J Cryptogr Eng
DOI 10.1007/s13389-015-0106-1

REGULAR PAPER

The bias–variance decomposition in profiled attacks

Liran Lerman[1,2] · Gianluca Bontempi[2] · Olivier Markowitch[1]

Received: 9 July 2014 / Accepted: 13 June 2015

Abstract The profiled attacks challenge the security of cryptographic devices in the worst case scenario. We elucidate the reasons underlying the success of different profiled attacks (that depend essentially on the context) based on the well-known bias–variance tradeoff developed in the machine learning field. Note that our approach can easily be extended to non-profiled attacks. We show (1) how to decompose (in three additive components) the error rate of an attack based on the bias–variance decomposition, and (2) how to reduce the error rate of a model based on the bias–variance diagnostic. Intuitively, we show that different models having the same error rate require different strategies (according to the bias–variance decomposition) to reduce their errors. More precisely, the success rate of a strategy depends on several criteria such as its complexity, the leakage information and the number of points per trace. As a result, a suboptimal strategy in a specific context can lead the adversary to overestimate the security level of the cryptographic device. Our results also bring warnings related to the estimation of the success rate of a profiled attack that can lead the evaluator to underestimate the security level. In brief, certify that a chip leaks (or not) sensitive information represents a hard if not impossible task.

Keywords Side-channel attack · Bias–variance decomposition · Profiled attack · Machine learning · Stochastic attack · Template attack

✉ Liran Lerman
llerman@ulb.ac.be

[1] Quality and Security of Information Systems, Département d'informatique, Université Libre de Bruxelles, Brussels, Belgium

[2] Machine Learning Group, Département d'informatique, Université Libre de Bruxelles, Brussels, Belgium

1 Introduction

Encryption methods aim to protect sensitive information handled by cryptographic devices while attackers challenge this protection. Historically, the encryption algorithms provide security against adversaries searching link between plaintext, ciphertext and the secret key. During the last decade, the literature showed that an attacker can target the implementation of the cryptographic scheme rather than its abstract representation. In this paper, we focus on side-channel attacks in which the adversary exploits physical leakage information from the cryptographic device such as the power consumption [29], the processing time [28] and the electromagnetic emanation [18].

Since the seminal work of Kocher [28], the evolution of techniques challenging the security of cryptographic devices has been characterized by an increase in the complexity of the statistical analysis. A few years later, Chari et al. [7] introduced profiled attacks as the strongest leakage analysis in an information theoretic sense. Also profiled attacks are particularly effective (1) when the adversary is only able to observe a single use of the key (e.g., in stream ciphers), (2) when the target instruction manipulates only the target value (e.g., a LOAD instruction applied on the secret key) and (3) when the cryptographic algorithm is unknown.

Profiled attacks include several approaches like template attack (TA) [7], stochastic attack (SA) [44], multivariate regression attack [46] and machine learning attack [1,24–26,31–35]. The evaluation criteria of these attacks are based on several metrics such as the number of measurements to find the secret information, the interpretability of the model, the success rate, the computational efficiency (i.e. the time needed to learn and to attack as well as the memory used), the guessing entropy and the information theoretic metric [45].

Published online: 27 June 2015

This paper focuses on the success rate, commonly used in previous side-channel attacks literature.

To estimate the success rate of a strategy, a classical approach estimates the parameters of a model (e.g., template attack) with a learning set collected on a (fully) controlled device and similar to the target one. Afterward, the adversary applies the model on a test set that represents a set of leakage information measured on the target device. The security level of the cryptographic device is estimated according to the success rate of the model to retrieve the secret key value.

Gierlichs et al. [20] presented empirical comparisons between stochastic attack and template attack based on datasets collected on two microcontrollers. They concluded that template attack outperforms stochastic attack during the attacking phase when there is enough traces in the profiling phase while stochastic attack (with a right degree) is the method of choice otherwise. However, the true leakage distribution was unknown and, as a result, the conclusion related to empirical results.

Machine learning developed a statistical formalism to characterize and to study formally the error rate of models: the bias–variance decomposition of the generalization error. In 1992, German et al. introduced this decomposition to the regression cases [19]. Later, in 2000, Domingos generalized the bias–variance decomposition to the classification cases [11,12]. The bias–variance formalism decomposes the error rate of a predictive model into three terms: the bias, the variance and the noise. As a result, to improve an attack, an adversary should apply a technique addressing the term dominating the error rate. In other words, several models, that have the same error rate, may require different strategies (according to the bias–variance decomposition) to reduce their errors. The bias–variance analysis also explains (1) why there is no universally optimal learning method for all possible contexts and (2) why some simple learner (e.g., stochastic attack of degree 1) can outperform powerful model (e.g., neural network with 1,000,000 neurons).

Whitnall et al. [49] analyzed formally the efficacy and the efficiency of template attack with respect to stochastic attack. More precisely, they focused on the accuracy of the approximations (as well as its impact during the attacking phase) of the data-dependent deterministic part of an univariate leakage, leaving multivariate analysis (i.e. using multiple points from a leakage trace) and noise estimation part as further works. They used several metrics in order to show that stochastic attack requires less (or equal) traces during the profiling phase while the efficiency of its attacking phase varies in function of the degree of the leakage function. We aim to take a further step by analyzing (based on simulations) multivariate noisy contexts (1) as it is commonly encountered in side-channel analysis and (2) as the most side-channel challenging concerns the high dimensionality nature of the data. We also show that two metrics used by Whitnall et al. (that relate to what we call the variance term and the bias term) represent the two weighted components (among the three that we present in this paper) of the success rate.

Recently, Durvaux et al. [13] presented a similar result for the mutual/perceived information metric and called it "*the estimation and the assumption errors*" of template attack and stochastic attack. Our paper aims to show (1) how to decompose the error rate of profiled (and non-profiled) attacks based on the bias–variance decomposition of the generalization error, and (2) how to reduce the error rate of profiled (and non-profiled) attacks based on the bias–variance diagnostic. The main reason to consider the generalization error instead of the mutual/perceived information metric is that most machine learning models and non-profiled attacks rate key candidates according to scores rather than probabilities. This prevents the computation of probabilities based metrics.

We make a detailed assessment of the presented metric by considering a large variety of contexts, ranging from a linear, quadratic and random leakage information to a low, medium and high level of noise. All experiments rely on datasets of different sizes created with a simulator allowing a fully control of the leakage information. We also provide several guidelines when analyzing cryptographic devices.

The rest of the paper is organized as follows. Section 2 discusses side-channel attacks and several profiled attacks. Section 3 introduces the bias–variance decomposition of the generalization error of an attack. Section 4 presents the experimental results on a large number of contexts, discusses the results and, eventually, gives guidelines for devices evaluations. Section 5 concludes this paper with several perspectives of future works.

2 Side-channel attacks

2.1 Preliminaries

In this paper, we assume that the adversary wants to retrieve the secret key value used when the cryptographic device (executing a known encryption algorithm) encrypts known plaintexts. To find the secret key, an adversary targets a key-related information $y_i \in \mathcal{Y}$ where $\mathcal{Y} = \{y_0, y_1, \ldots, y_{Y-1}\} = \{0, 1\}^{l_0}$ is denoted the set of classes.

