Start Simple and then Refine: Bias-Variance Decomposition as a Diagnosis Tool for Leakage Profiling

Liran Lerman, Nikita Veshchikov, Olivier Markowitch, François-Xavier Standaert

Abstract—Evaluating the resistance of cryptosystems to side-channel attacks is an important research challenge. Profiled attacks reveal the degree of resilience of a cryptographic device when an adversary examines its physical characteristics. So far, evaluation laboratories launch several physical attacks (based on engineering intuitions) in order to find one strategy that eventually extracts secret information (such as a secret cryptographic key). The certification step represents a complex task because in practice the evaluators have tight memory and time constraints. In this paper, we propose a principled way of guiding the design of the most successful evaluation strategies thanks to the (bias-variance) decomposition of a security metric of profiled attacks. Our results show that we can successfully apply our framework on unprotected and protected algorithms implemented in software and hardware.

Index Terms—Side-channel attacks, profiled attacks, bias-variance decomposition, diagnosis tool, evaluation tool.

1 Introduction

In 1996, Kocher introduced side-channel cryptanalyses that analyze physical characteristics (called *leakages* or *traces*) of cryptographic devices in order to extract secret information [21]. The rationale is that the leakages of (hardware and software) implementations depend on the manipulated data and the executed operations. As a result, cryptographic algorithms secured from a point of view of classical cryptanalysis can be insecured when implemented in devices. Kocher exploited the execution time in order to recover the secret key from several crypto systems. In 1999, Kocher extended the previous proposition with (differential) power analysis that compare the outputs of a leakage model (parameterized by secret key hypotheses) with the actual power leakages [22]. Nowadays, cryptanalytic side channel attacks also employ other sources of emanations such as the electromagnetic radiation [14] and the sound [15]. In this paper, we focus on side-channel attacks based on the power consumption and the electromagnetic radiation as both leakages can be addressed using the same techniques.

The evaluators of the robustness of cryptographic devices usually analyze several (evaluation) settings by varying, for example, the leakage dimension, the number of leakages as well as the distinguishers (that compare actual leakages with the modeled leakages). The large number of possible evaluation settings requires to consider a significant number of attack strategies that also depend on the a priori information on the target device. *Non-profiled attacks* (introduced by Kocher [22]) work under the assumption that the adversary has a priori knowledge on the physical behavior of the cryptographic device (e.g., the power consumption of a device linearly correlates to the manipulated data). By contrast, *profiled attacks* exploit an offline learning step (also known as profiling step) executed on a device (similar to the target device) in order to automatically extract knowledge about the leakage function [12]. We focus on profiled attacks introduced as the strongest leakage analysis in an information theoretic sense [4].

In practice, profiled attack strategies still require (1) to minimize the number of estimated parameters of the leakage model due to a limited number of measurements available during the learning phase (that can lead to *estimation error*), and (2) to assume some knowledge on the leakage distribution (that can lead to *assumption error*) when the adversary considers parametric profiled attacks. These constraints lead the cryptographic community to propose a plethora of profiled attacks, e.g., [4], [23], [25], [31].

1.1 Our contributions

In Eurocrypt 2014, Durvaux *et al.* proposed a *certification tool* to determine whether the estimated profiled attacks suffer from an assumption error or from an estimation error [11] (it was then simplified/specialized in 2016 [10]). Here we complement the certification tool by providing a *diagnosis tool* (based on the *bias-variance decomposition* [8], [9], [24]) that guides the design of the best profiled attack optimizing one or several constraints decided by the evaluation laboratory (e.g., maximizing the success probability of profiled attacks with 5 attack traces measured on the target device or minimizing the number of leakages required to reach the success probability of 0.7) or assumed by the implemented scheme (e.g., maximizing the success probability of profiled attacks from a single leakage measured on a fresh re-keying scheme [28]).

As far as we know, the certification process of cryptographic devices consists of testing several popular pro-

- *L. Lerman, N. Veshchikov and O. Markowitch are affiliated with the QualSec group from the Université libre de Bruxelles in Belgium.*
- *F.-X. Standaert is affiliated to the ICTEAM/ELEN/Crypto Group of the Université catholique de Louvain in Belgium.*
- *Corresponding author e-mail: llerman@ulb.ac.be*

filed attacks with several settings in order to find the best strategy. Nevertheless, this approach requires a significant computational power. Here, we promote a technique that starts with a specific evaluation setting (e.g., a low number of leakages with a simple attack) and then refines the attack according to our diagnosis tool. The rationale to consider simple settings at the beginning of the evaluation lies to a variance issue: the success probability of profiled attacks with excessive complexity could be lower than the success probability of simpler approaches (due to variance issues). Note also that complex approaches suffer from high time/memory complexity: the time/memory complexity increases as a function of (1) the number of points per leakage, (2) the number of leakages exploited during the learning and the attack steps, and (3) (especially against protected devices) the complexity of the attack.

Our diagnosis tool works in a systematic way (1) by extracting each term impacting the error rate, and (2) by reducing the contribution of those terms in order to reach a lower error rate, optionally with respect to constraints fixed by the evaluator (such as the number of leakages used during the learning step). As the last step, an evaluator can apply the certification tool of Durvaux *et al.* in order to gauge the quality of the best attack found with the diagnosis tool whether the framework advises a profiled attack providing probabilities for each target value (also known as a probability-based profiled attack)[1]. It is worth to note that our diagnosis tool works on probability-based and score-based profiled attacks. We apply our framework to test eight popular profiled attacks on actual datasets as well as on simulated leakages. We demonstrate the usefulness of our diagnosis tool by applying our framework on unprotected and protected (software and hardware) implementations.

1.2 Related works

Our framework represents a counterpart in the profiled attack approach of the recently proposed non-profiled *stepwise linear regression* attacks [36]. More precisely, the stepwise linear regression techniques tune the parameters of specific probability-based non-profiled attacks while our approach applies to all types of profiled attacks. Similarly, our diagnosis tool extends the papers of Choudary *et al.* that report several guidelines in the context of template attacks and stochastic attacks (representing well known probability-based profiled attacks) [5], [6]. All the hints proposed by Choudary *et al.* can be used in our diagnosis tool to avoid several numerical and estimation issues as well as to reduce the running time of the attacks (e.g., by exploiting dimensionality reduction techniques and efficient profiled attacks).

The main advantages of our approach are: (1) the framework can be applied to any profiled attack (including probability-based and score-based profiled attacks) thanks to the decomposition of a security metric (for example the success rate or success probability, put forward by Standaert *et al.* [33]), (2) the framework operates without any knowledge of the true leakage distribution, and (3) the framework works on unprotected and protected implementations.

1. The certification tool of Durvaux *et al.* operates only on probability-based profiled attacks.

We base our approach on the bias-variance decomposition [8], [9], which was recently introduced to the field of side-channel analysis by Lerman *et al.* [24]. Our paper differs from the paper of Lerman *et al.* in three main points:

1) We use the bias-variance decomposition for different goals. We do not only use this tool in order to find out what impacts the error rate of template attacks and stochastic attacks but we also exploit this tool in order to extract the best profiled attack (i.e., fine-tune the meta-parameters). Thus, we provide a framework that can be efficiently applied during the evaluation of cryptographic systems taking into account real-world constraints.
2) We study additional profiled attacks in order to cover all types of profiled attacks. We do not only analyze probability-based profiled attacks (i.e., template attacks and stochastic attacks) but we additionally study score-based profiled attacks and demonstrate that our diagnosis tool can be used on any profiled attacks.
3) We conducted the experiments in different settings. We do not only test attacks on simulated leakages (that highlight the usefulness of this approach from a theoretical point of view) but we also consider experiments on real power traces collected on an unprotected and a protected implementations (in hardware and in software), which allows to evaluate our diagnosis tool in practice.

1.3 Organization of this paper

The rest of the paper is organized as follows. Section 2 contains preliminary notions on physical attacks. Section 3 provides introduction to the bias-variance decomposition. Section 4 details our diagnosis tool, and the results of its application on profiled attacks against unprotected software as well as hardware implementations. Section 5 extends the results of Section 4 to a protected environment. Section 6 analyses the impact of an assumption made during the experiments. Finally, Section 7 concludes the paper.

2 Background on Side-Channel Attacks

2.1 Physical attacks

We assume that the adversary wants to retrieve the secret key used when the cryptographic device (that executes a known encryption algorithm) encrypts known plaintexts. In order to find the secret key, the adversary targets a set of key-related information (called the *target intermediate values*) with a *divide-and-conquer approach*. The divide-and-conquer strategy extracts information on separate parts of the secret key (e.g., the adversary extracts each byte of the key independently) and then combines the results in order to get the full secret key. In the following, we systematically use the term key to denote the target of our attacks, though, in fact, we address one byte at a time.

During the execution of the encryption algorithm, the cryptographic device processes a function f (e.g., the SBox of the block-cipher AES)

$$\begin{aligned} \mathsf{f}\colon \mathcal{P} \times \mathcal{K} &\to \mathcal{Y} \\ y &= \mathsf{f}_k(p), \end{aligned} \tag{1}$$

that outputs the target intermediate value y and where $k \in \mathcal{K}$ is a key-related information (e.g., one byte of the secret key), and $p \in \mathcal{P}$ represents information known by the adversary (e.g., one byte of the plaintext). For the sake of simplicity, we assume that (1) the function f is bijective, (2) the output of the function f is encoded using 8 bits, and (3) the adversary targets one byte of the key.

2.1.1 Side-channel attacks

Let ${}^{j}T_y$ be the j-th leakage measured when the device manipulates the value y. In the following, we represent each leakage with a vector of real values measured at different instants on the analyzed device. The number of samples (i.e., the length of the vector ${}^{j}T_y$) equals to n in unprotected contexts. We denote ${}^{j}_{t}T_y$ the j-th leakage (associated to the target value y) measured at time t such that:

$$
\begin{aligned}
{}^{j}_{t}T_y &= {}_{t}\mathsf{L}\left(y\right) + {}^{j}_{t}\epsilon_y, &(2)\\
&= {}_{t}\mathsf{L}\left(\mathsf{f}_k\left(p\right)\right) + {}^{j}_{t}\epsilon_y, &(3)
\end{aligned}
$$

where ${}^{j}_{t}\epsilon_y \in \mathbb{R}$ is the noise of the trace ${}^{j}_{t}T_y$ (i.e., a standard additive noise assumption defined by Mangard *et al.* [26]) following for example a Gaussian distribution, and ${}_{t}\mathsf{L}$ is the (deterministic) leakage function at time t. Let assume that the adversary targets the output of the function $\mathsf{f}_k\left(p\right)$. The function ${}_{t}\mathsf{L}$ can be *linear* or *nonlinear* of the output bits of $\mathsf{f}_k\left(p\right)$ [1]. More precisely, we say that the leakage functions are linear if they provide leakages ${}^{j}_{t}T_y$ depending on the weighted sum of each bit of the output of $\mathsf{f}_k\left(p\right)$ while nonlinear leakage functions provide leakages ${}^{j}_{t}T_y$ depending on the weighted sum of *products* of bits of the output of $\mathsf{f}_k\left(p\right)$. Formally, we consider polynomial (leakage) functions ${}_{t}\mathsf{L}\left(y\right)$, i.e.:

$$
{}_{t}\mathsf{L}\left(y\right) = {}_{t}c + \sum_{u} {}_{t}\alpha_u \ \mathsf{g}_u\left(y\right), \tag{4}
$$

where ${}_{t}c$ and ${}_{t}\alpha_u$ are real numbers while $\mathsf{g}_u(y)$ is a monomial of the form $\prod_{b\in\mathcal{B}} \mathsf{Bit}_b\left(y\right)$ in which $\mathsf{Bit}_b\left(y\right)$ returns the b-th bit of y and $\mathcal{B} \subset \{1, 2, ..., 8\}$. This represents a usual assumption in side-channel attacks [31]. Linear leakage functions consider that the cardinality of $\mathcal{B}$ in each monomial equals to 1 while nonlinear leakage functions contain monomials (with a non-zero coefficient) of the form $\prod_{b\in\mathcal{B}} \mathsf{Bit}_b\left(y\right)$ in which the cardinality of $\mathcal{B}$ is strictly bigger than 1. Evaluators often model leakage functions as (1) the Hamming weight of the manipulated value y (denoted $\mathsf{HW}(y)$) for software implementations (representing a linear leakage function), and (2) the Hamming distance (HD) between two manipulated values for hardware implementations (representing a nonlinear leakage function since if a and b are in $\{0, 1\}$, then the Hamming distance between a and b equals to $a + b - 2ab$)[2].

