Efficient profiled attacks on masking schemes

Liran Lerman and Olivier Markowitch

***Abstract*—Side-channel adversaries represent real-world threats against (certified and uncertified) cryptographic devices. Masking schemes represent prevailing countermeasures to reduce the success probabilities of side-channel attacks. However, masking schemes increase the implementation cost in term of power consumption, clock cycles, and random number generation. Investigation of tools evaluating the degree of resilience of cryptographic devices using masking (against side-channel attacks) represents an important aspect in certification procedures (e.g., Common Criteria, FIPS 140-2 and EMVco). Several side-channel evaluation techniques exist such as template attacks and machine learning based attacks. In this paper, we formalise results obtained in side-channel attacks community when targeting masked implementations. We report theoretical as well as practical results of parametric and non-parametric side-channel attacks on masking schemes. The theoretical part reports results based on a simulation of the execution of software devices while the practical part focuses on actual leakages measured during the execution of a software implementation in three different contexts (that contain different levels of noise).**

***Index Terms*—Side-channel analysis, Masking, Template attacks, Kernel density, Random forests.**

I. Introduction

Side-channel attacks exploit unintentional leakages measured on cryptographic devices in order to recover sensitive information such as the secret key. In order to protect the implementations against physical attacks (and to prevent information leakage), the designers employ (among others) masking techniques that split each sensitive information (that depends on the secret key) in d uniformly distributed variables (called shares). The masking schemes leak no information on the sensitive variables whenever the adversaries exploit points in the leakages associated to strictly less than d different shares in order to recover the sensitive information. As a result, the masking schemes force the adversaries to target all the shares at the same time, which leads theoretically to an increase of the number of required leakages (during the attack step) exponentially in the number of shares [14] when the amount of noise in the leakages is sufficiently high [29]. In a practical point of view, the security level provided by sound masked implementations can lead to unsecure executions due to several practical phenomena such as low physical noise levels [29], transition-based leakages [9] and couplings between the shares [2]. Several approaches exist in order to evaluate the degree of resilience of implementations against side-channel adversaries such as formal security proofs (e.g., the probing model of Ishai *et al.* [14]), leakage detection tests (e.g., Cryptography Research's non-specific fixed vs. random T-test [8], [12]) and (actual) physical attacks (e.g., the profiled attacks [7]). We opt for the last security evaluation approach in order to take into account all the defaults that impact the degree of resilience of cryptographic devices. It is worth to note that evaluators can also analyse abstract implementations with actual physical attacks.

Contact authors: Liran Lerman liran@lerman.be

The Bayes classifier represents the optimal attack that exploits all the information available on the key in the leakages. In practice, evaluators of cryptographic devices never exploit such approach due to the lack of time and leakages in order to build the method as well as the lack of knowledge on the phenomenon generating the leakages. In this paper, we promote efficient approaches exploiting (i) a smaller running time and (ii) less knowledge on the target device compared to the Bayes classifier. Following recent papers in the side-channel community discussing machine learning models, we focus on (learning) models called random forests, template attacks and profiled attacks based on the kernel density estimation method [5], [19], [20], [22]. The main issue of profiled attacks lies in the exploitation of leakages containing a large number of dimensions. Although the high dimensionality of leakages is common when evaluating masking implementations, the curse of dimensionality stipulates (and we demonstrate this phenomenon in this paper) that the difficulty to face such settings becomes more difficult as the dimensionality of leakages increases. Note also that this high dimensionality context also provides benefits to the side-channel adversaries: the more the number of dimensions, the larger the quantity of information available in the leakages on the target value. Similarly, Battistello *et al.* report that the more the number of shares, the more the quantity of information available to the adversary [3].

This paper complements our previous works on unprotected implementations [20]. More precisely, in this paper, first we aim to formalise results obtained in side-channel attacks when targeting masked implementations. We explain the results obtained by parametric (called template attacks) and non-parametric profiled attacks (called profiled attacks based on kernel density estimation method). Second, compared to the state-of-the art tools to evaluate masking schemes, we report a new more efficient approach (based on a machine learning algorithm called random forests) that leads to a higher success probability to extract the secret key from software masking implementations.

Similarly to our work in [20], we focus on analytical and practical results obtained on simulated measurements on a software implementation. We base our theoretical results on the bias-variance theory (presented by Lerman *et al.* [18] in side-channel attacks) and on the estimation/assumption errors (introduced by Durvaux *et al.* [10]) in order to discover what impacts the success probability to retrieve the secret key. Furthermore, we report the performances of profiled attacks against an 8-bit Atmel XMEGA target device using using

three kinds of actual measurements: (i) actual leakages with a low noise level, (ii) actual leakages with added noise and (iii) misaligned actual leakages. The reported results confirm the results obtained with the simulated leakages.

II. Background on side-channel attacks

A. Side-channel and masking

We consider side-channel adversaries targeting sensitive value y representing the output of a function $\mathsf{f}_k(p)$ where k is a part of the target key (e.g., one byte of the key) and p is a part of a known information by the adversary (e.g., one byte of the plaintext). The function f represents for instance the output of a Sbox function (e.g., $\mathsf{f}_k(p) = \mathsf{Sbox}(p \oplus k)$).

Let ${}^{j}\boldsymbol{T}_y$ be the j-th leakage (also known as trace) measured when the device manipulates the target value y. Let also $\boldsymbol{T}$ be a leakage associated to an unknown target value y. In the following, we represent each leakage with a vector of real values measured at different instants on the analysed device. We denote ${}^{j}_{t}T_y$ the j-th leakage (associated to the target value y) measured at time t such that:

$$
\begin{aligned}
{}^{j}_{t}T_y &= {}_{t}\mathsf{L}(y) + {}^{j}_{t}\epsilon_y, && (1)\\
&= {}_{t}\mathsf{L}(\mathsf{f}_k(p)) + {}^{j}_{t}\epsilon_y, && (2)
\end{aligned}
$$

where ${}^{j}_{t}\epsilon_y \in \mathbb{R}$ is the physical noise of the trace ${}^{j}_{t}T_y$ (i.e., an additive physical noise assumption defined by Mangard *et al.* [23]) following a Gaussian distribution with zero mean, and ${}_{t}\mathsf{L}$ is the deterministic leakage function at time t.

In order to protect the implementations against physical attacks, the designers employ (among others) masking techniques that split each sensitive information y (that depends on the secret key) in d uniformly distributed variables (called shares), denoted $\boldsymbol{x} = [x_1, ..., x_d] \in \mathcal{X}$, such that $y = \sum_{i=1}^{d} x_i$ (where $\sum$ represents the method combining shares) [6], [13]. Typically, the shares $[x_2, ..., x_d]$ represent $d-1$ uniformly distributed random values (called the masks) while x_1 equals to $y + \sum_{i=2}^{d} x_i$. In the following, $\mathcal{X}_y$ denotes the space of the vectors of shares in which $x_1 = y + \sum_{i=2}^{d} x_i$.

We analyse serial implementations manipulating one share per cycle and each cycle is associated to n_s points in the leakage (i.e., ${}^{j}\boldsymbol{T}_y$ contains $n_s \times d$ points in total). This follows the usual leakages measured on software implementations (see for example the paper of Barthe *et al.* [2]). The masking schemes leak no information on the sensitive variable y whenever the adversaries combine points associated to strictly less than d different shares.

B. Profiled attacks

This section describes one optimal/ideal *theoretical* profiled attack (called Bayes classifier) and three practical profiled attacks (called template attacks, profiled attacks based on kernel density estimation method and random forests). The optimality of a profiled attack depends on the choice of the criterion evaluating the attack. We characterise the effectiveness of a key-recovery side channel adversary by its *error rate*. The error rate represents the probability that the profiled attack outputs the wrong key value as the most likely key values.