During the execution of an encryption algorithm, the cryptographic device processes a function f (also known as a sensitive variable [42])

$$\begin{aligned} & f : \mathcal{P} \times \mathcal{O} \rightarrow \mathcal{Y} \\ & y_i = f_O(p), \end{aligned} \tag{1}$$

where

- $O \in \mathcal{O} = \{O_0, O_1, \ldots, O_{K-1}\} = \{0,1\}^{l_1}$ is a key-related information and l_1 is the size of the secret value used in f (e.g., one byte of the secret key).
- $p \in \mathcal{P} = \{p_0, p_1, \ldots, p_{P-1}\} = \{0,1\}^{l_2}$ represents a public information, l_2 is the size of the public value used in f (e.g., one byte of the plaintext) and P is the cardinality of $\mathcal{P}$.
- i is a number related to O and p.

Note that the values of $l_0,\ l_1$ and l_2 depend on the cryptographic algorithm and the device architecture.

Various cryptographic schemes use highly non-linear functions f because algorithms that are close to a linear function are susceptible to various (theoretical but powerful) attacks such as differential cryptanalysis [2] and linear cryptanalysis [37]. Intuitively, the higher the distance between a function f and a set of affine functions, the higher the non-linearity of f (we refer to [38] for an introduction to non-linear functions in cryptography). However, Prouff [41] highlighted that non-linear functions (used in a cryptographic algorithm) are less robust against side-channel attacks than linear functions and, recently, Heuser et al. [23] emphasized that the robustness of a function against side-channel attack is not directly linked to its non-linearity but rather to its resistance against differential cryptanalysis. As a result, usually, adversary targets non-linear functions f. An example of non-linear function f is the substitution box (SBox) of a block-cipher, e.g.,

$$y_i = f_O(p) = \mathrm{SBox}(p \oplus O), \tag{2}$$

where $\oplus$ is the bitwise exclusive-or.

Let

$${}^{j}T_i = \left\{ {}^{j}_{t}T_i \in \mathbb{R} \mid t \in [1;n] \right\} \tag{3}$$

be the j-th physical leakage information (called trace) associated to the target value y_i and ${}^{j}_{t}T_i$ be the leakage value at time t of the j-th trace associated to the target value y_i. We denote $\mathcal{T}$ a set of traces and $\mathbb{T}$ the set of all possible traces (i.e. $\mathcal{T} \subseteq \mathbb{T}$). We model the leakage information ${}^{j}_{t}T_i$ of the device at time t as a function of y_i such that

$${}^{j}_{t}T_i = {}_{t}L\left(y_i\right) + {}^{j}_{t}\epsilon_i, \tag{4}$$

where ${}^{j}_{t}\epsilon_i \in \mathbb{R}$ is the noise of the trace ${}^{j}_{t}T_i$ following a Gaussian distribution with zero mean and ${}_{t}L$ is the leakage model at time t. Examples of functions ${}_{t}L$ are the identity, the Hamming weight (HW) [36] and the weighted sum of bits of the target value y_i.

2.2 Profiled attacks

The profiled attack strategy represents an efficient attack by leakage estimations. It estimates (with a set of traces called learning set) a template $\Pr\left[{}^{j}T_i \mid y_i\right]$ for each target value during the profiling step (also known as learning step). The learning set (denoted $\mathcal{T}_{\mathrm{LS}} \subseteq \mathbb{T}$) is measured on a controlled device similar to the target chip. In our experiments, we limited to consider the same cryptographic device. We refer the interested reader to [14,39] that study practical issues when the controlled and the target devices differ.

Once a template is estimated for each target value, during the attacking step the adversary classifies a new trace T (measured on the target device) using the posteriori probability returned by a model $A(T)$

$$\begin{aligned} \hat{y} = A(T) &= \underset{y_i \in \mathcal{Y}}{\arg\max}\ \Pr\left[y_i \mid T\right] && (5)\\ &= \underset{y_i \in \mathcal{Y}}{\arg\max}\ \frac{\Pr\left[T \mid y_i\right] \times \Pr\left[y_i\right]}{\Pr\left[T\right]} && (6)\\ &= \underset{y_i \in \mathcal{Y}}{\arg\max}\ \Pr\left[T \mid y_i\right] \times \Pr\left[y_i\right]. && (7) \end{aligned}$$

In practice, the adversary uses an estimation of $\Pr[T \mid y_i]$ and $\Pr[y_i]$ (i.e. $\hat{\Pr}[T \mid y_i; \hat{\theta}_i]$ and $\hat{\Pr}\left[y_i\right]$ where θ_i is the parameter of the probability density function and the a priori probabilities $\hat{\Pr}\left[y_i\right]$ are estimated by frequency counts).

Let $\mathbb{T}$ be the set of all possible traces and $\mathcal{T}_{\mathrm{TS}}$ be a testing set of traces (where $\mathcal{T}_{\mathrm{TS}} = \left\{{}^{1}T, \ldots, {}^{N}T\right\}$ and ${}^{j}T \in \mathbb{R}^n\ \forall j \in \{1; N\}$). If a set $\mathcal{T}_{\mathrm{TS}} \subseteq \mathbb{T}$ of traces for a constant secret key is available, the adversary classifies this set by using the equation (or the log-likehood rule)

$$\hat{y} = \underset{y_i \in \mathcal{Y}}{\arg\max} \prod_{j=1}^{N} \hat{\Pr}\left[{}^{j}T \mid y_i; \hat{\theta}_i\right] \times \hat{\Pr}\left[y_i\right]. \tag{8}$$

Several approaches exist to estimate the probability $\Pr\left[{}^{j}T_i \mid y_i\right]$ such as the parametric template attack [7], the stochastic attack [44], the multivariate regression model [46] and the non-parametric machine learning models [25,31].

Template attacks Template attacks [7] assume that $\Pr\left[{}^{j}T_i \mid y_i\right]$ follows a Gaussian distribution for each target value, i.e.

$$\begin{aligned} \Pr\left[{}^{j}T_i \mid y_i\right] &\simeq \hat{\Pr}\left[{}^{j}T_i \mid y_i; \hat{\mu}_i, \hat{\Sigma}_i\right] && (9)\\ &= \frac{\mathrm{e}^{-\frac{1}{2}({}^{j}T_i - \hat{\mu}_i)\hat{\Sigma}_i^{-1}({}^{j}T_i - \hat{\mu}_i)^{\top}}}{\sqrt{(2\pi)^n \det(\hat{\Sigma}_i)}}, && (10) \end{aligned}$$

where $\det(\Sigma)$ denotes the determinant of the matrix Σ while $\hat{\mu}_i \in \mathbb{R}^n$ and $\hat{\Sigma}_i \in \mathbb{R}^{n \times n}$ are, respectively, the sample mean and the sample covariance matrix of the traces associated to the target value y_i. In what follows we will assume that the noise is independent of the target value. This property allows to estimate the same covariance matrix Σ for all the target values.