A *side-channel attack* is a process during which an attacker analyses leakages measured on a target device in order to extract information on the secret value. In order to protect the implementations against physical attacks, the designers employ (among others) d-order masking techniques that split each sensitive information y (that depends on the secret key) in $d + 1$ uniformly distributed variables (called shares), denoted $\{x_0, x_1, ..., x_d\}$, such that $y = \sum_{i=0}^{d} x_i$ (where $\sum$ represents the method combining shares) [3], [16]. Typically, the shares $\{x_1, ..., x_d\}$ represent d uniformly distributed random values (called the masks) while x_0 equals to $y + \sum_{i=1}^{d} x_i$. The masking schemes (of order d) leak no information on the sensitive variable y whenever the adversary combines strictly less than $d+1$ different instants in order to recover the sensitive information (when each sample depends on a different share). In the following, we assume that the masking techniques of order d use $d + 1$ shares, leading to $(d + 1, d + 1)$ secret sharing schemes. Furthermore, in protected contexts, we denote n the number of samples in the vector ${}^{j}T_y$ associated to one share (i.e., ${}^{j}T_y$ contains $n \times (d + 1)$ samples in total).

Profiled attacks (belonging to side-channel attacks) represent efficient attacks thanks to a learning step (also known as a profiling step). More precisely, these approaches build a distinguisher $\mathsf{A}(\mathcal{T}_{\mathrm{PS}}, \mathcal{T}_{\mathrm{AS}})$ that:

1) 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
2) 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.

The quality of the distinguisher can be analyzed by estimating the *success rate*, i.e. the probability that the distinguisher returns the right key based on a set of attack traces. The *error rate* equals to the probability that the distinguisher *does not* return the right key based on the attacking set.

The *Bayes classifier* (also known as the Bayes) denotes the best possible theoretical attack which practical attacks can reach when there is no assumption error and no estimation error. More formally, let $\mathsf{A}_b(\cdot)$ be the Bayes classifier that takes as input a set of attack traces $\mathcal{T}_{\mathrm{AS}}$, the function $\mathsf{A}_b(\cdot)$ represents a classifier that minimizes the error rate, i.e.:

$$
\begin{aligned}
\mathsf{A}_b(\mathcal{T}_{\mathrm{AS}}) &\in \underset{k\in\mathcal{K}}{\operatorname{argmax}} \Pr\left[k \mid \mathcal{T}_{\mathrm{AS}}\right] &(5)\\
&= \underset{k\in\mathcal{K}}{\operatorname{argmax}} \Pr\left[\mathcal{T}_{\mathrm{AS}} \mid k\right] \times \Pr\left[k\right]. &(6)
\end{aligned}
$$

Note that several values k could maximize the value $\Pr\left[\mathcal{T}_{\mathrm{AS}} \mid k\right] \times \Pr\left[k\right]$, leading to a set of possible keys (which explains the symbol $\in$ in Equation 5). Note also that, in the side-channel attacks literature, the Bayes classifier represents the model estimating Equation 6 while, in this paper, the Bayes classifier refers to the optimal classification with known probability density functions used in Equation 6.

2.2 Concrete distinguishers

In practice, an adversary estimates the Bayes classifier (i.e., the Equation 6) with concrete distinguishers detailed in this section. In the following, we define the *complexity of an attack* by the number of parameters to estimate (i.e., for a given distinguisher, an increase of the number of parameters to estimate leads to an increase of the complexity of the attack).

2. Note that the HD is linear if viewed as a function of the bit flips.

2.2.1 Template attacks

(Gaussian) *Template attacks* (TA) [4] estimates the Equation 6 by assuming that $\Pr\left[{}^{j}T_y \mid y\right]$ follows a Gaussian distribution $\mathcal{N}(\hat{\mu}_y, \hat{\Sigma}_y)$ for each value y where $\hat{\mu}_y$ and $\hat{\Sigma}_y$ are respectively the sample mean and the sample covariance matrix of the traces associated to y. In what follows we assume that the noise is independent of y in unprotected contexts [6]. This property allows to estimate the same physical noise (represented by Σ) for all the target values. By contrast, in protected settings exploiting first-order masking schemes, we assume that the adversary has no information on the shares (except the instants depending on the shares) during the profiling step leading to a dependency between the second-order moment Σ_y and the sensitive information y. 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. In the following we will relax this assumption, which leads to another attack (that we call stochastic-based mixture attacks). Note also that template attacks using no information on the mask values (during the profiling step) allow us to highlight the effect on the outputs given by the diagnosis tool when the evaluators exploit bias models. Indeed, in protected contexts, template attacks (unlike template-based mixture attacks that take into account the mixture structure of the probability density function) represent biased models since these attacks assume that the leakages associated to a sensitive information y (split in several shares $\{x_0, ..., x_d\}$) follow a Gaussian distribution (i.e., by estimating $\Pr\left[{}^{j}T_y \mid y\right]$ with one Gaussian distribution) while the leakages follow a multimodal distribution, i.e.:

$$\Pr\left[{}^{j}T_y \mid y\right] = \sum_{x_1,...,x_d} \Pr\left[{}^{j}T_y \mid x_1, ..., x_d, x_0 = y + \sum_{i=1}^{d} x_i\right] \times \Pr\left[x_0, ..., x_d\right], \quad (7)$$

where $\Pr\left[{}^{j}T_y \mid x_1, ..., x_d, x_0 = y + \sum_{i=1}^{d} x_i\right]$ (in which the symbols $\sum$ and $+$ represent the methods combining shares) follows a Gaussian distribution if we consider Gaussian noise. In other words, although the unbiased/optimal profiled attacks represent the Gaussian mixture templates attacks (in which the adversary estimates the multimodal Gaussian distribution with a mixture of Gaussian distributions), we exploit (unimodal) template attacks that work if the first or the second order moments contain information on the target value (e.g., unimodal template attacks applied on first-order masking schemes). Note that unimodal template attacks extract the same quantity of information from the leakages as template-based mixture attacks if the leakages contain a high physical Gaussian noise and if the target implementation executes a first-order masking scheme (as reported by Grosso *et al.* [17]).

During the attack step, the adversary classifies $\mathcal{T}_{\mathrm{AS}} = \left\{{}^{1}T, ..., {}^{N_a}T\right\}$ (where ${}^{j}T$ denotes the j-th measurement on the target device, and N_a is the number of attack traces) by using the equation:

$$\hat{k} \in \arg\max_{k \in \mathcal{K}} \prod_{j=1}^{N_a} \Pr\left[{}^{j}T \mid y = \mathsf{f}_k(p_j)\right] \times \Pr\left[y = \mathsf{f}_k(p_j)\right], \quad (8)$$

$$\approx \arg\max_{k \in \mathcal{K}} \prod_{j=1}^{N_a} \hat{\Pr}\left[{}^{j}T \mid y = \mathsf{f}_k(p_j); \hat{\theta}_y\right] \times \hat{\Pr}\left[y = \mathsf{f}_k(p_j)\right], \quad (9)$$

where $\hat{\theta}_y = \{\hat{\mu}_y, \hat{\Sigma}_y\}$, and p_j is the j-th plaintext used by the device when the adversary measured the j-th attack trace.

2.2.2 Stochastic attacks and stochastic-based mixture attacks

Stochastic attacks (SA) [31] model the leakage information at time t as a function of the target value y with a regression model h spanned by U functions g_u (where $u \in [1; U]$), i.e.:

$${}^{j}_{t}T_y = \mathsf{h}\left(y, {}_{t}\theta\right) + {}^{j}_{t}\epsilon_y, \quad (10)$$

$$= {}_{t}c + \sum_{u=1}^{U} {}_{t}\alpha_u \ \mathsf{g}_u\left(y\right) + {}^{j}_{t}\epsilon_y, \quad (11)$$

where ${}_{t}\theta = \{{}_{t}c, {}_{t}\alpha_1, ..., {}_{t}\alpha_U\} \in \mathbb{R}^{U+1}$ represents the parameter of the regression model h at time t. Stochastic attacks assume that g_u is a monomial of the form $\prod_{b \in \mathcal{B}} \mathsf{Bit}_b\left(y\right)$ where $\mathsf{Bit}_b\left(y\right)$ returns the b-th bit of y and $\mathcal{B} \subset \{1, 2, ..., 8\}$.

We define the *degree of a stochastic attack* as the maximum number of variables in each monomial of h with a non-zero coefficient. More formally, stochastic attacks of degree i contain all the monomials of the form $\prod_{b \in \mathcal{B}} \mathsf{Bit}_b\left(y\right)$ in which the cardinality of $\mathcal{B}$ is in $\{1, 2, ..., i\}$. We denote SAi stochastic attacks of degree i (e.g., SA1 denotes stochastic attacks of degree 1). Note that, in unprotected contexts, template attacks are equivalent to stochastic attacks of degree 8 when the adversary estimates only one Σ for all the target values (see Section 2.2 "Profiled attacks" in [24]). As a result, in unprotected contexts, stochastic attacks of degree strictly less than 8 represent distinguishers with lower complexity than template attacks.

In protected environments, each function h takes as input the value of one share. For example, if the device manipulates the share x_i at time t, then stochastic attacks modelize the deterministic part of the leakage at time t by $\mathsf{h}\left(x_i, {}_{t}\theta\right)$. As a result, unlike with template attacks, we assume that the stochastic attacks (that we call *stochastic-based mixture attacks* in protected contexts) know the value of the manipulated masks during the profiling step.