Practical profiled attacks represent efficient attacks thanks to a learning step (also known as a profiling step) on a device similar to the target device. More precisely, these approaches build a classifier $\mathsf{A}(\mathcal{T}_{\mathrm{PS}}, \mathcal{T}_{\mathrm{AS}})$ that: during the *profiling step*, it estimates a parameter θ with a set of leakages (called *profiling set* and denoted $\mathcal{T}_{\mathrm{PS}}$) containing N_p (profiling) traces per target value, and during the *attack step*, it returns the extracted secret key k from a set of attack leakages $\mathcal{T}_{\mathrm{AS}}$ (called *attacking set*) measured on the target device using a constant secret key. In the following, and as already assumed by Lemke-Rust *et al.* [15], all exploited practical profiled attacks (i.e., template attacks, profiled attacks based on kernel density estimation method and random forests) do not use the mask values during the profiling step in order to reduce the time complexity of the profiling step and of the attack step (which leads to efficient profiled attacks on masking schemes). This is also useful in contexts where the adversary does not know the mask values during the profiling step.

In a general case, when fixing the number of measurements in the attack sets, the error rate of a profiled attack depends on the collected profiling set. In our experiments, we remove the mentioned dependency by averaging over profiling sets and attack sets.

1) Bayes classifier: The *Bayes classifier* denotes the best possible theoretical attack which practical attacks can reach. More formally, let $\mathsf{A}_b(\cdot)$ be the Bayes classifier that takes as input a set of attack traces $\mathcal{T}_{\mathrm{AS}} = \left\{{}^{1}\boldsymbol{T}, {}^{2}\boldsymbol{T}, ..., {}^{N_a}\boldsymbol{T}\right\}$ (where ${}^{j}\boldsymbol{T}$ represents the j-th leakage in the attack set $\mathcal{T}_{\mathrm{AS}}$ and N_a is the number of attack leakages). The function $\mathsf{A}_b(\cdot)$ represents a classifier that minimises the probability of misclassification (also known as the error rate), i.e.:

$$
\begin{aligned}
\mathsf{A}_b(\mathcal{T}_{\mathrm{AS}}) &\in \underset{k \in \mathcal{K}}{\operatorname{argmax}} \Pr\left[k \mid \mathcal{T}_{\mathrm{AS}}\right] && (3)\\
&= \underset{k \in \mathcal{K}}{\operatorname{argmax}} \Pr\left[\mathcal{T}_{\mathrm{AS}} \mid k\right] \times \Pr\left[k\right], && (4)\\
&= \underset{k \in \mathcal{K}}{\operatorname{argmax}} \prod_{j=1}^{N_a} \Pr\left[{}^{j}\boldsymbol{T} \mid y = \mathsf{f}_k(p_j)\right] && (5)\\
&\qquad \times \Pr\left[y = \mathsf{f}_k(p_j)\right],
\end{aligned}
$$

where p_j is the j-th known plaintext used by the device when the adversary measured the j-th attack leakage ${}^{j}\boldsymbol{T}$, and $\Pr\left[{}^{j}\boldsymbol{T} \mid y\right] = \sum_{\boldsymbol{x} \in \mathcal{X}_y} \Pr\left[{}^{j}\boldsymbol{T} \mid \boldsymbol{x}\right] \times \Pr\left[\boldsymbol{x}\right]$. In the following, we assume that $\Pr\left[{}^{j}\boldsymbol{T} \mid \boldsymbol{x}\right]$ follows a multivariate Gaussian distribution and, as a result, $\Pr\left[{}^{j}\boldsymbol{T}_y \mid y\right]$ follows a finite linear combination (also known as a mixture) of multivariate Gaussian distribution (as already assumed in several papers published in the literature [6], [25], [28]).

Two approaches exist in order to estimate Equation 5: parametric and non-parametric approaches. The parametric approaches assume that the evaluators know the structure of the probability density functions (p.d.f.) and require the estimation of the parameters associated to the p.d.f. The non-parametric approaches exploit no prior knowledge on the structure of the probability density functions.

2) Template attacks: (Unimodal parametric Gaussian) *Template Attacks* (TA) [7] estimate Equation 5 by assuming that $\Pr\left[{}^{j}\boldsymbol{T}_y \mid y\right]$ follows a Gaussian distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}_y, \hat{\boldsymbol{\Sigma}}_y)$ for

each value y where $\hat{\boldsymbol{\mu}}_y$ and $\hat{\boldsymbol{\Sigma}}_y$ are the sample mean (also known as the first order moment) and the sample covariance matrix (also known as the second order moment) of the leakages associated to y. In what follows, we assume that the adversaries have no information on the shares during the profiling step leading to a dependency between the second-order moment $\boldsymbol{\Sigma}_y$ and the sensitive information y when exploiting masking schemes based on two shares per target value. Note that the adversary can know the mask values during the profiling step and still not use them in order to speed up the attack phase by exploiting less templates (i.e., by estimating the parameters of the p.d.f. providing $\Pr\left[{}^{j}\boldsymbol{T}_y \mid y\right]$ with one unimodal Gaussian distribution instead of estimating all the parameters associated to the 2^{d-1} p.d.f.).

3) Profiled attacks based on non-parametric kernel density estimation method: Profiled attacks based on non-parametric Kernel Density Estimation method (KDE) estimate the p.d.f. (associated to $\Pr\left[{}^{j}\boldsymbol{T}_y \mid y\right]$) by placing a (Kernel) function (denoted $\mathsf{K}_{\boldsymbol{\Sigma}}(\cdot,\cdot)$) at each leakage (associated to y). More precisely, the estimated p.d.f. associated to $\Pr\left[\boldsymbol{T} \mid y\right]$ equals to:

$$\hat{\mathsf{f}}_{\boldsymbol{\Sigma}}\left(\boldsymbol{T} \mid y\right) = \frac{1}{N_p}\sum_{j=1}^{N_p}\mathsf{K}_{\boldsymbol{\Sigma}}(\boldsymbol{T}, {}^{j}\boldsymbol{T}_y), \tag{6}$$

where $\boldsymbol{\Sigma}$ is the Kernel bandwidth and represents the smoothing parameter. In our experiments, we exploit the Gaussian Kernel function in which $\mathsf{K}_{\boldsymbol{\Sigma}}(\boldsymbol{T},\boldsymbol{\mu}) = \frac{1}{\sqrt{(2\pi)^{n_s\times d}|\boldsymbol{\Sigma}|}}e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{T}-\boldsymbol{\mu})}$. Note that other Kernel functions exist (such as the uniform, the triangular and the Epanechnikov Kernel functions) but this choice impacts only in a minor way the quality of the estimated p.d.f. (compared to the choice of the Kernel bandwidth value) [27]. We use the Scott's rule (that is widely recommended by introductory statistics text and proposed by the used statistical packages as default [1]) in order to estimate $\boldsymbol{\Sigma}$, which is equal to:

$$\begin{cases}\sqrt{\boldsymbol{\Sigma}_{ij}} = N_p^{\frac{-1}{n_s\times d+4}}\sigma_i & i=j\\ \sqrt{\boldsymbol{\Sigma}_{ij}} = 0 & i\neq j\end{cases}, \tag{7}$$

where $\boldsymbol{\Sigma}_{ij}$ represents the entry in the i-th row and j-th column of the matrix $\boldsymbol{\Sigma}$, and σ_i represents the standard deviation of the leakages at instant i.

4) Random forests: The Random Forests (RF) introduced by Breiman can be seen as a collection of classifiers using many decision trees as models [4]. It relies on model averaging. After the profiling phase, RF return the most consensual prediction for a target value through a majority vote among the set of trees. RF are based on three main principles. First, each tree is constructed with a different profiling set by re-sampling (with replacement) the original dataset. Secondly, the nodes of the trees are split using the best time sample among a subset of randomly chosen ones (by contrast to conventional trees where all the time samples are used). The size of this subset was set to the square of the number of time samples (i.e., $\sqrt{n_s \times d}$) as suggested by Breiman. These features allow obtaining decorrelated trees, which improves the accuracy of the resulting RF model. Finally, and unlike conventional decision trees as well, the trees of a RF are fully grown and are not pruned, which possibly leads to overfitting that is reduced by averaging the trees. The main (meta-) parameters of RF are the number of trees. Intuitively, increasing the number of trees reduces the instability of the models. We set this number to 500 in our simulated experiments (reported in Section IV-B and in Section IV-C) and 100 in our actual experiments (reported in Section IV-D), which was sufficient in our experiments in order to show the strength of this model compared to the other classifiers presented previously.

