongly different efficiencies. Overall, these experiments also follow the analysis in [30], which shows that the efficiency of non-profiled (second-order) side-channel distinguishers is difficult to predict and highly dependent on the implementation context.

#### 4.6 Discussion

**Generalization to higher-orders.** The previous descriptions and experiments were given for the example of a second-order attack. But the application of MIA naturally extends to attacks of any order (as the normalized product combining function). A third-order example is given in [9]. Among the three methods for applying MIA to multivariate contexts in Section 4.3, the first and third ones directly generalize to more dimensions. Computing the multivariate mutual information can also be done, by using the following recursion:

$$I(X; Y_1; Y_2; \dots; Y_d) = I(X; Y_1; Y_2; \dots; Y_{d-1}) - I(X; Y_1; Y_2; \dots; Y_{d-1} | Y_d).$$

**Choice of parameters.** Finally, choosing the right leakage model, number of bins or bandwidth impacts also higher-order MIA, just as discussed in Section 3.4 for the univariate case. Additionally, MIA gets rid of the combining function which removes the need to carefully select it. Directly characterizing the dependencies of the joint leakage distributions is also a better choice from an information theoretic point of view. As a consequence, MIA appears as a promising approach for dealing with any advanced application in which successfully attacking protected devices requires to process high dimensional leakages.

## 5 Conclusion and open problems

A comprehensive treatment of MIA was presented. It compiles a theoretical justification for this new distinguisher and its application to practically important scenarios. Namely, we considered both first-order side-channel attacks against an unprotected implementation and second-order side-channel attacks against a masked implementation. Our results put forward the generic nature of MIA and its potential to apply to a large range of cryptographic devices. They also raise several open questions, mainly related to the exploitation of this genericness. Since the application of MIA implies to select a number of parameters, the best selection of those parameters is an interesting scope for further research. For example, our results show that histograms are quite efficient to characterize low noise measurements while kernel density estimation better deals with noisy situations. But plugging in other efficient probability density estimation tools in the analysis of side-channel leakages, potentially taking advantage of certain reasonable assumptions, may lead to an increased efficiency for MIA. Finally, it would be interesting to apply MIA to implementation contexts where its genericness could be fully exploited, *e.g.* devices protected with logic styles that do not exhibit a simple (Hamming weight or distance) leakage model.

## References

1. S. Chari, C. S. Jutla, J. R. Rao, P. Rohatgi, *Towards Sound Approaches to Counteract Power-Analysis Attacks*, in the proceedings of CRYPTO 1999, Lecture Notes in Computer Science, vol. 1666, pp. 398-412, Santa Barbara, California, August 1999.
2. E. Brier, C. Clavier, F. Olivier, *Correlation Power Analysis with a Leakage Model*, in the proceedings of CHES 2004, Lecture Notes in Computer Science, vol. 3156, pp. 16-29, Boston, Massachusetts, USA, August 2004.
3. S. Chari, J. R. Rao, P. Rohatgi, *Template Attacks*, in the proceedings of CHES 2002, Lecture Notes in Computer Science, vol. 2523, pp. 13-28, San Francisco, California, USA, August 2002.
4. T.M. Cover, J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, 1991.
5. DPA Contest 2008/2009. <http://www.dpacontest.org/>.
6. D. Freedman, P. Diaconis, *On the Histogram as a Density Estimator*, in Probability Theory and Related Fields, vol. 57, num. 4, pp. 453-476, December 1981.
7. K. Gandolfi, C. Mourtel, F. Olivier, *Electromagnetic Analysis: Concrete Results*, in the proceedings of CHES 2001, Lecture Notes in Computer Science, vol. 2162, pp. 251-261, Paris, France, May 2001.
8. B. Gierlichs, L. Batina, P. Tuyls, B. Preneel, *Mutual Information Analysis - A Generic Side-Channel Distinguisher*, in the proceedings of CHES 2008, Lecture Notes in Computer Science, vol. 5154, pp. 426-442, Washington DC, USA, August 2008.
9. B. Gierlichs, L. Batina, B. Preneel, I. Verbauwhede, *Revisiting Higher-Order DPA Attacks: Multivariate Mutual Information Analysis*, in the proceedings of CT-RSA 2010, Lecture Notes in Computer Science, vol. 5985, pp. 221-234, San Francisco, CA, USA, March 2010.