Regarding the attack step against an unprotected implementation, the adversary uses Equation 8 by assuming that $\Pr\left[{}^{j}T_y \mid y\right]$ follows the Gaussian distribution $\mathcal{N}\left(\mathsf{h}\left(y, \theta\right), \Sigma\right)$ where $\mathsf{h}(y, \theta)$ represents the vector $[\mathsf{h}(y, {}_{1}\theta), \mathsf{h}(y, {}_{2}\theta), ..., \mathsf{h}(y, {}_{n}\theta)]$, n represents the number of samples (i.e., the length of ${}^{j}T_y$), and Σ is the covariance matrix of the residual term[3]. In a protected environment, due to the fact that the adversary has no a priori knowledge on the value of the manipulated masks during the attack

3. The residual term represents the deviation of the actual leakages (associated to known keys and known plaintexts) from the output provided by the estimated leakage model (parametrized with the same keys and plaintexts).

step, we consider stochastic-based mixture attacks iterating on all the possible mask values, leading to the selection of k maximizing the following equation:

$$\prod_{j=1}^{N_a} \sum_{x_1,...,x_d} \Pr\left[{}^jT \mid x_1,...,x_d, y = \mathsf{f}_k(p_j), x_0 = y + \sum_{i=1}^{d} x_i\right] \times \Pr\left[x_0,...,x_d\right]. \quad (12)$$

In other words, in protected contexts, we define the stochastic-based mixture attacks as a (finite) sum (also known as a mixture) of Gaussian distributions given that $\Pr\left[{}^jT \mid x_1,...,x_d, y = \mathsf{f}_k(p_j), x_0 = y + \sum_{i=1}^{d} x_i\right]$ follows a Gaussian distribution if we consider Gaussian noise. As a result, unlike profiled attacks which ignore the mixture structure, attacks (whether classical template or stochastic-based) which model the mixture structure represent unbiased models (if the leakages contain Gaussian noise and if the stochastic-based model contains the right degree) since such adversaries take into account the multimodal structure of the probability density function associated to each sensitive value y split in several shares.

2.2.3 Profiled correlation power analysis

In 2000, Coron *et al.* [7] and Mayer-Sommer [27] proposed the Correlation Power Analysis (CPA) that recover the key from the target device by selecting the key that maximizes the absolute value of the (Pearson) correlation between the actual leakages and the estimated leakages based on the assumed secret key. Profiled Correlation Power Analysis (PCPA) model the leakage function of each instant from a profiling set with a model. More precisely, PCPA based on template attacks (denoted PCPA-TA) consider that the leakage model associated to the target value y equals to $\hat{\mu}_y$ while PCPA based on stochastic attacks of degree i (denoted PCPA-SAi) assume that the leakage model equals to $\mathsf{h}(y, \hat{\theta})$. As a result, PCPA skip the estimation of the Σ parameter. In the following, we will show that the diagnosis tool also works on score-based profiled attacks by exploiting PCPA.

3 Bias-Variance decomposition of the error rate

In this section, we detail the bias-variance decomposition (of the error rate) used in our decision tool and recently introduced in the side-channel literature by Lerman *et al.* [24].

3.1 Preliminary notions

Domingos showed that the error rate of a classifier can be decomposed into three components [8], [9]: the error rate of the Bayes classifier $\mathsf{R}_b(\cdot)$, the bias $\mathsf{B}(\cdot)$ and the variance $\mathsf{V}(\cdot)$, generally leading to the equality:

$$\begin{aligned} \text{Error rate} =& \mathsf{E}_{\mathcal{T}_{\mathrm{AS}}}\left[c_1 \times \mathsf{R}_b(\mathcal{T}_{\mathrm{AS}})\right] \\ &+ \mathsf{E}_{\mathcal{T}_{\mathrm{AS}}}\left[\mathsf{B}(\mathcal{T}_{\mathrm{AS}})\right] && \text{Bias} \\ &+ \mathsf{E}_{\mathcal{T}_{\mathrm{AS}}}\left[c_2 \times \mathsf{V}(\mathcal{T}_{\mathrm{AS}})\right], && \text{Variance} \end{aligned} \quad (13)$$

where E denotes the mean operator, the two values c_1 and c_2 are in $\mathbb{R}$, and $\mathsf{R}_b(\mathcal{T}_{\mathrm{AS}})$, $\mathsf{B}(\mathcal{T}_{\mathrm{AS}})$ and $\mathsf{V}(\mathcal{T}_{\mathrm{AS}})$ are three functions providing values in $\mathbb{R}$.

The loss function (denoted $\mathsf{LOSS}(k, k')$) represents the cost of predicting k' when the true target value is k. In this paper we consider the zero-one loss function: the cost is zero when k equals to k' and one in the other case.

The *main prediction* (denoted $\mathsf{A}_m(\mathcal{T}_{\mathrm{AS}})$) represents the most frequent prediction on the set of attack traces $\mathcal{T}_{\mathrm{AS}}$ given by the estimated classifier when varying the profiling sets[4]. The bias term represents the difference (according to the loss function) between the main prediction and the prediction provided by the Bayes classifier, i.e.:

$$\mathsf{B}(\mathcal{T}_{\mathrm{AS}}) = \mathsf{LOSS}(\mathsf{A}_m(\mathcal{T}_{\mathrm{AS}}), \mathsf{A}_b(\mathcal{T}_{\mathrm{AS}})). \quad (14)$$

The variance measures the variation of a prediction on a set of attack traces as a function of different profiling sets, i.e.:

$$\mathsf{V}(\mathcal{T}_{\mathrm{AS}}) = \mathsf{E}_{\mathcal{T}_{\mathrm{PS}}}\left[\mathsf{LOSS}(\mathsf{A}_m(\mathcal{T}_{\mathrm{AS}}), \mathsf{A}(\mathcal{T}_{\mathrm{AS}}, \mathcal{T}_{\mathrm{PS}}))\right], \quad (15)$$

where $\mathsf{A}(\mathcal{T}_{\mathrm{AS}}, \mathcal{T}_{\mathrm{PS}})$ represents a profiled attack using the profiling set $\mathcal{T}_{\mathrm{PS}}$ and the attacking set $\mathcal{T}_{\mathrm{AS}}$.

Based on these notations, Domingos demonstrated that the multiplicative factors c_1 and c_2 equal:

$$c_1 = \Pr\left[\mathsf{A} = \mathsf{A}_b\right] - \Pr\left[\mathsf{A} \neq \mathsf{A}_b\right] \times \Pr\left[\mathsf{A} = k \mid \mathsf{A}_b \neq k\right], \quad (16)$$

$$c_2 = \begin{cases} -\Pr\left[\mathsf{A} = \mathsf{A}_b \mid \mathsf{A} \neq \mathsf{A}_m\right] & \mathsf{A}_m \neq \mathsf{A}_b \\ 1 & \mathsf{A}_m = \mathsf{A}_b \end{cases}, \quad (17)$$

where $\mathsf{A} = \mathsf{A}(\mathcal{T}_{\mathrm{AS}}, \mathcal{T}_{\mathrm{PS}})$, $\mathsf{A}_b = \mathsf{A}_b(\mathcal{T}_{\mathrm{AS}})$ and $\mathsf{A}_m = \mathsf{A}_m(\mathcal{T}_{\mathrm{AS}})$.

Figure 1 illustrates the bias-variance decomposition for two profiled attacks. In this figure, attack 1 contains a high bias (since the most frequent prediction given by attack 1 differs from the output of the Bayes classifier) but a smaller variance compared to attack 2.

In the following, we use the terms bias and variance to denote respectively the average of the bias (i.e., $\mathsf{E}_{\mathcal{T}_{\mathrm{AS}}}\left[\mathsf{B}(\mathcal{T}_{\mathrm{AS}})\right]$) and the weighted average of the variance (i.e., $\mathsf{E}_{\mathcal{T}_{\mathrm{AS}}}\left[c_2 \times \mathsf{V}(\mathcal{T}_{\mathrm{AS}})\right]$). As a result, this weighted variance term can be negative according to the value of c_2.

3.2 Estimation of the bias and of the variance terms

In practice, the evaluator has one data source which he randomly splits into two disjoint subsets: a profiling source and an attack source. The first source provides leakages used to build the classifiers while the second source generates leakages to estimate the error rate, the bias and the variance.

Following the proposition of Valentini *et al.* [34], in our experiments, in order to decompose the error rate of an attack $\mathsf{A}(\cdot,\cdot)$ we build a set of distinguishers (of the same complexity[5] and evaluation settings) created with different profiling sets generated by sampling (*with* replacement) the profiling source. This resampling approach provides several distinguishers with different estimated parameters $\hat{\theta}$. We estimate the bias and the variance terms by sampling (*without* replacement) the attack source. More precisely, we average the estimated bias and the estimated variance obtained on each attacking set sampled from the attack source.

4. The generated profiling sets have the same cardinality (e.g., in unprotected contexts, each set contains N_p leakages per target value and n samples per leakage) and they are sampled from the space of sets of leakages.

5. The complexity of an attack is defined by the number of parameters to estimate (as presented in Section 2.2).

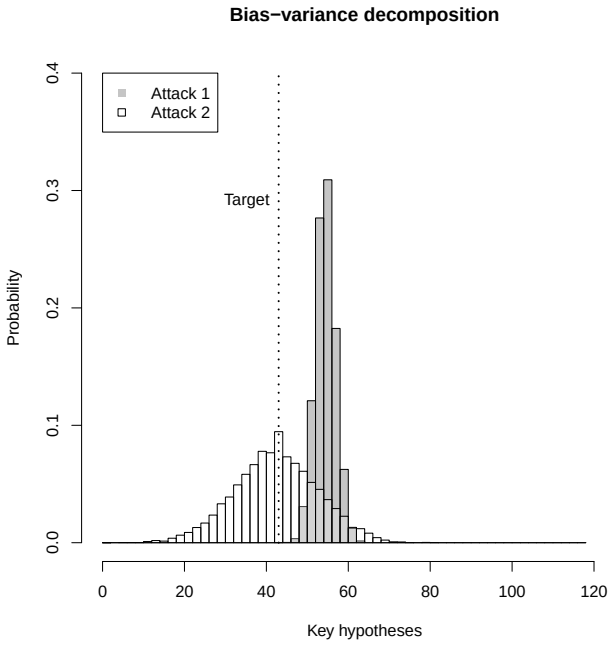


Fig. 1: Illustration of the bias-variance decomposition for two profiled attacks by reporting the probability to output a key value with a fixed attacking set when the actual (target) key value equals to 42. The probability to output a key value depends on the profiling set. For the sake of simplicity, we assume that the error rate of the Bayes classifier equals to zero.

It is worth to note that, in practice, the evaluator does not know which profiling set neither which attacking set will be used by the adversary. As a result, independently of the use of our diagnosis tool, an evaluator requires to build several profiled attacks (with the same complexity and evaluation settings but with different profiling sets) using several attacking sets in order to estimate the error rate. In this paper, we demonstrate that the evaluator can exploit all the constructed profiled attacks in order to obtain a diagnosis tool with a small overhead. Note also that by building more attacks and by increasing the number of attacking sets, we increase the accuracy of estimations of the bias, of the variance and of the error rate.

3.3 Oracle model

In a realistic evaluation scenario, an evaluator has no access to the Bayes classifier (representing the best physical attack minimizing the error rate) during the estimation of the bias and the variance terms. More precisely, the evaluators of crypto systems lack knowledge (1) on the structure of the leakage distributions (leading to assumption errors), and (2) on the best values (i.e., maximizing the success probability) of the estimated parameters used by the classifier (leading to estimation errors). In practice, we counteract this issue by replacing the Bayes classifier by the Oracle model that represents an idealization of the Bayes classifier. The *Oracle model* (also known as the Oracle) denotes an adversary that extracts and outputs the correct key from a set of attack traces regardless of the quantity of information present in the attacking set. In other words, the error rate of the Oracle model equals to zero.