[1]We used the function `gaussian_kde` from the `scipy` python package.

III. Bias-variance decomposition

This section discusses the assumption and the estimation errors of profiled attacks. We base all our practical experiments on the block-cipher PRESENT for efficiency reasons.

A. Assumption error

The assumption error represents the difference between the structure of the target probability density function associated to $\Pr\left[\boldsymbol{T} \mid y\right]$ and the structure of the assumed (by the adversary) probability density function associated to $\hat{\Pr}\left[\boldsymbol{T} \mid y\right]$. The probability $\Pr\left[\boldsymbol{T} \mid y\right]$ follows a multimodal distribution (more precisely a mixture of unimodal Gaussian distributions) since:

$$\Pr\left[\boldsymbol{T} \mid y\right] = \sum_{\boldsymbol{x}\in\mathcal{X}_y}\Pr[\boldsymbol{T} \mid \boldsymbol{x}], \tag{8}$$

where $\Pr[\boldsymbol{T} \mid \boldsymbol{x}]$ follows a Gaussian distribution. However, if each (unimodal) Gaussian distribution associated to $\Pr[\boldsymbol{T} \mid \boldsymbol{x}]$ contains the same mean, then the probability $\Pr\left[\boldsymbol{T} \mid y\right]$ follows a unimodal distribution. More formally, let $\mathsf{f}_y^{\mathsf{m}}(\boldsymbol{T})$ and $\mathsf{f}_y^{\boldsymbol{x}}(\boldsymbol{T})$ be the probability density functions specifying the probability $\Pr\left[\boldsymbol{T} \mid y\right]$ and $\Pr\left[\boldsymbol{T} \mid \boldsymbol{x}\right]$ respectively. Then, the function $\mathsf{f}_y^{\mathsf{m}}(\boldsymbol{T})$ equals to:

$$\mathsf{f}_y^{\mathsf{m}}(\boldsymbol{T}) = \frac{1}{\|\mathcal{X}_y\|}\sum_{\boldsymbol{x}\in\mathcal{X}_y}\mathsf{f}_y^{\boldsymbol{x}}(\boldsymbol{T}), \tag{9}$$

where $\|\mathcal{X}_y\|$ represents the cardinality of $\mathcal{X}_y$, and the function $\mathsf{f}_y^{\boldsymbol{x}}(\boldsymbol{T})$ equals to:

$$\mathsf{f}_y^{\boldsymbol{x}}(\boldsymbol{T}) = \frac{1}{\sqrt{(2\pi)^n\mathsf{det}\left(\boldsymbol{\Sigma}_{\mathsf{m},y}\right)}}e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathsf{m},y}^{-1}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})}, \tag{10}$$

where $\boldsymbol{\mu}_{\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{\mathsf{m},y}$ are the two parameters of the probability density function $\mathsf{f}_y^{\boldsymbol{x}}(\boldsymbol{T})$, and $n = n_s \times d$. Note that we follow the homoscedasticity assumption in which the parameter $\boldsymbol{\Sigma}_{\mathsf{m},y}$ is independent of the set of shares $\boldsymbol{x}$ but depends on the sensitive value y. The definition of $\mathsf{f}_y^{\boldsymbol{x}}(\boldsymbol{T})$ leads to the following:

$$\begin{aligned}&\mathsf{f}_y^{\mathsf{m}}(\boldsymbol{T})\\ &= \frac{1}{\|\mathcal{X}_y\|}\sum_{\boldsymbol{x}\in\mathcal{X}_y}\frac{1}{\sqrt{(2\pi)^n\mathsf{det}\left(\boldsymbol{\Sigma}_{\mathsf{m},y}\right)}}e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathsf{m},y}^{-1}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})},\end{aligned} \tag{11}$$

$$= \frac{1}{\|\mathcal{X}_y\|}\frac{1}{\sqrt{(2\pi)^n\mathsf{det}\left(\boldsymbol{\Sigma}_{\mathsf{m},y}\right)}}\sum_{\boldsymbol{x}\in\mathcal{X}_y}e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathsf{m},y}^{-1}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})}, \tag{12}$$

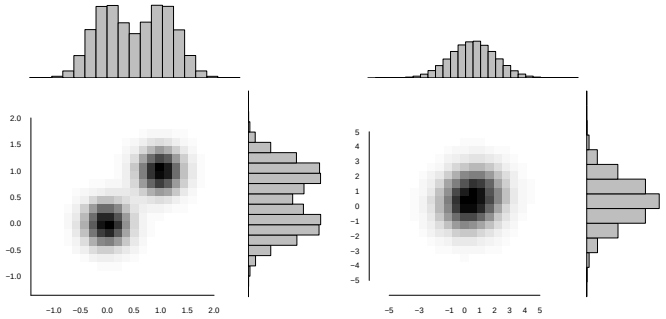


Fig. 1: Bivariate leakage distributions of a single-bit (equals to 0) split into two shares $[x_0, x_1]$. The variance of the physical noise equals to $\sigma^2 = 0.1$ on the left and $\sigma^2 = 2$ on the right.

when all the covariance matrices $\mathbf{\Sigma}_{\mathsf{m},y}$ are the same and $n = n_s \times d$.

Proposition III.1. *The assumption error* [2] *of a unimodal Gaussian TA decreases as a function of the physical noise and as a function of the distance between the values* $\boldsymbol{\mu}_{\boldsymbol{x}}$ *(representing the mean value of leakages associated to* $\boldsymbol{x}$*). In other words, the assumption errors of TA and non-parametric attacks are similar for high noise contexts and when the distance between the values* $\boldsymbol{\mu}_{\boldsymbol{x}}$ *is small.*

The annex reports the proof. In a side-channel point of view, an adversary can use TA instead of non-parametric profiled attacks (such as profiled attacks based on KDE) if the level of physical noise is high or the mean leakages are close enough. In other words, the higher the level of noise in the leakages, the smaller the assumption error of TA.

1) Experiments: We check Proposition III.1 with simulations. Let $y \in \{0,1\}$ be the target value split into two shares $x_0 \in \{0,1\}$ and $x_1 \in \{0,1\}$. Let ${}^j_t T_y$ be the j-th leakage associated to y, which is equal to:

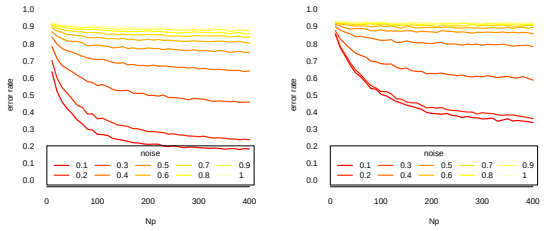
$$
\begin{aligned}
{}^j_t T_y &= {}_t\mathsf{L}\left(y\right) + {}^j_t\epsilon_y, && (13)\\
&= \mathsf{HW}\left(x_t\right) + {}^j_t\epsilon_y, && (14)
\end{aligned}
$$

where $\mathsf{HW}(\cdot)$ represents the Hamming weight function. Figure 1 plots the p.d.f. associated to $\hat{\Pr}\left[\boldsymbol{T} \mid y = 0\right]$ with low and high noise. In a low level of noise, the Mardia's Multivariate Normality Test rejects (with 20 000 leakages) the hypothesis that the leakages follow a Gaussian distribution. However, in a high level of noise, the same test (with 20 000 leakages) does not reject this hypothesis, which confirms the previous theoretical results on the assumption error of TA.

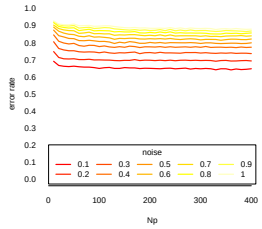
B. Estimation error

The previous section shows that the TA lead to a small assumption error if the leakages contain a high level of physical noise. However, this provides no information on the error rate of TA compared to non-parametric profiled attacks.