The complexity of template attack (i.e. the number of parameters to estimate) depends on the number of points per trace: the higher the number of points per trace, the higher the complexity of template attack. More precisely, template attack requires to estimate $2^{l_0} \times n + \frac{n \times (n+1)}{2}$ parameters ($2^{l_0} \times n$ for the expected values and $\frac{n \times (n+1)}{2}$ for the covariance matrix).

Stochastic attacks Stochastic attacks [44] (also known as linear regression attacks) model the leakage information at time t as a function of the secret value y_i with a regression model h spanned by U functions g_u (where $u \in [1; U]$), i.e.

$$ {}^j_t T_i = h\left(y_i, {}_t\theta\right) + {}^j_t\epsilon_i \tag{11}$$

$$ = {}_t c + \sum_{u=1}^{U} {}_t\alpha_u \; g_u\left(y_i\right) + {}^j_t\epsilon_i, \tag{12}$$

where ${}^j_t\epsilon_i \in \mathbb{R}$ is a residual Gaussian noise at time t on the j-th trace associated to the target value y_i, ${}_t\theta = \{{}_t c, {}_t\alpha_1, \ldots, {}_t\alpha_U\} \in \mathbb{R}^{U+1}$ is the parameter of the regression model h at time t and $\{g_1, \ldots, g_U\}$ is the basis used in the regression. Stochastic attack assumes that g_u is a monomial of the form $\prod_{b \in \mathcal{B}} \mathrm{Bit}_b\left(y_i\right)$ where $\mathrm{Bit}_b\left(y_i\right)$ returns the b-th bit of y_i and $\mathcal{B} \subset \{1, 2, \ldots, l_0\}$. Note that $\mathrm{Bit}_b\left(y_i\right)$ raised to the e-th power (where $e > 0$) equals to $\mathrm{Bit}_b\left(y_i\right)$.

The *degree of a monomial* equals the sum of all the exponents of its variables. For example, the monomial $\mathrm{Bit}_1\left(y_i\right) \times \mathrm{Bit}_5\left(y_i\right)$ has a degree two (there are two combined bits) while $\mathrm{Bit}_2\left(y_i\right) \times \mathrm{Bit}_4\left(y_i\right) \times \mathrm{Bit}_7\left(y_i\right)$ has a degree three (there are three combined bits). The *degree of a model* h equals the highest degree of its monomials (with a coefficient different from zero) that compose h. Furthermore, the *complexity* of a model equals its degree. For example, the following model has a degree two (there are two monomials, one of degree two and the other of degree one):

$$h\left(y_i, {}_t\theta\right) = 0.3 \times \mathrm{Bit}_1\left(y_i\right) \times \mathrm{Bit}_2\left(y_i\right) + \mathrm{Bit}_3\left(y_i\right). \tag{13}$$

It is worth to note that, with the previous definition of stochastic attack, the *maximal degree* of the function h is l_0 and, in this case, h contains the monomial $\prod_{b=1}^{l_0} \mathrm{Bit}_b\left(y_i\right)$ with a coefficient different from zero. Furthermore, $\frac{n \times (n+1)}{2} + n \times \sum_{i=0}^{l_0} \binom{l_0}{i} = \frac{n \times (n+1)}{2} + n \times 2^{l_0}$ represents the number of parameters of the stochastic attack with the maximal degree.

The ordinary least squares method allows to estimate the parameter ${}_t\theta$ by minimizing the sum of squared vertical distances between the traces and the responses predicted by the function h, i.e.

$$ {}_t\hat{\theta} = \arg\min_{{}_t\theta} \sum_{i=0}^{Y-1} \sum_{j=1}^{N} \left({}^j_t T_i - h\left(y_i, {}_t\theta\right) \right)^2. \tag{14}$$

Then, the attacker assumes that $\Pr\left[{}^j T_i \mid y_i\right]$ follows the Gaussian distribution $\mathcal{N}\left(h\left(y_i, \theta\right), \Sigma\right)$ where $h(y_i, \theta)$ represents the vector $\{h(y_i, {}_1\theta), h(y_i, {}_2\theta), \ldots, h(y_i, {}_n\theta)\}$ and $\Sigma \in \mathbb{R}^{n \times n}$ is the covariance matrix of the residual term. An extended version of stochastic attack removes the profiling step [10]. However, this approach is out of the scope of this work.

Reduction of stochastic attacks to template attack This section aims to show that template attack is a specific case of stochastic attack. More precisely, template attack represents a stochastic attack with the maximal degree.

The traces associated to the target value y_i are considered to follow the Gaussian distribution $\mathcal{N}\left(\mu_i, \Sigma\right)$ by template attack and $\mathcal{N}\left(h(y_i, \theta), \Sigma\right)$ by stochastic attack. As a result, the main difference between template attack and stochastic attack is that the first returns μ_i for each target value y_i while the second returns $h(y_i, \theta)$.

Let $E[X]$ be the expected value of the random variable X and $E[X|Y]$ be the conditional expectation of X given Y. Then, stochastic attack (using the ordinary least squares estimator) selects the parameter $\hat{\theta}$ of the regression model h that minimizes

$$\hat{\theta} = \arg\min_{\theta} E_i\left[\, E_j\left[\left\| {}^j T_i - h\left(y_i, \theta\right) \right\|\right]\right], \tag{15}$$

where $\|x\|$ represents the Euclidean norm of x. It is well known that the whole expression is minimized if

$$h\left(y_i, \hat{\theta}\right) = E_j\left[{}^j T_i \mid y_i\right] \tag{16}$$

(see for example [48].)

As a result, stochastic attack estimates $E_j\left[{}^j T_i \mid y_i\right]$ to minimize Eq. 15 while template attack estimates the same value (see Eq. 9).

Suppose that $E_j\left[{}^j T_i \mid y_i\right]$ differs for each target value y_i. Template attack is unaffected by this assumption. However, the model h of the stochastic attack needs to return one different estimated value for each target value y_i. This is fulfilled when the model h has the highest degree complexity. Note that template attack and stochastic attack estimate the same number of parameters: $\frac{n \times (n+1)}{2} + n \times 2^{l_0}$ parameters as seen previously.

2.3 Evaluation criteria

In side-channel attack, an adversary aims to maximize the effectiveness of attacks. An adversary estimates the performance of a model based on several metrics such as the number of measurements in order to find the secret information, the success rate to return the right target value, the computational efficiency (i.e. the time needed to learn and to attack as well as the memory used), the guessing entropy and the information theoretic metric [45]. These metrics give a global evaluation of an attack as well as the degree of resilience of a cryptosystem against side-channel attacks but lack of interpretability. When an attack fails, it is important to understand why the models do not work and how to improve the strategy. Several explanations are possible such as (1) it is due to a lack of data in the learning set, (2) it is due to a lack of interesting points in the traces, or (3) we need to consider another profiled attack.

This paper focuses on the success rate based on a single test trace though our strategy can easily be extended to several test traces. The main reason to consider the success rate is that most machine learning models and non-profiled attacks rate key candidates according to scores rather than probabilities. This prevents the computation of other probabilities based metrics.