We demonstrate in Section 4 and Section 5 that the Oracle model provides useful information on the bias and variance terms in practice. Furthermore, we rationalize the substitution of the Bayes classifier by the Oracle model in Section 6. Informally, the accuracy of this approach increases with the decrease of the overlap between the true probability density functions of the leakages (related to different keys).

3.4 Diagnosis tool and guidelines

The bias-variance decomposition of the error rate allows to figure out how to tune and to improve an attack. Several ways exist in order to reduce the bias or the variance. In the following, we consider three parameters that can be optimized: (1) the error rate and/or the number of attack leakages, or (2) the number of profiling leakages. For example, an evaluator may vary the number of profiling leakages in order to reach an error rate of 0.2 with 5 attack leakages.

Based on the considered constraints, the evaluator can reduce the variance by increasing the size of the profiling set [2]. Note however that the machine learning theory says that the increase of the profiling set does not impact the bias term (see for example the paper of Domingos [9]). The classifiers having small complexity/variance should be used (e.g., SA1 in unprotected contexts) if the evaluator requires to fix the profiling set to a small size [24]. Furthermore, the bias contribution in the error rate can be reduced by increasing the complexity of the model (potentially at the cost of increasing the variance contribution in the error rate) or by increasing the size of the attacking set [19]. Finally, if the number of points related to the target value increases, the bias term reduces but the variance term can increase (because we have more parameters to estimate) or can reduce (because we increase the quantity of information related to the target value in each leakage) [35]. Table 1 resumes the impact of each parameter on the bias and on the variance terms. Note that, in this paper, we consider DPA scenario (where we target an operation that involves known plaintexts and a fixed key) while other outcomes could be obtained in other settings.

It is worth to note that we reduce the variance of the estimation of metrics (1) by increasing the number of attacking sets provided by the attack source, (2) by increasing the number of estimated distinguishers, and (3) by reducing the noise in the leakage (in order to reduce the error of the assumption that the Bayes classifier equals to the Oracle model).

4 Experiments on unprotected device

In this section, we show how a simple approach (i.e., the bias-variance decomposition) guides the evaluators to find the best profiled attack against an unprotected device. More precisely, the evaluators reduce the error rate of a strategy by reducing the term dominating the error rate.

4.1 Acquisition setup

A set of $80\,000$ power traces was collected on an 8-bit Atmel (ATMega 328) microcontroller at a 16 MHz clock frequency.

	Bias	Variance
↗ size of profiling set	–	↘
↗ complexity of the model	↘	↗
↗ size of the attacking set	↘	?
↗ number of points of interest	↘	?

TABLE 1: Impact of each parameter on the bias and on the variance terms. The symbol ↗ and ↘ represent respectively an increase and a decrease of the term/parameter. The symbol – is used when the term does not change while the symbol ? represents an unknown effect on the term.

The power consumption of the device was measured using an Agillent Infiniium 9000 Series oscilloscope that was set up to acquire 200 MSamples/s. In order to measure the device's power consumption we inserted a 10 Ω resistor placed between the ground pin of the microcontroller and the ground of the power supply. In order to reduce noise in power traces we used averaging (done be the oscilloscope), thus each power trace represents an average of 64 single acquisitions. We separate the dataset of 80 000 power traces into two parts: the profiling source (containing 56 000 traces) and the attack source (containing 24 000 traces).

Our target device executes AES using a constant 128-bit key and random plaintexts. We target the first round of the cipher (manipulating the AES SBox) and focus on the first byte of the key.

4.2 Case studies

We vary five parameters in our experiments: (1) the number of points per trace, (2) the number of profiling traces per target value, (3) the number of attack traces, (4) the type of attack and (5) the leakage degree/complexity. The main purpose is to provide a large quantity of information allowing to play with a large quantity of stories (representing a sequence of steps allowing to reach a final setting starting from one setting) although we exhibit three stories that evaluators may meet against unprotected devices, leaving the other stories to be discovered by the interested readers.

Our experiments test eight popular families of profiled attacks: template attacks, three stochastic attacks (of degrees 1, 2 and 3), as well as profiled correlation power attacks based on template attacks and based on three different stochastic attacks (of degrees 1, 2 and 3).

We consider several scenarios by using three different sizes of the attacking set ($N_a \in \{5, 10, 20\}$), two different sizes of the profiling set ($N_p \in \{3, 10\}$), five different leakage dimensions ($n \in \{1, 2, 5, 10, 20\}$) and two different leakage functions.

We consider a linear feature selection[6] in order to reduce the dimensionality of the problem by selecting points that correlate (linearly) with the Hamming weight of the output of the SBox[7]. This approach allows to simulate two leakage functions: a linear leakage function (when the adversary targets the output of the SBox), and a nonlinear leakage function (when the adversary targets the input of the SBox). It is also worth to note that linear feature selections speed up the execution of the diagnosis tool. However, we advise to use a nonlinear feature selection algorithm (such as the minimum redundancy maximum relevance proposed by Peng *et al.* [29]) in an evaluation case in order to extract all the information available in the leakages.

For each case study, we estimate the parameters of each profiled attack 100 times (with different profiling sets). This number of estimated distinguishers already provides interesting insights on the quantity of bias and variance. An interesting future work can be to find strategies extracting the best number of profiled attacks to build.

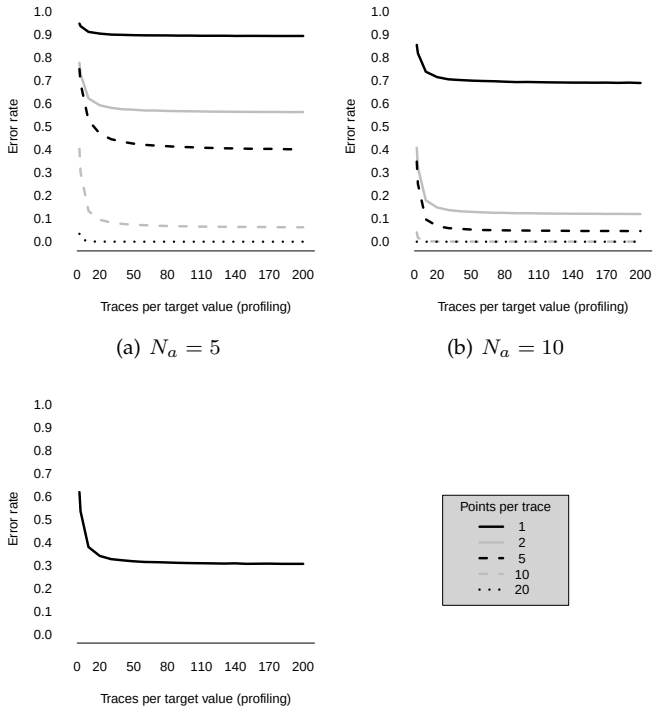
Section 2.2.2 reports that, in unprotected contexts, stochastic attacks of degree strictly less than 8 represent distinguishers with lower complexity than template attacks. As a result, template attacks contain a lower bias (but a higher variance) than stochastic attacks of degree strictly less than 8. This is why we assume in the following that the best profiled attack (that we can mount against unprotected devices) represents a template attack having a large profiling set (that leads to a small variance term). Figure 2 plots the error rate of template attacks as a function of the number of attack traces, the number of points per trace, as well as the number of profiling traces. This figure highlights (1) the lower bound of the error rate of considered attacks, and (2) the difficulty of estimating the actual bias and the actual variance terms due to the assumption that the Bayes classifier equals to the Oracle model. More precisely, Section 6 demonstrates that the accuracy of the bias-variance decomposition increases with the decrease of the error rate of the Bayes classifier. In the case study using 5 attack traces and 1 point per leakage, the best attack that we can mount converges to an estimated error rate of at least 0.89 (which represents the estimated error rate of the Bayes classifier). As a result, the bias-variance decomposition based on the Bayes classifier could significantly differ from the bias-variance decomposition based on the Oracle model (leading to a high estimation error of the decomposition of the error rate). Although small attacking sets can provide inaccurate estimation of the bias and variance terms, the rationale to consider such small sets by evaluators relies on several factors such as the running time of the evaluation (i.e., the smaller the attacking set, the lower the running time) and constraints decided by the executed cryptographic device (e.g., fresh re-keying scheme constraints the size of the attacking set to small values for evaluators). Nevertheless, we demonstrate in the following that an evaluator can still extract information on the contribution of the bias and the variance terms.

4.3 Bias-variance decomposition of probability-based profiled attacks

Tables 2 and 3 provide the bias-variance error rate decomposition of respectively stochastic attacks and template attacks when the adversary targets the output of the SBox (that simulates a linear leakage function). Due to space constraints, we describe only a couple of scenarios. Let assume that the

6. We estimate the Pearson correlation with 1 000 leakages from the profiling source.

7. Note that, during our experiments, the dimensionality reduction algorithm selected points correlated (linearly) the most with the output of the SBox. Nevertheless, this provide points highly correlated to the Hamming weight of the output of the SBox thanks to the high correlation between $\{0, ..., 2^{m-1}\}$ and $\mathsf{HW}(\{0, ..., 2^{m-1}\})$ for small m.



(a) $N_a = 5$ (b) $N_a = 10$

(c) $N_a = 20$ (d) Legend

Fig. 2: Error rate of template attacks as a function of the number of profiling traces per target value, the number of points per trace and the number of attack traces (N_a) for unprotected software implementation.

adversary applies template attacks with 10 attacking traces, 3 profiling traces per target value and 1 point per leakage. The bias reaches a high value (0.66) compared to the variance (0.16) leading to a high error rate (0.82). In order to reduce the bias, the adversary can use template attacks using more points per leakage. The adversary reaches the error rate of 0.25 by using 5 points per trace leading to a lower bias (0.04) but a higher variance (0.21) due to the increase of the number of estimated parameters. In order to reach a lower error rate of 0.10, the attacker can select stochastic attacks of degree 1 leading to a lower variance (0.01) at the cost of a small increase of the bias (0.09) due to the decrease of the complexity of the distinguisher.

Tables 4 shows the bias-variance error rate decomposition of template attacks and stochastic attacks when the adversary targets the input of the SBox while the leakage depends on the output of the SBox (i.e., a simulation of a nonlinear leakage function). Let assume that the adversary first applies stochastic attacks of degree 1 using 5 attack traces, 3 profiling traces and 1 point per leakage. The experiment leads to a high bias of 0.99, a very low variance of 0.00 and an error rate of 0.99. In this case, the adversary can reduce the bias term by using template attacks, which provide a bias of 0.89, a variance of 0.05, and an error rate of 0.94. The high bias of stochastic attacks and template attacks highlights that we should increase the number of points per leakage to, for example, 5 in order to reach a lower bias term of 0.39 leading to a decrease of the error rate to 0.69 (although the variance term increases to 0.29 due to the increase of the complexity of the distinguisher).