[2]We define the assumption error of a profiled attack as the L_2 norm of the difference between the true and the estimated (by the profiled attack) probability density functions. However, the proposition can easily be generalised to other norms without impacting the conclusion.



(a) KDE (b) RF



(c) TA

Fig. 2: Error rate of TA, RF and KDE as a function of the number of profiling leakages per target value (denoted N_p) for several standard deviations of the noise $\sigma \in [0.1; 1]$.

More precisely, Lerman *et al.* show that three terms impact the error rate of profiled attacks: the noise, the bias (also known as the assumption error for TA) and the variance (also known as the estimation error for TA) [18]. The previous section analyses the bias term while this section compares the variance term of profiled attacks with simulations.

1) Experiments: Let $y = \mathsf{f}_k\left(p\right) \in \{0,1\}^4$ be the target value split into two shares $x_0 \in \{0,1\}^4$ and $x_1 \in \{0,1\}^4$. The function $\mathsf{f}_k\left(p\right)$ equals to $\mathsf{Sbox}\left(k \oplus p\right)$ where $\mathsf{Sbox}\left(\cdot\right)$ represents the Sbox of PRESENT, p is the known plaintext, and k is the key. Let ${}^j_t T_y$ be the j-th leakage associated to the t-th share of y, i.e. ${}^j_t T_y = \mathsf{HW}\left(x_t\right) + {}^j_t\epsilon_y$. We generated 10 different TA and KDE (using different profiling sets). We estimated the error rate (to extract the secret k) of profiled attacks with 1 000 attack sets. Each attack set contains $N_a = 5$ attack traces. Figure 2 shows the error rates as a function of the number of profiling leakages for several physical noise levels. This figure highlights that, compared to the error rate of TA, the error rate of non-parametric approaches varies heavier as a function of the number of profiling traces. In other words, TA contain a lower variance term than non-parametric approaches. Therefore, compared to non-parametric approaches, TA require less leakages in the profiling set in order to estimate accurately the set of parameters.

In summary, the error rate of TA is lower than the error rate of non-parametric profiled attacks if (i) the level of noise is high (leading to a small assumption error for TA), (ii) the number of leakages in the profiling set is small (leading to a high estimation error for non-parametric profiled attacks), and (iii) the first and/or the second order moments contain information on y (since TA only estimate these moments) as reported in the next section.

IV. Experimental validation

The previous section highlights that TA can outperform non-parametric approaches if we met the following three conditions in practice: (i) the first and/or the second order moments contain information on y (since unimodal Gaussian TA do not estimate higher order moment), (ii) the leakages contain a high physical noise level (leading TA having small assumption errors), and (iii) the profiling set contains a small number of leakages (leading non-parametric approaches to have high estimation errors). In this section, we verify these (previously presented) results with experiments.

In each experiment, and for each profiled attack, we generated 10 profiling sets (in order to estimate 10 times the parameters of each profiled attack) and 100 attack sets (in order to estimate the error rate of each profiled attack). We collected different profiling and attack sets on the same device by choosing randomly different sets from the measured leakages. Therefore, each randomly selected leakages was randomly assigned to a set.

A. Implementation setting

We simulate the leakages of a masked software implementation. Recall that we decompose the output y of the Sbox in d shares (denoted $[x_1, ..., x_d]$) such that $y = \mathsf{Sbox}(x) = \bigoplus_{i=1}^{d} x_i$. We exploit the Sbox of PRESENT (that works in the finite field $\mathsf{GF}(2^4)$) for efficiency reasons. For each execution of the simulator, we generate $d-1$ (uniformly) random values $[x_2, ..., x_d]$ while $x_1 = \bigoplus_{i=2}^{d} x_i \bigoplus y$.

Each execution of our simulator manipulates one share per cycle and each cycle is associated to n_s points in the leakage. Each cycle generates n_s samples in the leakages that depends on the HW of the manipulated share (i.e., each sample equals to $\mathsf{HW}(x_i) + N$ where x_i is the manipulated share and $N \sim \mathcal{N}(0, \sigma^2)$ represents a Gaussian physical noise).

In our experiments, we consider adversaries that do not know/use the mask value during the profiling step. This implies that the adversaries have to select a large set of points in the leakages in order to have all the samples related to all the shares. This leads also to include samples that do not relate to shares. In our experiment, this phenomenon is reflected by the generation of n_{ns} non-informative samples that are equal to N where $N \sim \mathcal{N}(0, \sigma^2)$ represents a Gaussian physical noise. In summary, the parameters of our simulator are d (the number of shares), N_p (the number of profiling leakages for each value of y), N_a (the number of attack leakages), n_s (the number of instants related to one share in the leakages), n_{ns} (the number of non-informative points per leakage) and σ^2 (the variance of the Gaussian physical noise).

Although we assume that the adversaries do not know/use the mask value during the profiling step, we assume that the adversaries know the following information: (1) during the profiling step, the adversaries know the executed cryptographic algorithm, the set of plaintexts and the key during the profiling step, while (2) during the attack step the adversaries only know the cryptographic algorithm and the set of plaintexts. The purpose of the adversaries is to extract the key by targeting the output of the Sbox.

B. Results on two shares masking scheme

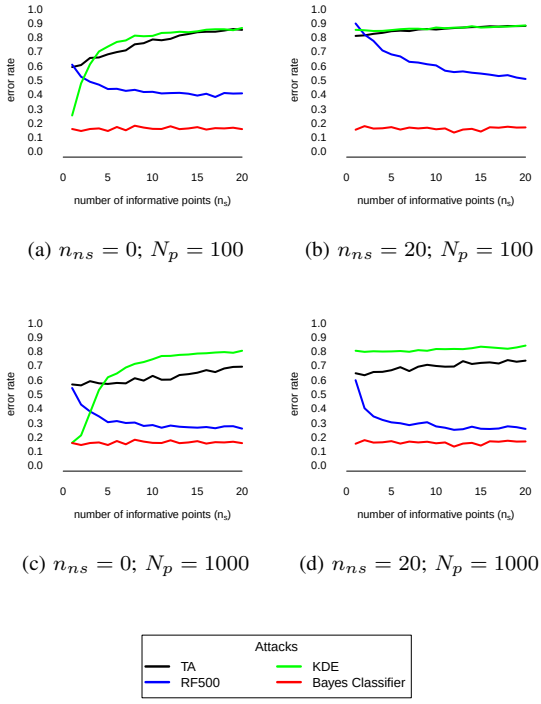
The first experiment evaluates the three practical profiled attacks against a masking scheme using two shares. Figure 3 reports the error rate of the four profiled attacks (i.e., TA, KDE, RF and Bayes classifier) in **low** physical noise level contexts as a function of the number of informative points (n_s), of the number of non-informative points (n_{ns}) and of number of leakages per target value in the profiling set (N_p). We observe four different settings:

- only the KDE lead to the same error rate as the Bayes classifier if the profiling set contains enough leakages (as seen in Figure 3c).
- if the leakages contain **few** informative points as well as **few** non-informative points, the exploitation of KDE leads to a lower error rate compared to the error rate of TA (as seen in Figure 3a and in Figure 3c). The reason is that, although TA has a lower variance than KDE, KDE exploit enough leakages during the profiling step in order to estimate accurately all its parameters (leading to a small estimation error) while containing a small assumption error. As a result, the small assumption and estimation errors of KDE provide better profiled attacks than TA (that contain a small estimation error but a high assumption error).
- if the leakages contain a **large** number of informative or non-informative points, the advantage of KDE to have a low assumption error vanishes compared to its estimation error (as seen in all the sub-figures of Figure 3 but mainly visible in Figure 3c and in Figure 3d). In other words, although the assumption error of TA is high, TA leads to a lower error rate thanks to a smaller estimation error.
- RF succeeds to exploit a large number of informative points (the more, the better). As a result, compared to TA and KDE, in a high number of non-informative points, RF succeeds to reduce the noise (due to the large number of non-informative points) by increasing the number of informative points (as seen in Figure 3a, in Figure 3b, in Figure 3c and in Figure 3d). Interestingly, several papers reports similar results on unprotected scenarios: RF are useful in order to deal with high-dimensional leakages (see for example [16], [20]). Note however that KDE outperform RF in contexts exploiting a low number of non-informative points thanks to a low level of physical noise (leading to a small estimation error for KDE).