Several papers provide explanation about which term influences the success rate of specific attacks (see for examples [15,41]). We aim to propose a decomposition of the success rate called the bias–variance decomposition of the error rate. This approach can be applied to any profiled and non-profiled attack. Furthermore, the decomposition includes all the involved elements, notably the cryptographic algorithm, the physical implementation aspect, and the attack strategy.

According to the bias–variance diagnostic, an adversary can apply different techniques depending on which term dominates the error rate. In other words, several models, with the same error rate, may require different strategies (according to the bias–variance decomposition) in order to have their errors reduced. The bias–variance decomposition allows also to explain why simple learner (e.g., stochastic attack of degree 1) can outperform powerful model (e.g., neural network).

3 Bias–variance decomposition as a metric for profiled attack

3.1 Preliminaries

Side-channel attack represents a classification problem. However, for the sake of simplicity, we first provide an intuition of the bias–variance formalism applied to the regression problem.

The notion of model complexity is strongly related to the accuracy of a predictive model. Vapnik and Chervonenkis introduced the VC dimension to describe model complexity (see for example [21]). Intuitively, the complexity of a regression model relates to the number of its parameters. The higher the number of parameters, the greater the flexibility of the model. For example, a linear regression $y = \alpha x + \beta$ with two parameters $\{\alpha, \beta\} \in \mathbb{R}^2$ can describe only relationships of degree one, while a quadratic regression $y = \alpha_1 x + \alpha_2 x^2 + \beta$ with three parameters $\{\alpha_1, \alpha_2, \beta\} \in \mathbb{R}^3$ can describe relationships of degree one and two. As a result, a naive approach would be to always choose the most complex model, so that we are sure to be able to approximate a wide range of problems. However, in practice, we have a learning set with a fixed size such that an increase of the number of parameters leads to an increase of the error of fitting. In a regression context, the error of fitting represents the sum of squares of differences between the true target value and the estimated target value. As a result, an adversary is confronted to two possibilities: (1) the model requires a higher complexity in order to be able to approximate the target function and (2) the model requires a lower complexity because the target function has a lower degree and the size of the learning set is fixed.

Figure 1 shows three linear regression models (with different degrees) estimated with a learning set of size 10 extracted from a noisy target function. The regression model of degree three fits "*well*" the target function while the regression models of degree one is too simple (it has a high error of fitting on the learning and testing set) and the regression model of degree seven is too complex (it has a low error of fitting on the learning set but a high error of fitting on a testing set). Figure 2 shows the error of fitting on the learning set and on the test set by varying the complexity of a linear regression model. In the cases where the model has a high error of fitting on a test set, we say that the models are not able to generalize to new data.

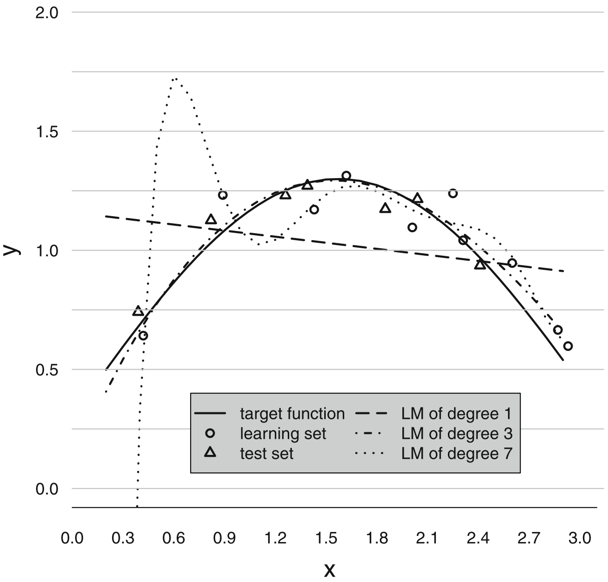


Fig. 1 Three linear regressions (LM) with different degrees estimated with a learning set of size 10 extracted from the noisy target function

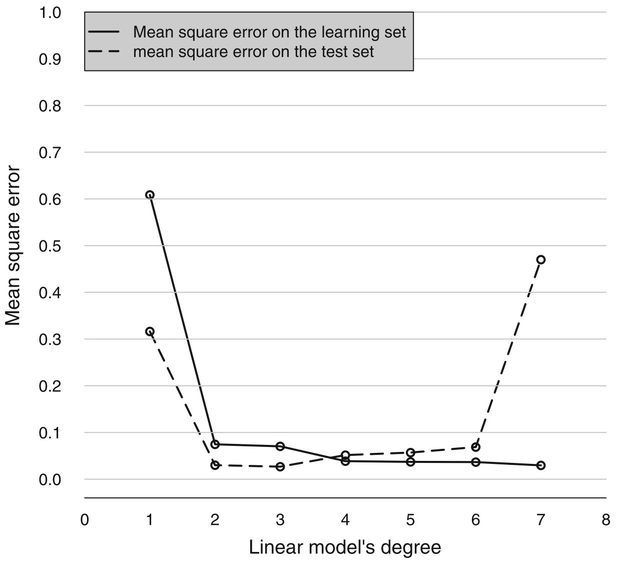


Fig. 2 Error of fitting on a learning set and on a test set of a linear regression models with different degrees estimated with a learning set of size 10 extracted from a noisy target function

Therefore, an adversary is confronted to the following dilemma: if the model is too simple, the model may not fit the examples available in the learning set while if the model is too complex, it will be able to fit exactly the examples available in the learning set but at the cost of a high error of fitting on a test set. In the first case, we say that the model underfits the data while in the second case it overfits.

To avoid the underfitting and the overfitting effects we need to understand the key factors that influence the error of fitting of a model on a test set. The bias–variance decomposition allows to study how the success rate depends on the following parameters

- the number of traces in the learning set,
- the noise level in the traces,
- the number of points per trace,
- the number of relevant points per trace,
- the number of irrelevant points per trace,
- the complexity of the model and
- the number of target values.

3.2 Bias–variance decomposition for classification

Let $\hat{y} = A({}^jT_i; \mathcal{T}_{\text{LS}})$ be the prediction on a test trace ${}^jT_i$ (associated to the target value y_i) of a model A using the learning set $\mathcal{T}_{\text{LS}}$. In a classification problem that considers a first-order success rate, the loss function $L(y_i, y_j)$ represents the cost of predicting y_j when the true target value is y_i. In this paper, we consider the zero–one loss function: the cost is zero when y_i equals y_j and one in the other cases, i.e.

$$L(y_i, y_j) = \begin{cases} 0 & y_j = y_i \\ 1 & \text{otherwise} \end{cases}. \tag{17}$$

The Mean Misclassification Error rate (MME) measures how well a learning model generalizes to unseen data. Mathematically, the MME equals the expected value of the loss function on a test set with a fixed size ($\mathcal{T}_{\text{TS}}$), i.e.

$$\text{MME} = E_{{}^jT_i \in \mathcal{T}_{\text{TS}}}\left[L(y_i, A({}^jT_i; \mathcal{T}_{\text{LS}}))\right]. \tag{18}$$

More precisely, the MME represents the probability that a learning model A (using a specific learning set $\mathcal{T}_{\text{LS}}$) returns a wrong target value associated to a trace from a specific test set $\mathcal{T}_{\text{TS}}$.