	N_a	N_p	n	Error rate composition		
				Bias	Variance	Total
SA1	5	3	1	0.93	0.00	0.94
			2	0.66	0.01	0.67
			5	0.51	0.03	0.54
			10	0.15	0.02	0.17
			20	0.02	0.01	0.02
		10	1	0.93	0.00	0.93
			2	0.66	0.00	0.67
			5	0.52	0.01	0.52
			10	0.15	0.01	0.16
			20	0.02	0.00	0.02
	10	3	1	0.66	0.01	0.67
			2	0.21	0.00	0.21
			5	0.09	0.01	0.10
			10	0.00	0.00	0.00
			20	0.00	0.00	0.00
		10	1	0.66	0.00	0.67
			2	0.21	0.00	0.21
			5	0.09	0.00	0.09
			10	0.00	0.00	0.00
			20	0.00	0.00	0.00
SA2	5	3	1	0.93	0.01	0.94
			2	0.64	0.04	0.68
			5	0.50	0.09	0.58
			10	0.12	0.06	0.18
			20	0.01	0.01	0.02
		10	1	0.93	0.00	0.93
			2	0.65	0.01	0.66
			5	0.50	0.03	0.53
			10	0.12	0.02	0.14
			20	0.01	0.00	0.01
	10	3	1	0.78	0.03	0.81
			2	0.19	0.04	0.23
			5	0.08	0.05	0.12
			10	0.00	0.00	0.00
			20	0.00	0.00	0.00
		10	1	0.78	0.02	0.80
			2	0.19	0.01	0.20
			5	0.07	0.01	0.09
			10	0.00	0.00	0.00
			20	0.00	0.00	0.00

TABLE 2: Error rate decomposition of stochastic attacks of degree 1 and 2 (denoted respectively SA1 and SA2) targeting the output of the SBox of an unprotected software implementation. Each distinguisher uses N_a attack traces, N_p profiling traces per target value, and n points per trace.

4.4 Bias-variance decomposition of score-based profiled attacks

Table 5 shows the bias-variance decomposition of the error rate of score-based profiled attacks based on profiled correlation power analysis exploiting one point per leakage and targeting the output of the SBox. Let assume that the attacker first tests a PCPA-SA1 (i.e., profiled correlation power analysis using stochastic attacks of degree 1) with 10 attacking traces and 3 profiling traces per target value. The strategy exhibits a high bias of 0.82 and a low variance of 0.01 leading to an error rate of 0.83. In order to reduce the error rate, the attacker can change PCPA-SA1 to more complex distinguishers such as PCPA-TA (i.e., profiled correlation power analysis using template attacks). This change decreases the bias term to 0.67 at the cost of an increase of the variance term to 0.17 leading to an increase of the error rate of 0.84. Note that although we increase the error rate, we know that this change can be helpful whether we increase eventually the size of the profiling

	N_a	N_p	n	Error rate composition		
				Bias	Variance	Total
TA	5	3	1	0.89	0.05	0.94
			2	0.54	0.19	0.73
			5	0.40	0.29	0.69
			10	0.06	0.24	0.30
			20	0.00	0.01	0.01
		10	1	0.89	0.02	0.91
			2	0.55	0.07	0.62
			5	0.39	0.14	0.53
			10	0.06	0.07	0.13
			20	0.00	0.00	0.00
	10	3	1	0.66	0.16	0.82
			2	0.11	0.22	0.32
			5	0.04	0.21	0.25
			10	0.00	0.02	0.02
			20	0.00	0.00	0.00
		10	1	0.67	0.07	0.74
			2	0.11	0.07	0.18
			5	0.04	0.05	0.10
			10	0.00	0.00	0.00
			20	0.00	0.00	0.00

TABLE 3: Error rate decomposition of template attacks (denoted TA) targeting the output of the SBox of an unprotected software implementation. Each distinguisher uses N_a attack traces, N_p profiling traces per target value, and n points per trace.

	N_a	n	Error rate composition		
			Bias	Variance	Total
SA1	5	1	0.99	0.00	0.99
		2	0.99	0.00	0.99
		5	0.99	0.00	0.99
	10	1	0.99	0.00	0.99
		2	0.99	0.00	0.99
		5	0.99	0.00	0.99
SA2	5	1	0.99	0.00	0.99
		2	0.97	0.01	0.98
		5	0.97	0.01	0.98
	10	1	0.97	0.01	0.98
		2	0.95	0.01	0.96
		5	0.94	0.03	0.97
SA3	5	1	0.96	0.01	0.97
		2	0.92	0.02	0.94
		5	0.90	0.05	0.95
	10	1	0.92	0.02	0.94
		2	0.80	0.06	0.86
		5	0.74	0.15	0.89
TA	5	1	0.89	0.05	0.94
		2	0.54	0.19	0.73
		5	0.39	0.29	0.69
	10	1	0.65	0.17	0.82
		2	0.11	0.21	0.32
		5	0.04	0.21	0.25

TABLE 4: Error rate decomposition of template attacks (denoted TA) and stochastic attacks of degrees 1, 2 and 3 (denoted respectively SA1, SA2 and SA3) targeting the input of the SBox of an unprotected software implementation. Each distinguisher uses N_a attack traces, 3 profiling traces per target value, and n points per trace.

set to 10 profiling traces per target value. Indeed, this last change decreases the variance to 0.08 (whereas the bias term remains constant) leading to a decrease of the error rate to 0.74. This example highlights that the security metric without its decomposition does not provide enough information on how to improve the efficiency of attacks. Note that an adversary could reduce the bias term by increasing the number of points per leakage, which allows to reduce the error rate of the profiled attacks.

	N_a	N_p	Error rate composition		
			Bias	Variance	Total
PCPA-SA1	10	3	0.82	0.01	0.83
		10	0.82	0.00	0.82
	20	3	0.41	0.01	0.42
		10	0.41	0.01	0.42
PCPA-SA2	10	3	0.80	0.03	0.83
		10	0.81	0.01	0.82
	20	3	0.37	0.07	0.44
		10	0.38	0.02	0.40
PCPA-SA3	10	3	0.77	0.07	0.84
		10	0.78	0.03	0.81
	20	3	0.33	0.13	0.46
		10	0.32	0.05	0.37
PCPA-TA	10	3	0.67	0.17	0.84
		10	0.66	0.08	0.74
	20	3	0.17	0.26	0.43
		10	0.17	0.08	0.25

TABLE 5: Error rate decomposition of profiled correlation power analysis using template attacks (denoted PCPA-TA) and stochastic attacks of degrees 1, 2 and 3 (denoted PCPA-SA1, PCPA-SA2 and PCPA-SA3) targeting the output of the SBox of an unprotected software implementation. Each distinguisher uses N_a attack traces, N_p profiling traces per target value, and 1 points per trace.

Noisy hardware contexts

The previous sections show that an evaluator can exploit our diagnosis tool on unprotected software implementations. This section attests that an adversary can also apply the same diagnosis tool on a more challenging hardware noisy context. More precisely, we employ the public dataset of the second version of the DPAContest[8]. The public dataset contains $1\,640\,000$ traces collected on an FPGA implementing the unprotected AES-128. We consider 70% of the dataset for the profiling source and the reminder for the attack source. We target the Hamming distance between the first byte of the ciphertext and the first byte of the input of the SBox executed during the last round. For the sake of place, Table 6 reports the results only for template attacks and profiled correlation power analysis based on template attacks. Let assume that the evaluator starts with template attacks using $1\,000$ attack traces, 200 profiling traces per target value and 2 points per leakage. The bias term equals to 0.89 while the variance term reaches 0.02 leading to an error rate of 0.91. In order to reduce the error rate, we can reduce the bias term by increasing the number of instants per trace to 50, which provides a smaller bias term of 0.58 but a higher variance term of 0.19 (with a smaller error rate of 0.77). We can reduce the error rate by reducing the variance with a larger profiling set of $2\,000$ profiling traces per target value. This last setting provides an error rate of 0.59 (with a bias of 0.55 and a small variance of 0.03). It is worth to note that the evaluator can reduce the bias term (as well as the error rate) by increasing the attacking set (as shown in the Table 6).

8. http://www.dpacontest.org/v2/

Furthermore, other techniques could be exploited to reduce the error rate such as targeting other (or several) sensitive values and adopting a key enumeration strategy (which was probably applied by best attacks in the DPAContest v2).

	N_a	N_p	n	Error rate composition		
				Bias	Variance	Total
TA	1 000	200	2	0.89	0.02	0.91
			25	0.69	0.12	0.80
			50	0.58	0.19	0.77
			75	0.56	0.23	0.79
		2 000	2	0.90	0.00	0.90
			25	0.68	0.02	0.70
			50	0.55	0.03	0.59
			75	0.53	0.04	0.58
	4 000	200	2	0.72	0.03	0.75
			25	0.41	0.16	0.57
			50	0.34	0.16	0.51
			75	0.31	0.22	0.53
		2 000	2	0.72	0.01	0.73
			25	0.39	0.03	0.41
			50	0.32	0.00	0.32
			75	0.29	0.02	0.31
PCPA-TA	1 000	10	1	0.83	0.09	0.92
		50		0.82	0.04	0.86
		200		0.82	0.01	0.83
		2 000		0.82	0.00	0.82
	4 000	10	1	0.29	0.33	0.62
		50		0.30	0.08	0.38
		200		0.30	0.01	0.32
		2 000		0.29	0.00	0.30

TABLE 6: Error rate decomposition of template attacks (denoted TA) and profiled correlation power analysis using template attacks (denoted PCPA-TA) targeting the Hamming distance between the ciphertext and the input of the last round of an unprotected hardware implementation of AES. Each distinguisher uses N_a attack traces, N_p profiling traces per target value, and n points per trace.

5 Experiments on protected device

This section extends the previous results when considering a more challenging protected (software) implementation. More precisely, we analyze a masking countermeasure representing a well-known technique (due to its theoretical soundness) in order to increase the degree of resilience of an implementation against side-channel attacks. Our diagnosis tool naturally extends to protected devices because our framework decomposes a security metric that can be estimated on any device.

5.1 Acquisition setup

The dataset (of $80\,000$ power traces) measured on the protected device was collected with the same acquisition board used for the unprotected software case studies. The profiling source and the attack source contain respectively $56\,000$ and $24\,000$ leakages. We also applied the same filtering technique (i.e., an average of 64 single acquisitions). The main difference with the previous (unprotected) setting lies in the target implementation. More precisely, this section analyses the implementation of a first-order masked AES SBox (with 2 shares) based on table lookups [30], [32]. This strategy pre-computes a new masked AES SBox in memory (denoted SBox^*) for each execution of the cryptographic algorithm such that:

$$\mathsf{SBox}^*_k\left(x \oplus m_{\mathsf{in}}\right) = \mathsf{SBox}\left(x\right) \oplus m_{\mathsf{out}} \;\; \forall x \in \{0,1\}^8 \qquad (18)$$

for a given pair of input and output mask bytes (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.

As previously, our target device uses a random fixed 128-bit key and random plaintexts. We target the first byte of the key used during the first round of the masked AES.

5.2 Case studies

We vary four parameters in our experiments: (1) the number of points per trace, (2) the number of profiling traces, (3) the number of attack traces and (4) the type of attack. Unlike the unprotected context, we focus here only on two popular profiled attacks in order to highlight the usefulness of our diagnosis tool in protected environments. More precisely, we test template attacks and stochastic-based mixture attacks of degree 1, leaving the other attacks as future work.

Due to the increase of the complexity of the case studies (related to the implemented protection), we increase the size of the attacking set to $N_a \in \{25, 50, 100\}$ and the size of the profiling set to $N_p \in \{20, 40, 60, 80, 100\}$. In our experiment, we also consider two leakage dimensions ($n \in \{2, 5\}$ per share, leading to $\{4, 10\}$ points per leakage) but only one (linear) leakage function. Similarly to the unprotected setting, for each case study, we estimate the parameters of each profiled attack 100 times (with different profiling sets sampled with replacement from the profiling source) in order to estimate the bias, the variance and the error rate.