Figure 4 extends the previous results in high noise contexts. In this setting,

- TA systematically outperform KDE due to a high noise (leading to a small assumption error for TA) and to a small estimation error for TA (compared to the estimation error of KDE). Furthermore, TA lead to the same error rate as the Bayes classifier if the profiling set contains enough leakages.
- RF outperform TA (and KDE) in high dimensionality contexts (i.e., when the number of points is large compared to the number of leakages in the profiling set). In other words, RF succeed to reduce the level of physical

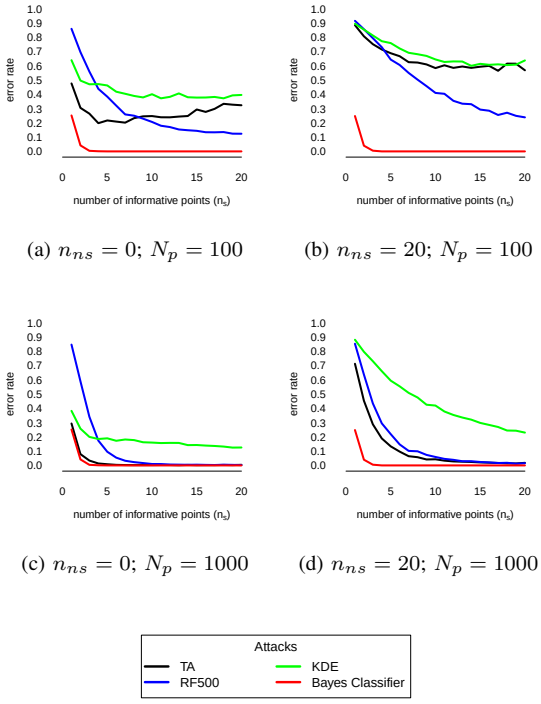
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2018.2879295, IEEE Transactions on Information Forensics and Security



(a) $n_{ns} = 0$; $N_p = 100$ (b) $n_{ns} = 20$; $N_p = 100$

(c) $n_{ns} = 0$; $N_p = 1000$ (d) $n_{ns} = 20$; $N_p = 1000$

Fig. 3: Error rate of TA, RF, KDE and Bayes classifier as a function of the number of instants (denoted n_s) per share in the leakage. We decompose the target value in 2 shares. We use N_p profiling leakages per target value in the profiling set and $N_a = 5$ attack leakages in the attack set. Each leakage contains $n_{ns} \in [0, 20]$ non-informative points. The standard deviation of the physical noise equals to $\sigma = 0.1$.



(a) $n_{ns} = 0$; $N_p = 100$ (b) $n_{ns} = 20$; $N_p = 100$

(c) $n_{ns} = 0$; $N_p = 1000$ (d) $n_{ns} = 20$; $N_p = 1000$

Fig. 4: Error rate of TA, RF, KDE and Bayes classifier as a function of the number of instants (denoted n_s) per share in the leakage. We decompose the target value in 2 shares. We use $N_p \in [100, 1000]$ profiling leakages per target value in the profiling set and $N_a = 100$ attack leakages in the attack set. Each leakage contains $n_{ns} \in [0, 20]$ non-informative points. The standard deviation of the physical noise equals to $\sigma = 1$.

noise in the leakages by exploiting a large number of points per leakage (as seen in Figure 4a and in Figure 4b).

In summary, with two shares, RF outperform the other approaches when the number of informative or non-informative points is high compared to the size of the profiling set (as reported by Lerman *et al.* in unprotected scenarios [20]). The number of non-informative points could be high due to the fact that, in practice, the adversaries do not know where are the informative points (if he does not know the mask values). More precisely, the adversaries require to use a large number of points to cover all the manipulated shares in the leakages (which could lead to a high number of non-informative points).

C. Results on three shares masking scheme

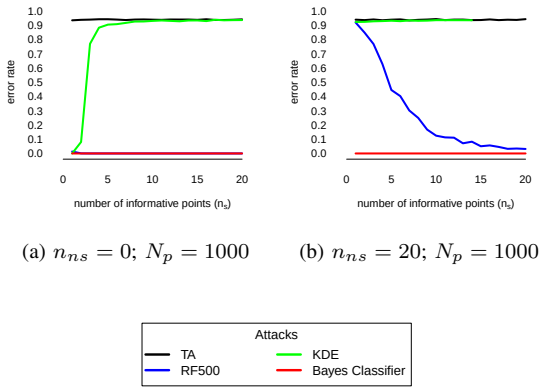
The next experiment focuses on three shares masking scheme. Figure 5 and Figure 6 report the error rate of the three practical profiled attacks in **low** and **high** physical noise level contexts as a function of the number of informative points (n_s), of the number of non-informative points (n_{ns}) and of number of leakages per target value in the profiling set (N_p). The figures report three results:

- TA recover no information on the secret key since all the information occurs in the third order moment (while TA only estimate the first and the second order moments).
- KDE outperform RF if the number of points per leakage is low compared to the number of leakages in the profiling set (as seen in Figure 6a).
- In high noise level contexts, the large number of informative points allows to reduce the level of noise in the leakages. As a result, in high noise level contexts and in high dimensionality contexts, RF outperform KDE thanks to the ability to exploit a large number of informative points (as reported by Lerman *et al.* in unprotected scenarios [20]). In other words, in high noise level contexts and in high dimensionality contexts, RF provide the closest results to the Bayes Classifier (as seen in Figure 6a and in Figure 6b).

D. Confirmation on actual measurements

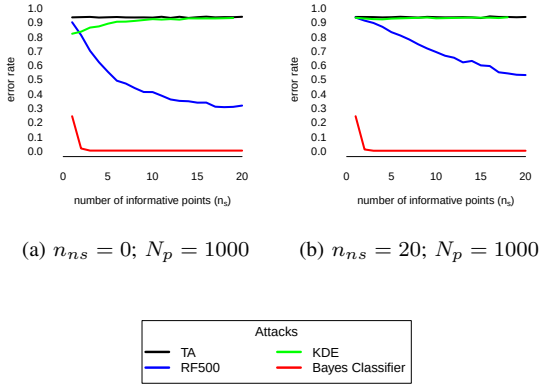
In order to confirm the previous results obtained with simulated leakages, this section presents the results on actual measurements by considering the performances of profiled attacks (i) on actual leakages, (ii) on actual leakages with added noise and (iii) on misaligned actual leakages.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2018.2879295, IEEE Transactions on Information Forensics and Security



(a) $n_{ns} = 0$; $N_p = 1000$ (b) $n_{ns} = 20$; $N_p = 1000$

Fig. 5: Error rate of TA, RF and KDE as a function of the number (denoted n_s) of instants per share in the leakage. We decompose the target value in 3 shares. We use $N_p = 1\,000$ profiling leakages per target value in the profiling set and $N_a = 200$ attack leakages in the attack set. Each leakage contains $n_{ns} \in [0, 20]$ non-informative points. The standard deviation of the physical noise equals to $\sigma = 0.1$. The RF line is located below the Bayes Classifier line when $n_{ns} = 0$.



(a) $n_{ns} = 0$; $N_p = 1000$ (b) $n_{ns} = 20$; $N_p = 1000$

Fig. 6: Error rate of TA, RF and KDE as a function of the number of instants (denoted n_s) per share in the leakage. We decompose the target value in 3 shares. We use $N_p = 1\,000$ profiling leakages per target value in the profiling set and $N_a = 1\,000$ attack leakages in the attack set. Each leakage contains $n_{ns} \in [0, 20]$ non-informative points. The standard deviation of the physical noise equals to $\sigma = 1$.