10. L. Goubin, J. Patarin, *DES and Differential Power Analysis*, in the proceedings of CHES 1999, Lecture Notes in Computer Science, vol. 1717, pp. 158-172, Worcester, MA, USA, August 1999.
11. P. Hall, S. J. Sheather, M.C. Jones, J.S. Marron, *On Optimal Data-Based Bandwidth Selection in Kernel Density Estimation*, Biometrika, vol 78, pp 263-270, 1991.
12. W. Härdle, *Smoothing Techniques: With Implementation in S.*, Springer series in statistics, December 1990.
13. K. H. Knuth, *Optimal Data-Based Binning for Histograms*, <http://arxiv.org/abs/physics/0605197>, May 2006.
14. P. Kocher, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems*, in the proceedings of Crypto 1996, Lecture Notes in Computer Science, vol. 1109, pp. 104-113, Santa-Barbara, CA, USA, August 1996.
15. P. Kocher, J. Jaffe, B. Jun, *Differential Power Analysis*, in the proceedings of Crypto 1999, Lecture Notes in Computer Science, vol. 1666, pp. 398-412, Santa-Barbara, CA, USA, August 1999.
16. K. Lemke, C. Paar, *Gaussian Mixture Models for Higher-Order Side-Channel Analysis*, in the proceedings of CHES 2007, Lecture Notes in Computer Science, vol. 4227, pp. 14-27, Vienna, Austria, September 2007.
17. S. Mangard, E. Oswald, T. Popp, *Power Analysis Attacks*, Springer, 2007.
18. T. S. Messerges, *Using Second-Order Power Analysis to Attack DPA Resistant Software*, in the proceedings of CHES 2000, Lecture Notes in Computer Science, vol. 1965, pp. 238-251, Worcester, Massachusetts, USA, August 2000.
19. A. Moradi, N. Mousavi, C. Paar, M. Salmasizadeh, *A Comparative Study of Mutual Information Analysis under a Gaussian Assumption*, in the proceedings of WISA 2009, Lecture Notes in Computer Science, vol. 5932, pp. 193-205, Busan, Korea, August 2009.
20. E. Prouff, M. Rivain, *Theoretical and Practical Aspects of Mutual Information Based Side-Channel Analysis*, in the proceedings of ACNS 2009, Lecture Notes in Computer Science, vol. 5536, pp. 499-518, Paris, France, June 2009.
21. E. Prouff, M. Rivain, R. Bévan, *Statistical Analysis of Second-Order DPA*, in IEEE Transactions on Computers, vol. 58, num. 6, pp. 799-811, June 2009.
22. E. Prouff, R. McEvoy, *First-Order Side-Channel Attacks on the Permutation Tables Countermeasure*, in the proceedings of CHES 2009, Lecture Notes in Computer Science, vol. 5747, pp. 81-96, Lausanne, Switzerland, September 2009.
23. J.-J. Quisquater, D. Samyde, *ElectroMagnetic Analysis (EMA): Measures and Countermeasures for Smart Cards*, in the proceedings of eSmart 2001, Lecture Notes in Computer Science, vol. 2140, pp. 200-210, Cannes, France, September 2001.
24. M. Rivain, E. Dottax, E. Prouff, *Block Ciphers Implementations Provably Secure Against Second-Order Side-Channel Analysis*, in the proceedings of FSE 2008, Lecture Notes in Computer Science, vol. 5086, pp. 127-143, Lausanne, Switzerland, February 2008.
25. D. W. Scott, *On Optimal and Data-Based Histograms*, in Biometrika, vol. 66, num. 3, pp. 605-610, December 1979.
26. D. W. Scott, S. R. Sain, *Multi-dimensional Density Estimation*, Handbook of Statistics, vol. 24: Data Mining and Data Visualization, North-Holland Publishing, 2004.
27. B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall - CRC Press, April 1986.