Section 2.2.2 reports that, in protected contexts, stochastic-based mixture attacks are unbiased models (if the leakages contain Gaussian noise and if the stochastic-based models contain the right degree) since stochastic-based mixture attacks (like any attack exploiting the mixture structure of the density function) take into account the multimodal structure of the density function associated to each value y split in several shares. In order to estimate the complexity of the scenarios based on the collected dataset, Table 7 reports the (estimation) of the lower bound of the error rate of stochastic-based mixture attacks (and of template attacks) by considering a large profiling set (of 400 leakages per target value). The purpose of this table is to show in the following that, even with a high estimation error of the decomposition of the error rate (due to a large difference between the Bayes classifier, represented by the stochastic-based mixture attacks, and the Oracle model), we can extract information on the bias and on the variance.

5.3 Bias-variance decomposition of profiled attacks

Tables 8 and 9 provide the bias-variance error rate decomposition of respectively template attacks and stochastic-based mixture attacks (of degree 1) targeting the output of the SBox implemented with the masking scheme. Let assume that the adversary applies template attacks with 25 attacking traces, 20 profiling traces per target value and 2

	N_a	N_p	n	Error rate
TA	25	400	2	0.90
			5	0.29
	50	400	2	0.69
			5	0.02
	100	400	2	0.28
			5	0.00
SA	25	400	2	0.78
			5	0.05
	50	400	2	0.46
			5	0.00
	100	400	2	0.08
			5	0.00

TABLE 7: Error rate of Template Attacks (TA) and Stochastic-based mixture Attacks (SA) (with N_p traces per target value) targeting the output of the masked SBox as a function of the number of points (n) per share in each trace and the number of attack traces (N_a).

points per share in the leakage. The bias reaches a high value (0.87) compared to the variance (0.11) leading to a high error rate (0.97). In order to reduce the bias, the adversary can increase the number of points per share to 5, leading to a lower bias (0.41) but a higher variance (0.52) due to the increase of the number of parameters, which leads to a slightly lower but still high error rate of 0.93. In order to reduce the bias and the variance terms, the adversary can exploit stochastic-based mixture attacks (at the cost of a higher execution time) leading to a very low bias (0.05), a very low variance (0.00) and a very low error rate (0.06).

Noisy contexts

In order to confirm the soundness of our approach, we also analyze the (previously presented) masking scheme by considering more noisy leakages in which we do not apply the filtering method (that represents averaging of several single acquisitions). The conclusion remains the same except that we need more attack and profiling leakages in order to reach similar success rates. More precisely, if an adversary considers stochastic-based mixture attacks of degree 1 with 5 points per share in each leakage, 1 profiling leakage per target value and 25 attack leakages, then the success rate reaches the value 0.84 with a high bias of 0.74 and a low variance of 0.10. Consequently, an increase of the size of the profiling set should affect only slightly the error rate. For example, by increasing the size of the profiling set to 20 traces per target value, we observe that stochastic-based mixture attacks achieve an error rate of 0.75 still composed of a high bias (0.75) and a low variance (0.00). In order to reduce the error rate, our diagnosis tool advices to reduce the bias term. An attacker achieves this result by increasing the number of points per share in the leakages to 10 in order to reach a lower error rate of 0.59 (with a lower bias of 0.29 but a higher variance of 0.30). In this last scenario the adversary can substantially reduce the error rate by reducing the variance term with a larger profiling set (composed of 20 leakages per target value), leading to a lower error rate of 0.35 (in which the variance equals to 0.03 and the bias equals to 0.33).

	N_a	N_p	n	Error rate composition		
				Bias	Variance	Total
TA	25	20	2	0.87	0.11	0.97
			5	0.41	0.52	0.93
		40	2	0.87	0.08	0.95
			5	0.26	0.51	0.76
		60	2	0.87	0.07	0.94
			5	0.24	0.41	0.65
		80	2	0.87	0.06	0.93
			5	0.23	0.34	0.57
		100	2	0.88	0.04	0.92
			5	0.22	0.30	0.52
	50	20	2	0.67	0.27	0.94
			5	0.08	0.77	0.85
		40	2	0.60	0.28	0.88
			5	0.01	0.46	0.47
		60	2	0.63	0.21	0.84
			5	0.01	0.26	0.27
		80	2	0.62	0.19	0.81
			5	0.01	0.17	0.18
		100	2	0.62	0.16	0.78
			5	0.01	0.12	0.14
	100	20	2	0.29	0.57	0.86
			5	0.00	0.66	0.66
		40	2	0.23	0.46	0.69
			5	0.00	0.12	0.12
		60	2	0.22	0.37	0.59
			5	0.00	0.03	0.03
		80	2	0.22	0.30	0.52
			5	0.00	0.01	0.01
		100	2	0.22	0.26	0.48
			5	0.00	0.00	0.00

TABLE 8: Error rate decomposition of template attacks (denoted TA) targeting the output of the masked SBox. Each distinguisher uses N_a attack traces, N_p profiling traces per target value, and n points per share in the trace.

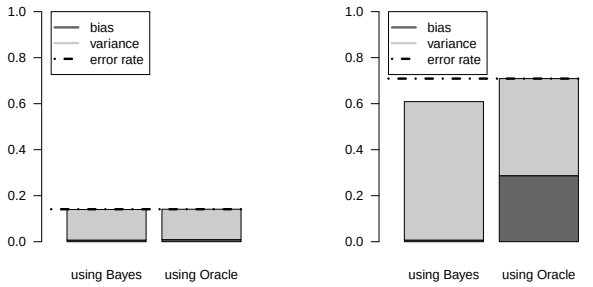
	N_a	N_p	n	Error rate composition		
				Bias	Variance	Total
SA1	25	20	2	0.78	0.00	0.78
			5	0.05	0.00	0.06
	50	20	2	0.46	0.00	0.46
			5	0.00	0.00	0.00
	100	20	2	0.08	0.00	0.07
			5	0.00	0.00	0.00

TABLE 9: Error rate decomposition of stochastic-based mixture attacks of degree 1 (denoted SA1) targeting the output of the masked SBox. Each distinguisher uses N_a attack traces, N_p profiling traces per target value, and n points per share in the trace.

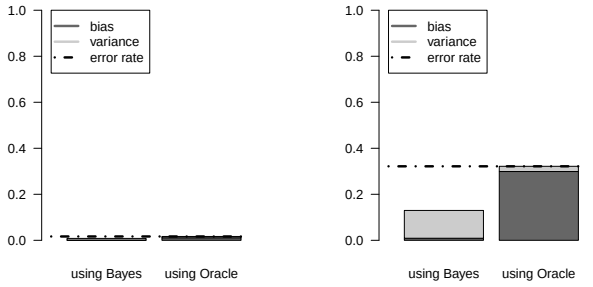
6 VALIDATING ASSUMPTIONS

During an evaluation process, we assume that the evaluator exploits the Oracle model instead of the Bayes classifier in order to estimate the bias and the variance terms. This substitution is necessary (1) due to a lack of information on the Bayes classifier in practice[9], and (2) in order to provide quickly and easily the decomposition of error rates. This section gauges the impact of this assumption on template attacks and on stochastic attacks of degree 1 through: (1) simulated datasets (where we have the Bayes classifier), and (2) the comparison of the decomposition of the error rate

9. In real-world scenarios, the evaluators do not know the probability density function of leakages, which leads to suboptimal profiled attacks compared to physical attacks using the Bayes classifier.



(a) TA with low noise (b) TA with high noise



(c) SA1 with low noise (d) SA1 with high noise

Fig. 3: Error rate, bias and variance estimated with the Bayes classifier and with the Oracle model for Template Attacks (TA) and Stochastic Attacks of degree 1 (SA1) using $N_p = 2$ profiling traces per target value and $N_a = 15$ attack traces. The standard deviation of the noise equals to 3 for high noise and 1 for low noise. The signal-to-noise ratio equals to 2.01 for low noise and 0.22 for high noise. The Bayes classifier has an error rate of 0.012 and 0.301 for respectively low and high noise level. The leakages were generated by the simulator presented in Section 6.1.

based on the Bayes classifier with respect to the decomposition based on the Oracle model. In all our cases, the attacker targets the output of the AES SBox.

6.1 Unprotected contexts

In the unprotected contexts, we generated synthetic leakages having 2 points (denoted ${}_1T$ and ${}_2T$) related to the Hamming weight of the SBox:

$$ {}_1T = \mathsf{HW}\left(\mathsf{SBox}\left(p \oplus k\right)\right) + \epsilon_1, \tag{19}$$

$$ {}_2T = \mathsf{HW}\left(\mathsf{SBox}\left(p \oplus k\right)\right) + \epsilon_2, \tag{20}$$

where ϵ_1 and ϵ_2 represent the (independent) Gaussian noise of the leakage. We estimated the parameters of each classifier $10\,000$ times with different profiling sets containing $N_p = 2$ traces per target value. Each classifier extracts the key based on $N_a = 15$ attack traces in the attacking set and the attack source contains $1\,000$ attacking sets. Figure 3 shows the results by considering low and high noise level.

Figure 3 indicates that the higher the error rate of the Bayes classifier[10], the higher the distance between the decomposition of the error rate based on the Bayes classifier and the Oracle model. The rationale is that the decomposition based on the Oracle model overestimates the success rate of the Bayes classifier leading to an overestimation of the bias term that impacts the estimation of the weight c_2 of the variance term (see Equation (13)).

6.2 Protected contexts

Protected environments represent more complicated contexts for the evaluation of the cryptographic devices. Furthermore, the protected contexts extend the previous section by comparing the bias-variance decomposition based on the Oracle model with respect to the Bayes classifier when one model contains a bias in the classifier. More precisely, template attacks (unlike template-based mixture attacks that take into account the mixture structure of the probability density function) contain bias by assuming that the distribution of leakages follows a (unimodal) Gaussian distribution while, in fact, the distribution follows a multimodal distribution (as shown in the following).

Lets consider simulated leakages where one point ${}_1T$ relates to the Hamming weight of the masked SBox, and one point ${}_2T$ depends on the Hamming weight of the output mask:

$$ {}_1T = \mathsf{HW}\left(\mathsf{SBox}\left(p \oplus k\right) \oplus m_{\mathsf{out}}\right) + \epsilon_1, \tag{21}$$

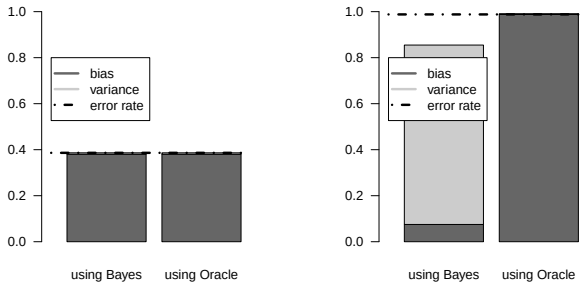
$$ {}_2T = \mathsf{HW}\left(m_{\mathsf{out}}\right) + \epsilon_2. \tag{22}$$

Based on these simulated leakages, Figure 4 shows the bias, the variance and the error rate of template attacks and stochastic-based mixture attacks (of degree 1) using $1\,000$ profiling traces per target value, $N_a = 100$ attack traces with low and high noise. We compute each parameter of each profiled attack $5\,000$ times. The attack source contains 200 different attacking sets. The results point out two important remarks on masked environments when considering the Bayes classifier: (1) an increase of the noise level causes a reduction of the bias term of template attacks, and (2) stochastic-based mixture attacks outperform template attacks thanks to a lower bias and variance terms.