1) Performances over actual leakages: We measured actual traces from a device executing a masked implementation of PRESENT. The cryptographic device encrypts random plaintexts with a constant 80-bit key. We target the first round of the cipher (manipulating the SBox) and we focus on the first nibble of the key. A set of $100\,000$ power traces was collected on an 8-bit Atmel XMEGA (XMEGA128D4-U) target device (for the ChipWhisperer CW308 UFO Board) at a 7.37 MHz clock frequency. The power consumption of the device was measured using the ChipWhisperer-Pro (CW1200) oscilloscope that was set up to acquire 1600 samples (with a rate of 29.538 MS/s). We analyse leakages measured when the device executes the implementation of a first-order masked SBox (with 2 shares) based on table lookups [24], [26]. This strategy pre-computes a new masked SBox in memory (denoted SBox^*) for each execution of the cryptographic algorithm such that $\mathsf{SBox}^*_k(x \oplus m_{\mathsf{in}}) = \mathsf{SBox}(x) \oplus m_{\mathsf{out}} \quad \forall x \in \{0,1\}^4$ for a given pair of input and output masks (denoted respectively m_{in} and m_{out}) that are independent and identically distributed from a uniformly random source. Our implementation avoids a first order leakage by providing the masked input byte (i.e., $p \oplus k \oplus m_{\mathsf{in}}$ where p and k represent the plaintext and the key) to the executed implementation.

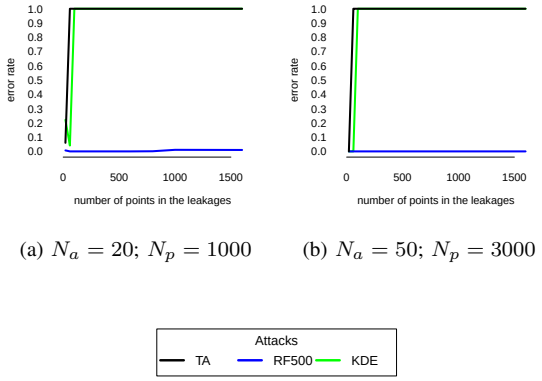
In order to estimate the quantity of information available in the leakages on the target value, we compute the Pearson correlation between the traces (at each instant) and the target value. The maximum absolute value of the Pearson correlation between the traces and the masked Sbox equals to 0.73 while the maximum absolute value of the Pearson correlation between the traces and the mask equals to 0.76.

We separate the dataset of $100\,000$ power traces into two disjoint subsets: the profiling source (containing $50\,000$ traces) and the attack source (containing $50\,000$ traces). The first source provides leakages used to build the models while the second source includes leakages to estimate the error rate of each model. We vary the number of attack traces, the number of profiling traces and the number of points per leakage. We select points that correlate (linearly) with (1) the HW of the masked SBox and (2) with the HW of the output mask.

We compare TA, RF (with 100 trees) and KDE. Figure 7 reports the error rate of these models. This figure shows that the error rate of RF is (almost) zero, which demonstrates that the quantity of noise in the multivariate leakages (compared to the quantity of signal) is very low. This figure also confirms the results provided in the simulated contexts: RF extract the key with a high probability in high dimensionality contexts while TA and KDE face a more and more difficult estimation problem when the size of the leakages increases. In the extreme contexts, when the evaluators do not know which points to select (i.e., when the evaluators consider the whole leakages since they do not know the mask values during the profiling step), only RF extract the key with a high success probability. Note also that, in low dimensionality contexts, the error rate of TA is (slightly) higher than the error rate of KDE due to the low level of noise (as reported with simulated leakages).

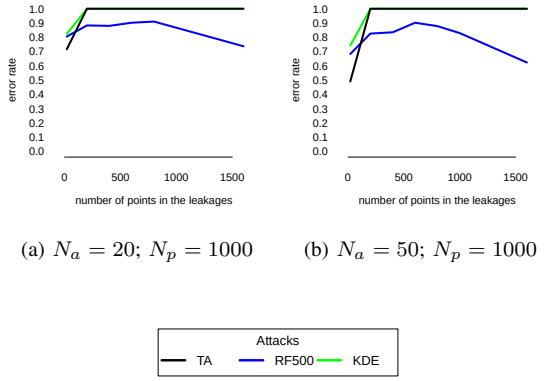
2) Performances over leakages with added noise: This section extends the previous results by increasing the (Gaussian)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2018.2879295, IEEE Transactions on Information Forensics and Security



(a) $N_a = 20$; $N_p = 1000$ (b) $N_a = 50$; $N_p = 3000$

Fig. 7: Error rate of TA, RF and KDE as a function of the number of points in each leakage. We decompose the target value in 2 shares. Half of the points in each leakage correlates to the mask and the rest of the points correlates to the masked Sbox. We use $N_p \in [1\,000, 3\,000]$ profiling leakages per target value in the profiling set and $N_a \in [20, 50]$ attack leakages in the attack set.



(a) $N_a = 20$; $N_p = 1000$ (b) $N_a = 50$; $N_p = 1000$

Fig. 8: Error rate of TA, RF and KDE as a function of the number of points in each leakage. We add Gaussian noise. We decompose the target value in 2 shares. Half of the points in each leakage correlates to the mask and the rest of the points correlates to the masked Sbox. We use $N_p \in [1\,000]$ profiling leakages per target value in the profiling set and $N_a \in [20, 50]$ attack leakages in the attack set.

noise in the measurements. By adding a univariate Gaussian noise having a standard deviation of 0.01, the maximum absolute value of the Pearson correlation between the traces and the masked Sbox decreases to 0.52 while the maximum absolute value of the Pearson correlation between the traces and the mask decreases to 0.48. Figure 8 plots the error rate of TA, RF (with 100 trees) and KDE. This figure shows that the error rates of TA, RF and KDE increase when the measurements contain more (Gaussian) noise. In low dimensionality contexts, although TA contain a higher assumption error compared to KDE, the error rate of TA is lower than the error rate of KDE due to a smaller estimation error for TA compared to the estimation error of KDE (as reported with simulated leakages). In high dimensionality contexts (i.e., when considering the whole leakages), RF outperform TA and KDE.

3) Performances over misaligned leakages: This section presents results on (temporal) misaligned leakages via random shifts of the acquired leakages during the execution of the masked implementation of PRESENT. These random shifts simulate a dysfunction in the power supply, an unstable clock, a lack of a good trigger signal or due to countermeasures such as random delay interrupts. Each leakage is randomly time-shifted from x points where x is sampled from a random variable following the discrete uniform distribution $\mathrm{unif}\{-\mathrm{S}, \mathrm{S}\}$ where $\mathrm{S} \in [2, 10]$. By considering $\mathrm{S} = 2$, the maximum absolute value of the Pearson correlation between the traces and the masked Sbox decreases to 0.16 while the maximum absolute value of the Pearson correlation between the traces and the mask decreases to 0.14. By considering $\mathrm{S} = 10$, the maximum absolute value of the Pearson correlation between the traces and the masked Sbox decreases to 0.07 while the maximum absolute value of the Pearson correlation between the traces and the mask decreases to 0.07.

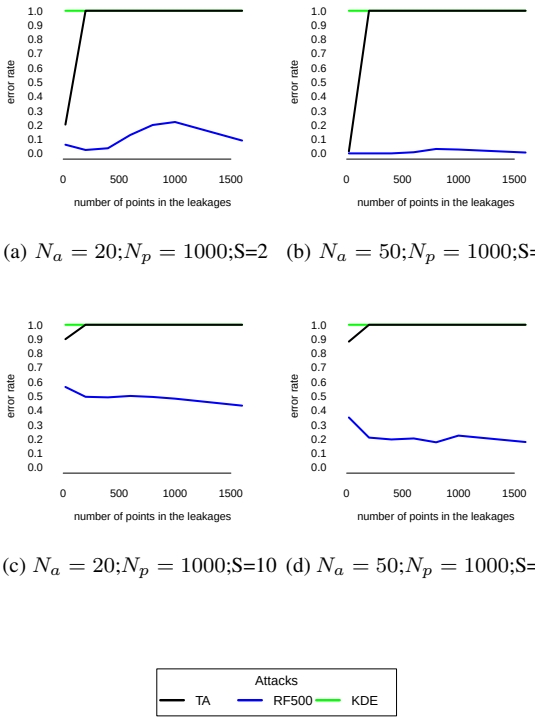
Figure 9 confirms the results obtained previously: TA outperform KDE in high noise contexts (resulted from misaligned leakages). Furthermore, as also reported previously, only RF extract the key with the highest success probability (compared to the other tested profiled attacks) when considering the whole leakages.