In a general case, the MME depends on the random nature of the learning set with a fixed size and of the testing set. We remove the mentioned dependency by averaging over training and testing sets and we call it the mean integrated misclassification error rate (MIME), i.e.

$$\text{MIME} = E_{{}^jT_i}\left[E_{\mathcal{T}_{\text{LS}}}\left[L(y_i, A({}^jT_i; \mathcal{T}_{\text{LS}}))\right]\right]. \tag{19}$$

Several approaches exist to decompose this value [4,5,9,11,12,17,22,27,30,47]. We consider the decomposition proposed by Domingos [11,12] that can be generalized to several loss functions, leaving analysis of other decomposition approaches as further work.

Domingos decomposes the MIME in three weighted components: the noise (N), the bias (B) and the variance (V), i.e.

$$\begin{aligned} \text{MIME} &= E_{{}^jT_i}\left[E_{\mathcal{T}_{\text{LS}}}\left[L(y_i, A({}^jT_i; \mathcal{T}_{\text{LS}}))\right]\right] && (20)\\ &= E_{{}^jT_i}\left[c_1 \times N({}^jT_i)\right] \quad \text{Noise} && (21)\\ &\quad + E_{{}^jT_i}\left[B({}^jT_i)\right] \quad \text{Bias} \\ &\quad + E_{{}^jT_i}\left[c_2 \times V({}^jT_i)\right], \quad \text{Variance} \end{aligned}$$

where $\left\{c_1, c_2, N({}^jT_i), B({}^jT_i), V({}^jT_i)\right\} \in \mathbb{R}^5$. Domingos proved this decomposition (see Theorem 2 of [11]) in the general multi-class problem for zero-one loss function.

Let $A_b({}^jT_i)$ be the *optimal prediction* returned by the Bayes classifier on the test trace ${}^jT_i$. The Bayes classifier represents a classification model that minimizes the probability of misclassification, i.e.

$$A_b({}^jT_i) = \underset{y_i \in \mathcal{Y}}{\text{argmax}} \Pr\left[{}^jT_i \mid y_i\right] \times \Pr\left[y_i\right]. \tag{22}$$

As a result, the Bayes classifier requires the knowledge of the true probability density function of $\Pr\left[{}^jT_i \mid y_i\right]$.

The noise term represents the unavoidable component of the error rate, incurred independently of the learning algorithm nor on the training set and due to the random nature of the phenomenon. The noise term represents mathematically

$$N({}^{j}T_{i}) = L\left(y_i, A_b\left({}^{j}T_{i}\right)\right). \tag{23}$$

Let $A_m({}^{j}T_{i})$ be the *main prediction* that represents the most frequent prediction of the analyzed model on a test trace ${}^{j}T_{i}$. The model can vary its prediction due to the random nature of the learning set. The bias term represents the difference (according to the loss function) between the main prediction and the optimal prediction. This value ranges between zero and one: (1) zero if on average over all the training set the test trace is correctly predicted, and (2) one if on average over all the training set the test trace is incorrectly predicted. In other words, the bias measures the systematic error of a learner (i.e. how accurate a model is across different training sets). Mathematically the bias term equals

$$B({}^{j}T_{i}) = L(A_m({}^{j}T_{i}), A_b({}^{j}T_{i})). \tag{24}$$

The variance term equals the average loss incurred by predictions relative to the main prediction. It measures the variation of a prediction on a test set in function to different training sets (i.e. how sensitive the learning model is to changes in training set). As a result, the variance is independent of the true target value: it equals zero for a learning model that returns the same target value regardless of the learning set. Mathematically, the variance term equals

$$V({}^{j}T_{i}) = E_{\mathcal{T}_{\mathrm{LS}}}\left[L(A_m({}^{j}T_{i}), A({}^{j}T_{i}; \mathcal{T}_{\mathrm{LS}}))\right]. \tag{25}$$

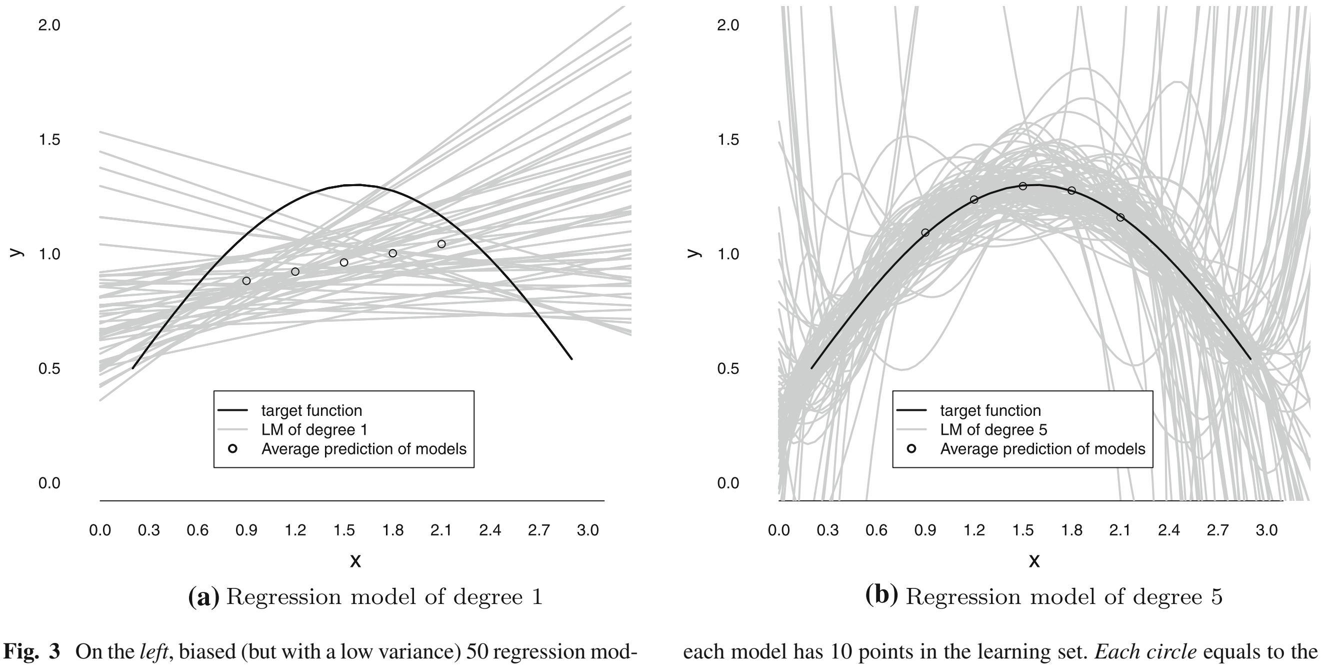
Finally, Domingos demonstrated that the multiplicative factors c_1 and c_2 equal:

$$c_1 = \Pr[A = A_b] \tag{26}$$
$$-\Pr[A \neq A_b] \times \Pr[A = y_i \mid A_b \neq y_i],$$

$$c_2 = \begin{cases} -\Pr[A = A_b \mid A \neq A_m] & A_m \neq A_b \\ 1 & A_m = A_b \end{cases}, \tag{27}$$

where $A = A({}^{j}T_{i})$, $A_b = A_b({}^{j}T_{i})$ and $A_m = A_m({}^{j}T_{i})$.