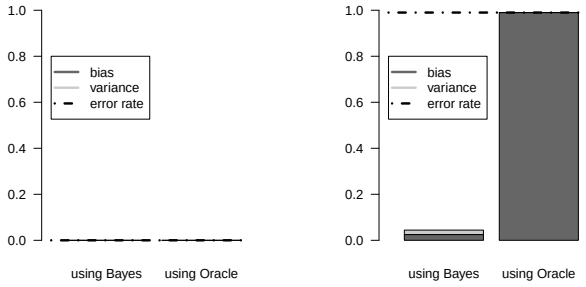
The observation on the variance is expected since stochastic-based mixture attacks have a lower number of parameters to estimate compared to template attacks[11]. Regarding the results on the bias, in a low noise level scenario, the leakages follow a multimodal distribution (i.e., a mixture of several Gaussian distributions) and can be accurately modeled by stochastic-based mixture attacks (as well as by template-based mixture attacks). In a high noise level setting, the leakages can be represented by a unimodal distribution (e.g., one Gaussian distribution) and can be accurately modeled by template attacks (representing one Gaussian distribution per target value) as well as by stochastic-based mixture attacks. Figure 5 illustrates this argument by plotting one leakage distribution (estimated with the same simulator described in Equations (21) and (22)) per target value and for two noise levels. In other words, compared to profiling attacks which ignore the mixture structure, attacks (whether classical template or stochastic-based attacks) which model the mixture structure

10. Several parameters impact the error rate of the Bayes classifier such as the noise level, the number of informative points per leakage and the size of the attacking set.

11. Template attacks estimate one covariance matrix per target value while stochastic-based mixture attacks compute only one covariance matrix for all the target values.

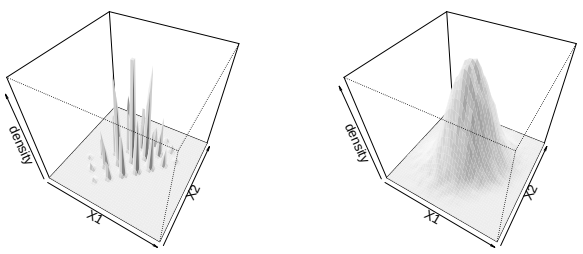


(a) TA with low noise (b) TA with high noise

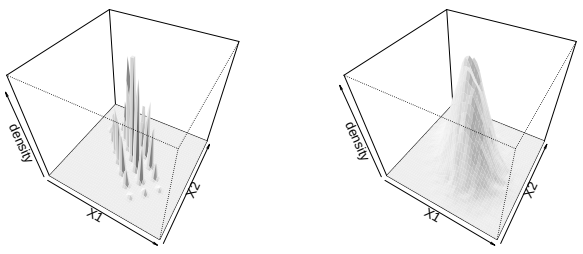


(c) SA1 with low noise (d) SA1 with high noise

Fig. 4: Error rate, bias and variance estimated with the Bayes classifier and with the Oracle model for Template Attacks (TA) and Stochastic-based mixture Attacks of degree 1 (SA1) using $N_p = 1\,000$ profiling traces per target value and $N_a = 100$ attack traces. The standard deviation of the noise equals to 3 for high noise and 0.001 for low noise. The leakages were generated by the simulator presented in Section 6.2.



(a) $\mathsf{SBox}\,(p \oplus k) = 3$; $\sigma = 0.01$ (b) $\mathsf{SBox}\,(p \oplus k) = 3$; $\sigma = 1$



(c) $\mathsf{SBox}\,(p \oplus k) = 95$; $\sigma = 0.01$ (d) $\mathsf{SBox}\,(p \oplus k) = 95$; $\sigma = 1$

Fig. 5: Leakage distribution associated to a target value (denoted $\mathsf{SBox}\,(p \oplus k)$) as well as to a noise level (with a standard deviation denoted σ) when varying the mask value (m_{out}). In each simulated trace, the instants $X1$ and $X2$ relate to $\mathsf{HW}\,(\mathsf{SBox}\,(p \oplus k) \oplus m_{\mathsf{out}})$ and to $\mathsf{HW}\,(m_{\mathsf{out}})$.

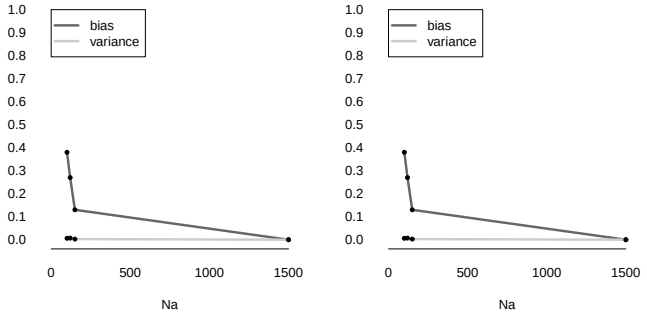
have higher abilities to fit the structure of the multimodal leakage distributions (that leads to a higher complexity and a smaller bias for stochastic-based mixture attacks than for template attacks in protected contexts). However, this ability decreases as a function of the noise level.

An important difference between theoretical results and real case studies is the exploitation of the Oracle instead of the Bayes. Section 6.1 reveals that the use of the Oracle in unprotected contexts leads to an overestimation of the bias term and an underestimation of the variance term. The same phenomenon appears in protected contexts as plotted in Figures 4 for template attacks and for stochastic-based mixture attacks. The error rate of stochastic-based mixture attacks provides an estimation of the error rate of the Bayes. The higher this error rate, the stronger the degradation of the estimation of the bias and the variance terms (due to the simplification provided by the Oracle model). More precisely, the difference between the estimated components of the error rate based on the Oracle and based on the Bayes is small when the Bayes has a low error rate. Moreover, the difference between the two estimations increases as a function of the increase of the error rate of the Bayes.

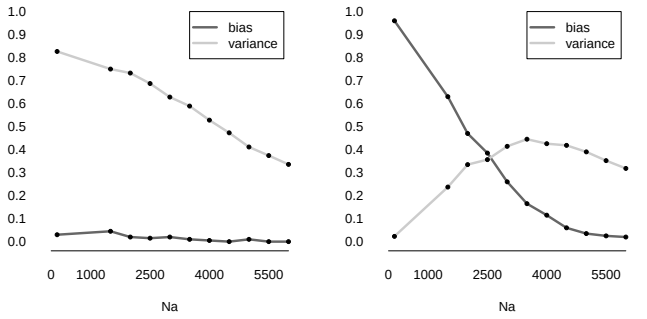
6.3 Can the diagnosis tool be useful in practice?

Section 6.1 and Section 6.2 indicate that an evaluator can accurately estimate the bias and variance as long as the distance between the error rates of the Bayes and the Oracle is small. In a low noise level setting (in which the Bayes equals to the Oracle) the bias impacts the error rate heavily which forces the evaluator to select the best profiled attack (thanks to an accurate estimation of the bias through the Oracle model). However, the higher the noise level, the higher the difference between the error rates of the Bayes classifier and the Oracle model, and the higher the estimation errors of the bias and the variance are. However, in protected contexts, Section 6.2 exhibits that, in high noise level settings, all probability-based profiled attacks have a similar bias term computed from the Bayes classifier (i.e., the *actual* bias term), which leads essentially to variance problems.

We can reduce the estimation errors of the bias and the variance by increasing (1) the number of informative points, (2) the number of attack traces, and (3) any parameter providing key-related information to the Bayes classifier. Figures 6 and 7 illustrate this phenomenon by reporting the estimation of the bias and variance for respectively template attacks and stochastic-based mixture attacks based on the Oracle and based on the Bayes as a function of the number of attack leakages. The traces represent measurements on a masked implementation presented in Section 6.2. We use $1\,000$ profiling traces per target with a low and a high noise. We compute each parameter of each attack $5\,000$ times. The attack source contains 200 different sets. The figure shows that the distance between the estimated terms based on the Oracle and based on the Bayes increases as a function of the noise. However, this distance decreases as a function of the number of attack traces.

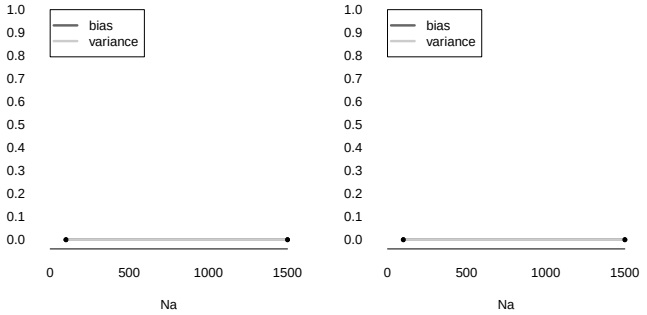


(a) Estimated with Bayes and low noise (b) Estimated with Oracle and low noise

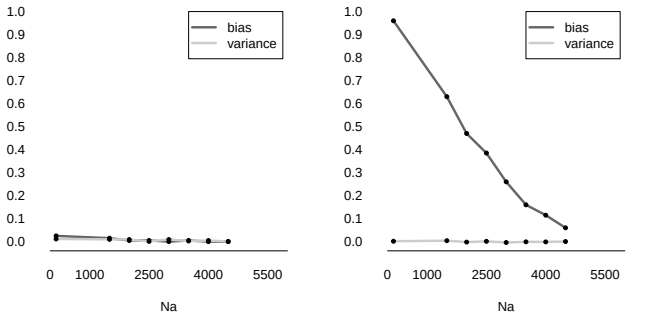


(c) Estimated with Bayes and high noise (d) Estimated with Oracle and high noise

Fig. 6: Error rate, bias and variance estimated with the Bayes and with the Oracle for template attacks using $N_p = 1\,000$ profiling traces per target as a function of the number of attack traces (N_a). The standard deviation of the noise equals to 3 for high noise and 0.001 for low noise. The leakages were generated by the simulator presented in Section 6.2.



(a) Estimated with Bayes and low noise (b) Estimated with Oracle and low noise



(c) Estimated with Bayes and high noise (d) Estimated with Oracle and high noise

Fig. 7: Error rate, bias and variance estimated with the Bayes classifier and with the Oracle model for stochastic-based mixture attacks of degree 1 using $N_p = 1\,000$ profiling traces per target value as a function of the number of attack traces (N_a). The standard deviation of the noise equals to 3 for high noise and 0.001 for low noise. The leakages were generated by the simulator presented in Section 6.2.

Concretely, all our experiments report that our diagnosis tool can be exploited in practice when the attacking set contains enough information on target, i.e., when the bias-variance decomposition based on our Oracle is *good enough*. Additionally, we showed that there are systematic ways to improve the quality of this decomposition, even if the leakage model is imperfect. In fact, the only condition we strictly need is that the leakage model from which the evaluator starts is sound (in the sense of the work presented at Eurocrypt 2009 [33]), i.e., that it asymptotically leads to successful key recoveries. There remains the problem that evaluators cannot always know in advance how accurate is the bias-variance decomposition. We can also observe that even in case the bias-variance decomposition is poorly approximated by the Oracle (as in Figure 6 and in Figure 7), we can extract relevant intuitions about how to improve the evaluations by comparing the decompositions of two attacks with increasing complexities (and number of attack traces). Note also that a large number of (security) metrics (e.g., the error rate and the guessing entropy) lack of accuracy in high noise contexts. So overall the presented tools can be viewed as an ingredient in order to (more) rapidly refine the quest for the best physical evaluation of a given implementation.