Compared to the results of RF exploiting leakages with added noise (reported in Section IV-D2), RF based attacks provide a higher success probability to extract the secret key when using misaligned leakages although that misaligned leakages leads to a lower (Pearson) correlation between the traces and the target values. The rationale could be that machine learning models can detect misaligned leakages (containing a high signal-to-noise ratio when realigned) and treat these leakages differently in order to realign measurements (as already reported by Cagli *et al.* [5] and by Lerman *et al.* [19]).

V. Conclusion

In conclusion, four conditions in the leakages increase concrete security against side-channel cryptanalysis: (i) a high physical noise to have a high estimation error, (ii) a large number of shares to force the estimation of high-order statistical moments (that increases exponentially the task of the adversary), (iii) a small number of informative points per share to avoid reduction of the noise by averaging points associated to the same share, and (iv) a large number of non-informative points to increase the variance of profiled attacks.

One practical consequence of our results is that we can improve the side-channel evaluation of masking implementations by exploiting (i) simple attacks having a smaller variance term (i.e., TA instead of KDE) when the leakages contain a high level of noise, and (ii) RF when the leakages contain a high number of informative points, non-informative points or



(a) $N_a = 20; N_p = 1000$;S=2 (b) $N_a = 50; N_p = 1000$;S=2

(c) $N_a = 20; N_p = 1000$;S=10 (d) $N_a = 50; N_p = 1000$;S=10

Fig. 9: Error rate of TA, RF and KDE as a function of the number of points in each leakage. We randomly shift the leakages from x points where x is sampled from a random variable following $\mathrm{unif}\{-\mathrm{S}, \mathrm{S}\}$ and where $\mathrm{S} \in [2, 10]$. We decompose the target value in 2 shares. Half of the points in each leakage correlates to the mask and the rest of the points correlates to the masked Sbox. We use $N_p \in [1\,000]$ profiling leakages per target value in the profiling set and $N_a \in [20, 50]$ attack leakages in the attack set.

shares. In practice, the evaluators can face a high number of points in the leakages due to the fact that they do not know where are the informative points (if they do not know the mask values). More precisely, the evaluators require to use a large number of points to cover all the manipulated shares in the leakages (if he does not know the mask values). Note that the evaluators can know the mask values during the profiling step and still not use them in order to speed up the attack phase (cf. Section II-B2). The results also highlight that evaluators can apply profiled attacks based on RF in an automatic way, i.e. without a priori knowledge on the analysed devices.

In this paper, we demonstrate that RF can outperform conventional approaches (i.e., TA and KDE) against masked implementations in high dimensionality contexts. As a future work, we envisage to generalise our experiments by analysing others machine learning models. As already shown in several papers [5], [22] as well as during the presentation of Elie Bursztein at CHES2018 [3], neural networks (and more precisely deep learning which represents neural networks with a large quantity of layers, neurones and parameters) have a high capacity to increase the success of attacks. As reported by Elie Bursztein at CHES2018 as well as by Liu *et al.* [21], this advantage goes along with two challenges to overcome before being competitive with other machine learning models (such as RF). First, the large quantity of different models (e.g., convolutional deep neural networks, recurrent neural networks, long short-term memory and multi-layer perceptrons) as well as the vast number of structures in each model (e.g., depending on the number of layers and the number of neurones) lead to a high computational time complexity during the profiling step and require a high expertise on deep learning. Second, in high dimensionality contexts (in which RF represent the best model against the targeted masked implementations), deep learning requires a large quantity of measurements (also known as big data settings) in order to estimate accurately its large set of parameters (as discussed in [11], [30]).

As a result, although we do not claim that RF is the universally best machine learning model for such contexts (as formalised by the no-free-lunch theorem), several works already report the high success probability of RF to extract the secret key from leakages while not requiring a large quantity of measurements neither a high time complexity for the selection of its structure [1], [16], [17], [20]. In this paper, we confirm these advantages on devices protected with a masking scheme.

[3]https://ches.iacr.org/2018/

APPENDIX

Proof. To prove Proposition III.1, let assume that each leakage contains n points (i.e., $n = n_s \times d$). Let also $\mathsf{f}^{\mathsf{u}}_y(\boldsymbol{T})$ be a unimodal Gaussian distribution associated to the target y (and estimated by a unimodal Gaussian TA), i.e.:

$$\mathsf{f}^{\mathsf{u}}_y(\boldsymbol{T}) = \frac{1}{\sqrt{(2\pi)^n \mathsf{det}\,(\boldsymbol{\Sigma}_{\mathsf{u},y})}} e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_{\mathsf{u},y})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_{\mathsf{u},y}(\boldsymbol{T}-\boldsymbol{\mu}_{\mathsf{u},y})}. \quad (15)$$

The assumption error (denoted AE) of a unimodal Gaussian TA (estimating $\mathsf{f}^{\mathsf{u}}_y(\boldsymbol{T})$) equals to the L_2 norm of the difference between the functions $\mathsf{f}^{\mathsf{u}}_y(\boldsymbol{T})$ and $\mathsf{f}^{\mathsf{m}}_y(\boldsymbol{T})$, i.e. $\mathsf{AE} = \int \left|\mathsf{f}^{\mathsf{u}}_y(\boldsymbol{T}) - \mathsf{f}^{\mathsf{m}}_y(\boldsymbol{T})\right|_2 \mathrm{d}\boldsymbol{T}$, where $|x|_2$ represents the L_2 norm. The assumption error is nonnegative and equals to zero if and only if the parametric estimator $\mathsf{f}^{\mathsf{u}}_y(\boldsymbol{T})$ and the true density $\mathsf{f}^{\mathsf{m}}_y(\boldsymbol{T})$ are the same. By incorporating the definition of $\mathsf{f}^{\mathsf{m}}_y$ (given in Equation 12), the assumption error equals to:

$$\mathsf{AE} = \int \left|\mathsf{f}^{\mathsf{u}}_y(\boldsymbol{T}) - \frac{1}{\|\mathcal{X}_y\|} \frac{1}{\sqrt{(2\pi)^n \mathsf{det}\,(\boldsymbol{\Sigma}_{\mathsf{m},y})}} \sum_{\boldsymbol{x}\in\mathcal{X}_y} e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_{\mathsf{m},y}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})}\right|_2 \mathrm{d}\boldsymbol{T} \quad (16)$$

Let ϵ, ϵ' be small real values and c be a big real value[4]. In Equation 16, we can replace all the terms $e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_{\mathsf{m},y}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})}$ with the constant value $e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_{y})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_{y}(\boldsymbol{T}-\boldsymbol{\mu}_{y})}$ if all the terms $e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_{\mathsf{m},y}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})}$ are similar to the value $e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_{y})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_{y}(\boldsymbol{T}-\boldsymbol{\mu}_{y})}$, i.e. if:

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_y \; |(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_{\mathsf{m},y}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}}) - (\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}'})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_{\mathsf{m},y}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}'})|_2 < \epsilon. \quad (17)$$

[4]We discuss the impact of ϵ, ϵ' and c on the assumption error at the end of the proof.