Figure 3 illustrates the bias–variance decomposition of the mean square error of regression models of degree 1 and 5 while the target function has a degree 3. The figure plots 50 regression models for each case using a learning set of 10 points. The complex model (of degree 5) has a low bias (i.e., on average over the learning set, the model fits the target function) but a high variance (i.e., the model highly varies as a function of the learning set) while the simple model has a high bias (i.e., on average over the learning set, the model does not fit the target function) but a low variance (i.e., the model slightly varies as a function of the learning set). The simplest model takes less into account the learning set and, as a result, this model has the smallest variance (i.e., the model does not vary much as a function of the learning set) but a high bias. On the other hand, the most complex model fits the learning set and, as a result, the model has the highest variance (i.e., the model highly varies as a function of the learning set) but a low bias.



(a) Regression model of degree 1

(b) Regression model of degree 5

Fig. 3 On the *left*, biased (but with a low variance) 50 regression models of degree 1 and on the *right* unbiased (but with a high variance) 50 regression models of degree 5. The target function has a degree 3 and each model has 10 points in the learning set. *Each circle* equals to the average prediction of regression models

3.3 Discussion

A complex model has a high variance and a low bias while a simple model has a low variance and a high bias. Equation 21 shows that we need to have low bias and low variance since both contribute to the error rate. Geman et al. [19] show that there is a tension between these goals and called it the *bias–variance tradeoff*. This tradeoff stipulates that on the one hand the purpose of the profiling step is to minimize the error rate on the learning set (resulting in a low bias) but on the other hand more sensitivity to the learning set results in a high variance. As a result, there is a tradeoff to be made to minimize the error rate.

Equation 21 reveals also that as long as c_2 is negative, the error rate can decrease by increasing the variance. As a result, high variance does not necessarily result in high error rate. Indeed, maximal bias and zero variance give to a maximal error rate. Therefore, unstable model may be beneficial on biased cases. At the same time, high variance does not necessarily lead to a low error rate. In fact, the optimal level of variance should be selected according to the bias term (that influences the multiplicative factor c_2). In other words, there is a strong interaction effect between the variance and the bias in the error rate of a side-channel attack.

Note that we consider the bias–variance decomposition applied to a classification problem but the bias–variance decomposition can also be applied to a regression problem such as in a stochastic attack context.

3.4 How to estimate the bias and the variance?

The bias and the variance represent theoretical terms that cannot be computed in a realistic setting. Indeed, the adversary requires the knowledge of the (unknown) probability density function of $\Pr[T \mid y_i]$ to estimate the Bayes classifier.

For this reason, we consider a simulated environment where a number of training and testing sets can be created for the sake of the analysis. This approach allows to understand what impacts the error rate of profiled attacks and how to increase the success rate based on the results of this analysis.

In practice, we create several learning sets (the higher the number of learning sets, the better). Then, we generate a large testing set (the larger the size of the testing set, the better). Each trace follows a known Gaussian distribution that allows to estimate the Bayes classifier. These sets allow to estimate the bias, the variance, the noise and the multiplicative factors.

4 Experiments and discussion

We detail in this section the results of our experiments of the bias–variance decomposition applied to template attack and to stochastic attack. The purpose is to understand what factor impacts the success rate of a profiled model.

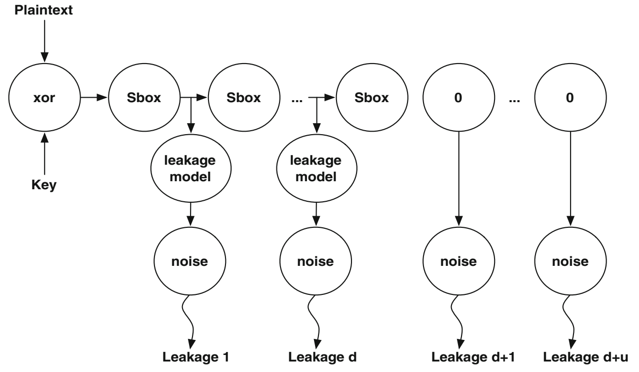


Fig. 4 Scheme of the traces generator

4.1 Target implementation

We considered a variety of realistic scenarios based on

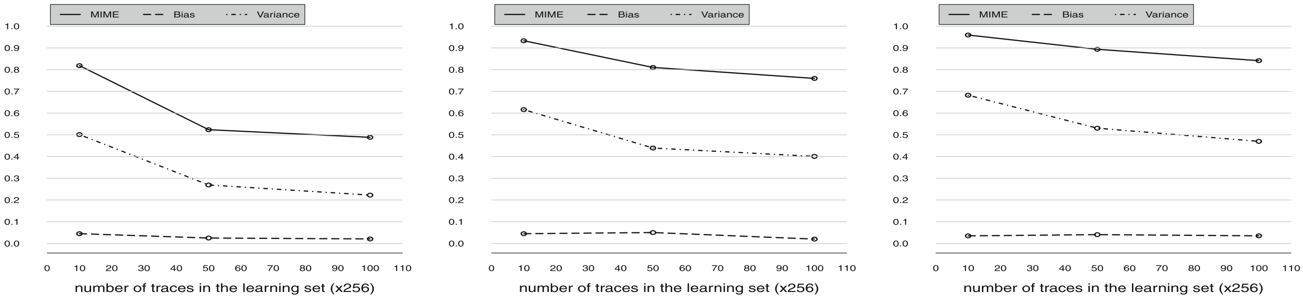
- a linear, quadratic and random leakage model,
- a low, medium and high noise level,
- different numbers of informative points,
- different numbers of uninformative points and
- different numbers of traces in the learning set.

As illustrated in Fig. 4 each trace has d leakages related to the output of a non-linear function (called Sbox) and u noisy leakages following a Gaussian distribution with zero mean. The standard deviation and the mean of the leakage models are respectively one and zero. We use the Sbox of the Advanced Encryption Standard (AES) as non-linear function. Finally, the low, the medium and the high noise level have respectively a variance of 1, 2 and 3 that leads to a signal-to-noise ratio (SNR) of 1, $\frac{1}{2}$ and $\frac{1}{3}$.