7 Conclusions

In a theoretical point of view, the diagnosis tool (based on the Bayes classifier) specifies the source of failure of an attack (e.g., a high bias or a high variance). In practice, the evaluators lack knowledge on the Bayes classifier. The main contribution of this paper lies on a practical instantiation of the diagnosis tool with the Oracle (that always outputs the right target value). As a result, on the one hand, based on this diagnostic, the evaluators can decide what to apply in order to increase the success of attacks. For example, an increase of the complexity of the attack should significantly affect the error rate if the diagnostic tool detects a high bias. On the other hand, based on this diagnostic, the evaluators can also decide what should not be applied. For example, an increase of the size of the profiling set slightly affects the error rate if the diagnostic tool detects a low variance. As a future work, we will focus on the best choice from a set of possibilities (that depends on the term impacting the most the error rate) after the report of the diagnostic.

The diagnosis tool generates a set of profiling and attacking sets to estimate the bias and the variance. However, the exploitation of the diagnosis tool leads to a small overhead compared to an approach estimating only the error rate. The rationale is that, independently of the use of the diagnosis tool, an evaluator requires to build several profiled attacks (with the same complexity and evaluation settings but with different profiling sets) using several attacking sets in order to limit problems related to overfitting (leading to biased

estimations of the error rate)[12]. In other words, the diagnosis tool exploits all the data that was already generated anyway (i.e., the profiling models and the attacking sets) in order to extract more information than only the error rate.

The main limitation of the presented tool represents the accuracy of the diagnostic. More precisely, the greater the quantity of information in the attacking set the better the estimation of the diagnostic. In a practical point of view, all our experiments report that the presented tool can be efficiently exploited to report diagnostics about attacks on unprotected and protected implementations in hardware and in software. The reason lies on the small difference between the Oracle and the Bayes. Future works will focus on the accuracy of the tool on devices executing high order masking schemes (in which the Oracle differs significantly from the Bayes), and on a better estimation of the error rate of the Bayes classifier. For example, the error rate of the Bayes classifier could be the minimum error rates (computed with theoretical metrics such as the success exponent [18]) found from a set of profiled attacks.

Finally, we envision to apply the tool on dimensionality reduction algorithms in order to understand how to discover the best attack during an evaluation process.

ACKNOWLEDGMENTS

This research was partly supported by the Brussels Region INNOVIRIS project SCAUT and by the EU H2020 project REASSURE (Grant agreement 731591).

REFERENCES

[1] Mehdi-Laurent Akkar, Régis Bevan, Paul Dischamp, and Didier Moyart. Power analysis, what is now possible... In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 489–502. Springer, 2000.

[2] Damien Brain and Geoffrey I. Webb. On the effect of data set size on bias and variance in classification learning. In D. Richards, G. Beydoun, A. Hoffmann, and P. Compton, editors, *Proceedings of the Fourth Australian Knowledge Acquisition Workshop (AKAW '99)*, pages 117–128, Sydney, 1999. The University of New South Wales.

[3] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Wiener [37], pages 398–412.

[4] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.

[5] Marios O. Choudary and Markus G. Kuhn. Efficient stochastic methods: Profiled attacks beyond 8 bits. In Marc Joye and Amir Moradi, editors, *Smart Card Research and Advanced Applications - 13th International Conference, CARDIS 2014, Paris, France, November 5-7, 2014. Revised Selected Papers*, volume 8968 of *Lecture Notes in Computer Science*, pages 85–103. Springer, 2014.

[6] Omar Choudary and Markus G. Kuhn. Efficient template attacks. In Francillon and Rohatgi [13], pages 253–270.

[7] Jean-Sébastien Coron, Paul C. Kocher, and David Naccache. Statistics and secret leakage. In Yair Frankel, editor, *Financial Cryptography, 4th International Conference, FC 2000 Anguilla, British West Indies, February 20-24, 2000, Proceedings*, volume 1962 of *Lecture Notes in Computer Science*, pages 157–173. Springer, 2000.

12. We refer the readers to the book of Hastie *et al.* that presents models assessments based on the estimation of the error rate [19].

[8] Pedro Domingos. A unified bias-variance decomposition and its applications. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 231–238. Morgan Kaufmann, 2000.

[9] Pedro Domingos. A unified bias-variance decomposition for zero-one and squared loss. In Henry A. Kautz and Bruce W. Porter, editors, *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA.*, pages 564–569. AAAI Press / The MIT Press, 2000.

[10] François Durvaux, François-Xavier Standaert, and Santos Merino Del Pozo. Towards easy leakage certification. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 40–60. Springer, 2016.

[11] François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to certify the leakage of a chip? In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 459–476. Springer, 2014.

[12] Paul N. Fahn and Peter K. Pearson. IPA: A new class of power attacks. In Koç and Paar [20], pages 173–186.

[13] Aurélien Francillon and Pankaj Rohatgi, editors. *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, volume 8419 of *Lecture Notes in Computer Science*. Springer, 2014.

[14] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

[15] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 444–461. Springer, 2014.

[16] Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Koç and Paar [20], pages 158–172.

[17] Vincent Grosso, François-Xavier Standaert, and Emmanuel Prouff. Low entropy masking schemes, revisited. In Francillon and Rohatgi [13], pages 33–43.

[18] Sylvain Guilley, Annelie Heuser, and Olivier Rioul. A key to success - success exponents for side-channel distinguishers. In Alex Biryukov and Vipul Goyal, editors, *Progress in Cryptology - INDOCRYPT 2015 - 16th International Conference on Cryptology in India, Bangalore, India, December 6-9, 2015, Proceedings*, volume 9462 of *Lecture Notes in Computer Science*, pages 270–290. Springer, 2015.

[19] Trevor J. Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[20] Çetin Kaya Koç and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*. Springer, 1999.

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

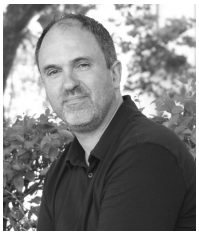
[22] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Wiener [37], pages 388–397.

[23] Liran Lerman, Gianluca Bontempi, and Olivier Markowitch. Side Channel Attack: an Approach Based on Machine Learning. In *Second International Workshop on Constructive SideChannel Analysis and Secure Design*, pages 29–41. Center for Advanced Security Research Darmstadt, 2011.

[24] Liran Lerman, Gianluca Bontempi, and Olivier Markowitch. The bias-variance decomposition in profiled attacks. *J. Cryptographic Engineering*, 5(4):255–267, 2015.
[25] Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. Breaking cryptographic implementations using deep learning techniques. In Claude Carlet, M. Anwar Hasan, and Vishal Saraswat, editors, *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*, volume 10076 of *Lecture Notes in Computer Science*, pages 3–26. Springer, 2016.
[26] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.
[27] Rita Mayer-Sommer. Smartly analyzing the simplicity and the power of simple power analysis on smartcards. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 78–92. Springer, 2000.
[28] Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni. Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In Daniel J. Bernstein and Tanja Lange, editors, *Progress in Cryptology - AFRICACRYPT 2010, Third International Conference on Cryptology in Africa, Stellenbosch, South Africa, May 3-6, 2010. Proceedings*, volume 6055 of *Lecture Notes in Computer Science*, pages 279–296. Springer, 2010.
[29] Hanchuan Peng, Fuhui Long, and Chris H. Q. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238, 2005.
[30] Emmanuel Prouff and Matthieu Rivain. A generic method for secure sbox implementation. In Sehun Kim, Moti Yung, and Hyung-Woo Lee, editors, *Information Security Applications, 8th International Workshop, WISA 2007, Jeju Island, Korea, August 27-29, 2007, Revised Selected Papers*, volume 4867 of *Lecture Notes in Computer Science*, pages 227–244. Springer, 2007.
[31] Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.
[32] Kai Schramm and Christof Paar. Higher order masking of the AES. In David Pointcheval, editor, *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*, volume 3860 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2006.
[33] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.
[34] Giorgio Valentini and Thomas G. Dietterich. Low bias bagged support vector machines. In Tom Fawcett and Nina Mishra, editors, *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 752–759. AAAI Press, 2003.
[35] Peter van der Putten and Maarten van Someren. A bias-variance analysis of a real world learning problem: The coil challenge 2000. *Machine Learning*, 57(1-2):177–195, 2004.
[36] Carolyn Whitnall, Elisabeth Oswald, and François-Xavier Standaert. The myth of generic dpa...and the magic of learning. In Josh Benaloh, editor, *Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings*, volume 8366 of *Lecture Notes in Computer Science*, pages 183–205. Springer, 2014.
[37] Michael J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.

Liran Lerman Liran Lerman received the PhD degree in the department of Computer Science at the Université libre de Bruxelles (in Belgium) in 2015. In 2010, he received with honors (grade magna cum laude) the master degree from the same university. During his PhD thesis, he was a teaching assistant and a student doing research as part of a Machine Learning Group (MLG) and the Cryptography and Security Service (QualSec). Currently, he is a post-doctoral researcher of the QualSec. His research relates to machine learning, side-channel attacks and countermeasures.

Nikita Veshchikov Nikita Veshchikov got his Bachelor in Computer Sciences in 2009 at Université Libre de Bruxelles (ULB) in Belgium. He continued studies in the same field and got a Master in Computer Sciences with advanced studies of embedded systems in 2011 at the same university. During his master thesis he studied reverse engineering and anti-patching techniques. Since 2011 Nikita works as a teaching assistant and works on his thesis in the field of side-channel attacks. His is mostly interested in simulators and automated tools for side-channel analysis.



Olivier Markowitch Olivier Markowitch is professor at the Université libre de Bruxelles. He is teaching algorithmics and cryptography courses. He is leading the QualSec Research Group who is active in the fields of side channel attacks and cryptographic protocols. He is currently at the head of the Computer Science Department.

François-Xavier Standaert François-Xavier Standaert was born in Brussels, Belgium in 1978. He received the Electrical Engineering degree and PhD degree from the Universiteé catholique de Louvain, respectively in 2001 and 2004. In 2004-2005, he was a Fulbright visiting researcher at Columbia University, Department of Computer Science, Network Security Lab and at the MIT Medialab, Center for Bits and Atoms. In 2006, he was a founding member of IntoPix s.a. From 2005 to 2008, he was a post-doctoral researcher of the UCL Crypto Group and a regular visitor of the two aforementioned laboratories. Since 2008, he is associate researcher of the Belgian Fund for Scientific Research (F.R.S.-FNRS) and professor at the UCL Institute of Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM). In 2010, he was program co-chair of CHES (which is the flagship workshop on cryptographic hardware). In 2011, he was awarded a Starting Independent Research Grant by the European Research Council. In 2016, he has been awarded a Consolidator Grant by the European Research Council. From 2017 to 2020, he will be board member (director) of the International Association for Cryptologic Research (IACR). His research interests include cryptographic hardware and embedded systems, low power implementations for constrained environments (RFIDs, sensor networks, ...), the design and cryptanalysis of symmetric cryptographic primitives, physical security issues in general and side-channel analysis in particular.