The following two conditions satisfy Equation 17:

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_y \quad |\boldsymbol{\mu}_{\boldsymbol{x}} - \boldsymbol{\mu}_{\boldsymbol{x}'}| < \epsilon', \tag{18}$$

$$\forall i \quad \lambda_i > c, \tag{19}$$

where λ_i is the i-th eigen value of $\boldsymbol{\Sigma}$. The substitution of all the terms $e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathrm{m},y}^{-1}(\boldsymbol{T}-\boldsymbol{\mu}_{\boldsymbol{x}})}$ with the constant value $e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_y)^{\mathrm{T}}\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{T}-\boldsymbol{\mu}_y)}$ leads to the following:

$$\begin{aligned}\mathsf{AE} = &\int |\mathsf{f}_y^{\mathsf{u}}(\boldsymbol{T}) \qquad (20)\\ &- \frac{1}{\|\mathcal{X}_y\|}\frac{1}{\sqrt{(2\pi)^n \mathsf{det}\,(\boldsymbol{\Sigma}_y)}}\sum_{\boldsymbol{x}\in\mathcal{X}_y} e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_y)^{\mathrm{T}}\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{T}-\boldsymbol{\mu}_y)}|_2\\ &\mathrm{d}\boldsymbol{T}\\ = &\int |\mathsf{f}_y^{\mathsf{u}}(\boldsymbol{T}) \qquad (21)\\ &- \frac{1}{\sqrt{(2\pi)^n \mathsf{det}\,(\boldsymbol{\Sigma}_y)}} e^{-\frac{1}{2}(\boldsymbol{T}-\boldsymbol{\mu}_y)^{\mathrm{T}}\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{T}-\boldsymbol{\mu}_y)}|_2\,\mathrm{d}\boldsymbol{T}.\end{aligned}$$

Thus, we minimise the assumption error AE when $\boldsymbol{\Sigma}_{u,y} = \boldsymbol{\Sigma}_y$ and $\boldsymbol{\mu}_{\mathrm{u},y} = \boldsymbol{\mu}_y$. This result demonstrates that the multimodal Gaussian function $\mathsf{f}_y^{\mathsf{m}}(\boldsymbol{T})$ can be modelised with a unimodal Gaussian function $\mathsf{f}_y^{\mathsf{u}}(\boldsymbol{T})$ (estimated by a unimodal Gaussian TA) if the level of noise is high (cf., Equation 19) or if the distance between the values $\boldsymbol{\mu}_{\boldsymbol{x}}$ is small (cf., Equation 18). In other words, the smaller the value of ϵ' or the bigger the value of c, the smaller the value of ϵ, and the smaller the assumption error of a unimodal Gaussian TA. □

REFERENCES

[1] V. Banciu, E. Oswald, and C. Whitnall. Reliable information extraction for single trace attacks. In W. Nebel and D. Atienza, editors, *DATE 2015, Grenoble, France, March 9-13, 2015*, pages 133–138. ACM, 2015.

[2] G. Barthe, F. Dupressoir, S. Faust, B. Grégoire, F-X. Standaert, and P-Y. Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In J-S. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, volume 10210 of *LNCS*, pages 535–566, 2017.

[3] A. Battistello, J-S. Coron, E. Prouff, and R. Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. In B. Gierlichs and A. Y. Poschmann, editors, *CHES 2016, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *LNCS*, pages 23–39. Springer, 2016.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] E. Cagli, C. Dumas, and E. Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures - profiling attacks without pre-processing. In W. Fischer and N. Homma, editors, *CHES 2017, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *LNCS*, pages 45–68. Springer, 2017.

[6] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi. Towards sound approaches to counteract power-analysis attacks. In M. J. Wiener, editor, *CRYPTO '99, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *LNCS*, pages 398–412. Springer, 1999.

[7] S. Chari, J. R. Rao, and P. Rohatgi. Template attacks. In B. S. Kaliski Jr., Ç. Kaya Koç, and C. Paar, editors, *CHES 2002, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *LNCS*, pages 13–28. Springer, 2002.

[8] J. Cooper, E. De Mulder, G. Goodwill, J. Jaffe, G. Kenworthy, and P. Rohatgi. Test vector leakage assessment (TVLA) methodology in practice (extended abstract). In *ICMC 2013*, 2013.

[9] J-S. Coron, C. Giraud, E. Prouff, S. Renner, M. Rivain, and P. K. Vadnala. Conversion of security proofs from one leakage model to another: A new issue. In W. Schindler and S. A. Huss, editors, *COSADE 2012, Darmstadt, Germany, May 3-4, 2012. Proceedings*, volume 7275 of *LNCS*, pages 69–81. Springer, 2012.

[10] F. Durvaux, F-X. Standaert, and N. Veyrat-Charvillon. How to certify the leakage of a chip? In Phong Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *LNCS*, pages 459–476. Springer, 2014.

[11] I. J. Goodfellow, Y. Bengio, and A. C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016.

[12] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi. A testing methodology for side channel resistance validation. In *NIST non-invasive attack testing workshop*, 2011.

[13] L. Goubin and J. Patarin. DES and differential power analysis (the "duplication" method). In Ç. Kaya Koç and C. Paar, editors, *CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *LNCS*, pages 158–172. Springer, 1999.

[14] Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. In D. Boneh, editor, *CRYPTO 2003, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *LNCS*, pages 463–481. Springer, 2003.

[15] K. Lemke-Rust and C. Paar. Gaussian mixture models for higher-order side channel analysis. In P. Paillier and I. Verbauwhede, editors, *CHES 2007, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *LNCS*, pages 14–27. Springer, 2007.

[16] L. Lerman, G. Bontempi, and O. Markowitch. Side Channel Attack: an Approach Based on Machine Learning. In *COSADE 2011*, pages 29–41. Center for Advanced Security Research Darmstadt, 2011.

[17] L. Lerman, G. Bontempi, and O. Markowitch. Power analysis attack: an approach based on machine learning. *International J. of Applied Cryptography*, 3(2):97–115, 2014.

[18] L. Lerman, G. Bontempi, and O. Markowitch. The bias-variance decomposition in profiled attacks. *J. Cryptographic Engineering*, 5(4):255–267, 2015.

[19] L. Lerman, Z. Martinasek, and O. Markowitch. Robust profiled attacks: should the adversary trust the dataset? *IET Information Security*, 11(4):188–194, 2017.

[20] L. Lerman, R. Poussier, G. Bontempi, O. Markowitch, and F-X. Standaert. Template attacks vs. machine learning revisited (and the curse of dimensionality in side-channel analysis). In S. Mangard and A. Y. Poschmann, editors, *COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers*, volume 9064 of *LNCS*, pages 20–33. Springer, 2015.

[21] P. Liu, K-K. Raymond Choo, L. Wang, and F. Huang. SVM or deep learning? A comparative study on remote sensing image classification. *Soft Comput.*, 21(23):7053–7065, 2017.

[22] H. Maghrebi, T. Portigliatti, and E. Prouff. Breaking cryptographic implementations using deep learning techniques. In C. Carlet, M. A. Hasan, and V. Saraswat, editors, *SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*, volume 10076 of *LNCS*, pages 3–26. Springer, 2016.

[23] S. Mangard, E. Oswald, and F-X. Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.

[24] E. Prouff and M. Rivain. A generic method for secure sbox implementation. In S. Kim, M. Yung, and H-W. Lee, editors, *WISA 2007, Jeju Island, Korea, August 27-29, 2007, Revised Selected Papers*, volume 4867 of *LNCS*, pages 227–244. Springer, 2007.

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

[26] K. Schramm and C. Paar. Higher order masking of the AES. In D. Pointcheval, editor, *CT-RSA 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*, volume 3860 of *LNCS*, pages 208–225. Springer, 2006.

[27] D.W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley Series in Probability and Statistics. Wiley, 2015.

[28] F-X. Standaert, T. Malkin, and M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In A. Joux, editor, *EUROCRYPT 2009, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *LNCS*, pages 443–461. Springer, 2009.

[29] F-X. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, and S. Mangard. The world is not enough: Another look on second-order DPA. In M. Abe, editor, *ASIACRYPT 2010, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *LNCS*, pages 112–129. Springer, 2010.

[30] Q. Zhang, L. T. Yang, Z. Chen, and P. Li. A survey on deep learning for big data. *Information Fusion*, 42:146–157, 2018.