4.2 Template attack

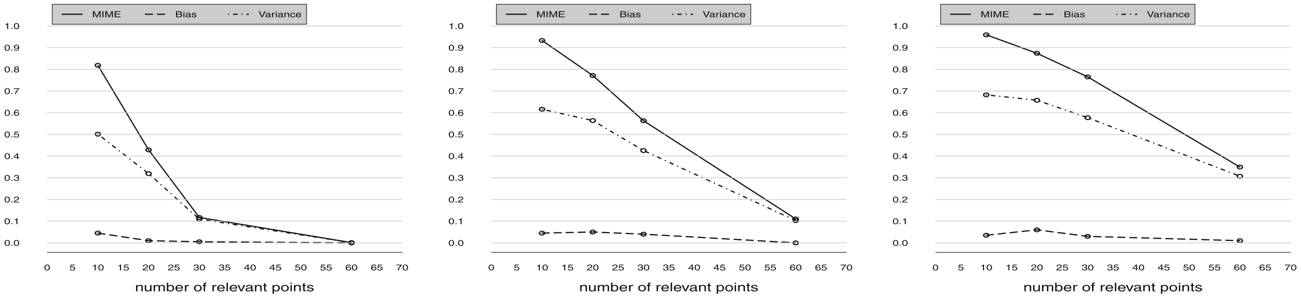
Figures 5, 6, 7 and 8 show the MIME, the bias and the variance when increasing respectively the number of traces in the learning set (Np), the number of relevant points (d), the number of irrelevant points (u) and the leakage model. As expected, we reduce the error rate (1) by increasing the number of traces in the learning set, (2) by increasing the relevant points, (3) by reducing the irrelevant points and (4) by reducing the noise level. Furthermore, an increase of the relevant points has a higher impact on the error rate than a reduction of the irrelevant points.

It is interesting to remark that the variance has a higher impact on the error rate than the bias. This result suggests that template attack has a high complexity and, as a result, this



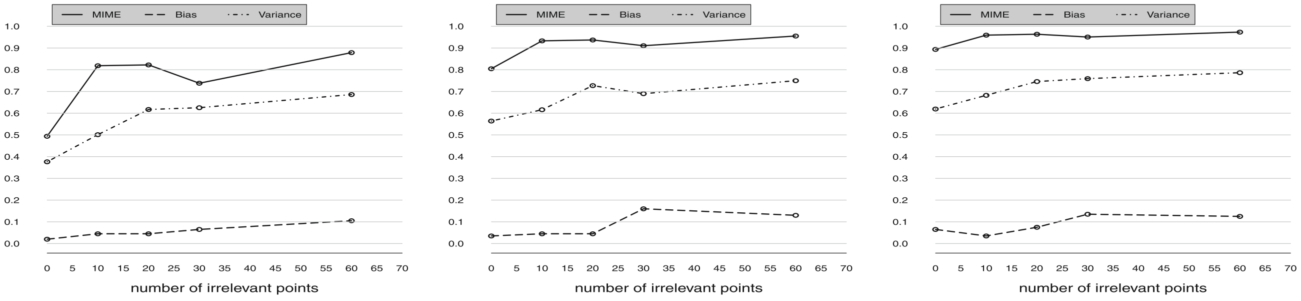
(a) SNR equals 1 **(b)** SNR equals $\frac{1}{2}$ **(c)** SNR equals $\frac{1}{3}$

Fig. 5 MIME, bias and variance of a template attack as a function of the number of the number of traces in the learning set. There are ten informative points per trace, ten non-informative points per trace and a random leakage



(a) SNR equals 1 **(b)** SNR equals $\frac{1}{2}$ **(c)** SNR equals $\frac{1}{3}$

Fig. 6 MIME, bias and variance of a template attack as a function of the number of informative points per trace. There are 10 × 256 traces in the learning set, 10 non-informative points per trace and a random leakage



(a) SNR equals 1 **(b)** SNR equals $\frac{1}{2}$ **(c)** SNR equals $\frac{1}{3}$

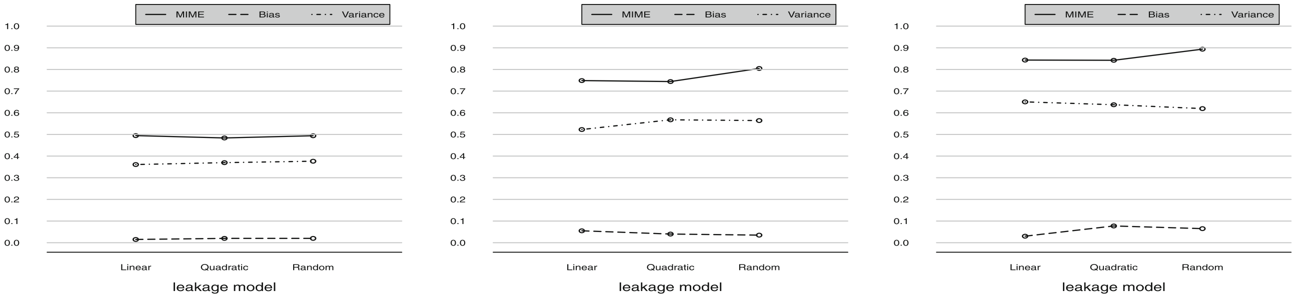
Fig. 7 MIME, bias and variance of a template attack as a function of the number of non-informative points per trace. There are 10 × 256 traces in the learning set, 10 informative points per trace, and a random leakage

attack requires the adoption of methods that reduce the variance term. The high complexity of template attack explains also why the error rate of template attack seems independent on the leakage model: template attack can represent any non-linear relation between the leakage model and the target value.

4.3 Stochastic attack

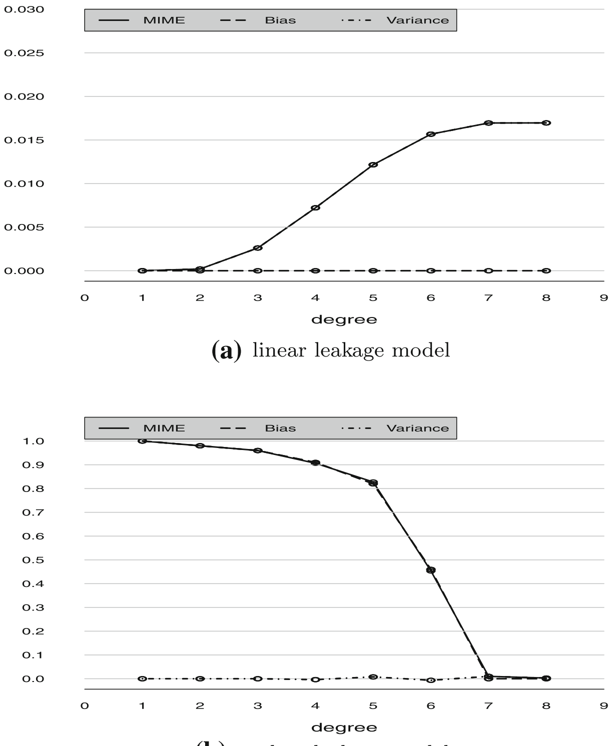
Stochastic attack differs from template attack by its ability to vary its complexity (i.e. its degree level). As a result, we focus on this aspect by varying the degree and the leakage model.