28. F.-X. Standaert, T. G. Malkin, M. Yung, *A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks*, in the proceedings of Eurocrypt 2009, Lecture Notes in Computer Science, vol. 5479, pp. 443-461, Cologne, Germany, April 2009, extended version available on the Cryptology ePrint Archive, Report 2006/139, <http://eprint.iacr.org/2006/139>.
29. F.-X. Standaert, B. Gierlichs, I. Verbauwhede, *Partition vs. Comparison Side-Channel Distinguishers: An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks*, in the proceedings of ICISC 2008, Lecture Notes in Computer Science, vol. 5461, pp. 253-267, Seoul, Korea, December 2008.
30. F.-X. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, S. Mangard, *The World is Not Enough: Another Look on Second-Order DPA*, Cryptology ePrint Archive, Report 2010/180, <http://eprint.iacr.org/2010/180>.
31. R. A. Tapia, J. R. Thompson, *Nonparametric Density Estimation*, John Hopkins University Press, Baltimore, Maryland, 1978.
32. B. A. Turlach, *Bandwidth Selection in Kernel Density Estimation: a Review*, in *CORE and Institut de Statistique*, 1993.
33. N. Veyrat-Charvillon, F.-X. Standaert, *Mutual Information Analysis: How, When and Why?*, in the proceedings of CHES 2009, Lecture Notes in Computer Science, vol. 5747, pp. 429-443, Lausanne, Switzerland, September 2009.
34. K. Tiri, M. Akmal, I. Verbauwhede, *A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand DPA on Smart Cards*, in the proceedings of ESSCIRC 2003, Estoril, Portugal, September 2003.
35. L. Batina, B. Gierlichs, K. Lemke-Rust, *Differential Cluster Analysis*, in the proceedings of CHES 2009, Lecture Notes in Computer Science, vol. 5747, pp. 112-127, Lausanne, Switzerland, September 2009.
36. S. Watanabe, *Information Theoretical Analysis of Multivariate Correlation*, in IBM Journal of Research and Development, vol 4, pp. 66-82, 1960.
37. S. Aumonier, *Generalized Correlation Power Analysis*, in the proceedings of the ECRYPT Workshop on Tools For Cryptanalysis, Kraków, Poland, September 2007.

## A Corrected results for the 2nd-order attacks in [20]

**Table 2.** Second-order attack on DES S-box – Number of measurements required to achieve a success rate of 90% according to the noise standard deviation  $\sigma$ .

Attack \ $\sigma$	0.5	1	2	5	7	10
2O-CPA ( $\phi = H_w$ , abs. difference)	300	800	5000	200000	$10^6+$	$10^6+$
2O-CPA ( $\phi = H_w$ , norm. product)	300	400	3000	70000	300000	$10^6+$
2O-MIA <sub>H</sub> ( $\phi = \text{Id}$ , Scott's Rule)	1200	7000	75000	$10^6+$	$10^6+$	$10^6+$
2O-MIA <sub>H</sub> ( $\phi = \text{Id}$ , Rule in [8])	1800	7000	40000	1000000	$10^6+$	$10^6+$
2O-MIA <sub>K</sub> ( $\phi = \text{Id}$ )	600	2500	25000	600000	$10^6+$	$10^6+$
2O-MIA <sub>H</sub> ( $\phi = H_w$ , Scott's Rule)	600	2700	34000	$10^6+$	$10^6+$	$10^6+$
2O-MIA <sub>H</sub> ( $\phi = H_w$ , Rule in [8])	350	1300	9000	350000	$10^6+$	$10^6+$
2O-MIA <sub>K</sub> ( $\phi = H_w$ )	300	1300	9000	n.a.	n.a.	n.a.