Figure 9 shows the MIME, the bias and the variance in function of the degree level of the stochastic attack with 2 × 256 traces in the learning set, three informative points per trace and a very low noise variance of 10^{-3}. In a linear leakage model context, the error rate increases with the degree level due to the variance. In a random leakage model context, the error rate decreases with the degree level due to the bias. Note that the result of template attack on this context can be seen with the stochastic attack of degree 8. As a



(a) SNR equals 1 **(b)** SNR equals $\frac{1}{2}$ **(c)** SNR equals $\frac{1}{3}$

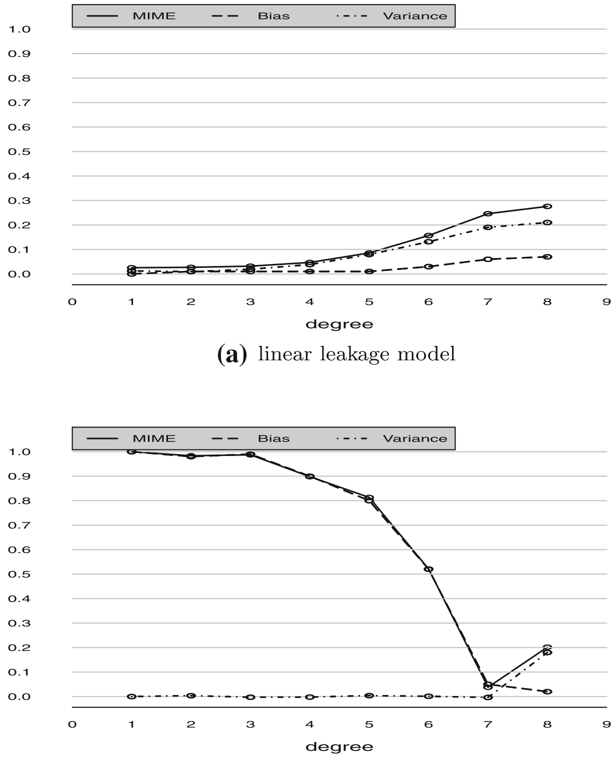
Fig. 8 MIME, bias and variance of a template attack as a function of the leakage model. There are 10×256 traces in the learning set, 10 informative points per trace and 0 non-informative points per trace



(a) linear leakage model

(b) random leakage model

Fig. 9 MIME, bias and variance of a stochastic attack as a function of its degree and the leakage model. There are 2×256 traces in the learning set, 3 informative points per trace and a noise variance of 10^{-3}



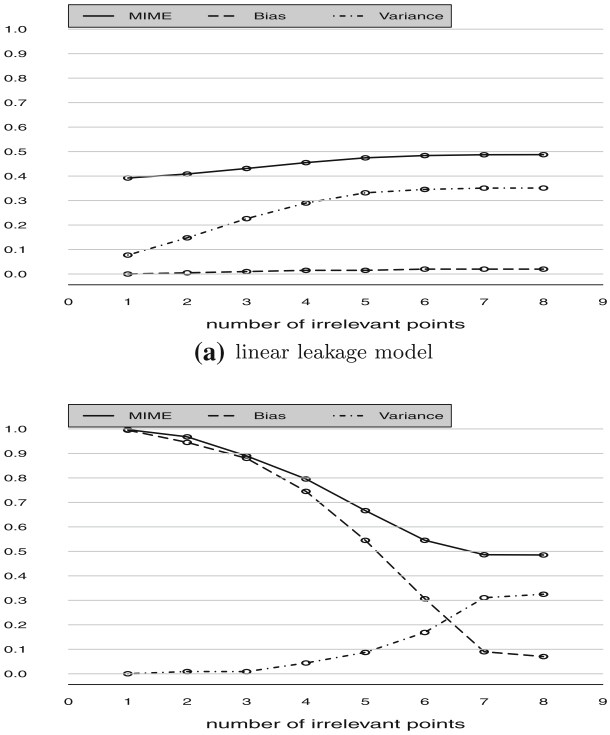
(a) linear leakage model

(b) random leakage model

Fig. 10 MIME, bias and variance of a stochastic attack as a function of its degree and the leakage model. There are 260 traces in the learning set, 3 informative points per trace and a noise variance of 10^{-3}

result, in a linear leakage model, stochastic attack with a low degree level outperforms template attack due to the fact that stochastic attack has a lower variance than template attack. In a random leakage model, template attack outperforms stochastic attack (with a degree less than 8) due to the fact that template attack has a lower bias than stochastic attack. These results are consistent with the bias–variance theory. On the one hand, a simple model (i.e. strictly less than the degree of the leakage model) has a high bias and a low variance. On the other hand, a complex model (i.e. strictly higher than the degree of the leakage model) has a low bias and a high variance. The model reaches the lowest error rate (i.e. low bias and low variance) when selecting the same degree as the leakage model.

Figure 10 shows the effect when the size of the learning set decreases to 260 traces. In a linear leakage model context, the reduction of the size of the learning set increases essentially the variance term. In a random leakage model context, stochastic attack with degree 7 minimizes the error rate while a stochastic attack of a lower degree has a higher bias and



(a) linear leakage model

(b) random leakage model

Fig. 11 MIME, bias and variance of a stochastic attack as a function of its degree and the leakage model. There are 10 × 256 traces in the learning set, 10 informative points per trace and a noise variance of 1

a stochastic attack of a higher degree has a higher variance due to a small learning set.

Figure 11 plots the result of a higher noise level of 1. This figure allows to compare stochastic attack with the previous results of template attack in a more realistic scenario where the signal-to-noise is lower. As seen previously, in a linear leakage model context, stochastic attack increases its variance term by increasing its degree leads to increase its error rate. Interestingly, in a random leakage model, stochastic attack reduces its bias term by increasing its degree leading to a slower increasing of its variance term compared to the error rate. As a result, the error rate decreases although the variance term increases. The machine learning theory stresses that a higher noise variance (e.g., a noise variance of 2 or 3) should essentially increase the variance term of stochastic attacks and, as a result, should increase the error rate (cf, the results of template attacks as well as the results of the transition from a noise variance of 10^{-3} to 1).

4.4 Discussion and guidelines for devices evaluation

The error rate evaluates the efficiency of profiled attacks. A high error rate due to wrong decisions of the evaluator (e.g., a too low stochastic degree) can lead to overestimate the security level of the cryptographic device. A principal motivation for presenting the bias–variance decomposition is to reduce the error rate based on the bias–variance diagnostic. Our results show that the error rate of a template attack depends essentially on the variance term. This is due to the fact that template attack has a high complexity (as seen in Sect. 2.2). On the opposite side, the stochastic attack with a low degree level depends essentially on the bias term while a stochastic attack with a high degree level depends on the variance term. These results are coherent with the theory: a complex model has a high variance and a low bias while a simple model has a low variance and a high bias.

The variance of template attack depends essentially on the number of points per trace: the larger the dimension of traces, the higher the number of parameters to estimate and, as a result, the higher the variance. Stochastic attack has similar result but this strategy can decrease the variance by reducing the complexity of its regression model h while maintaining the same number of points per trace. For example, stochastic attack can reduce its variance term (by reducing its degree) leading stochastic attack to outperform template attack in a linear leakage model context.

As long as the variance needs to be reduced, aggregation methods can be used. These approaches combine several models (that have high variance) leading to a reduction of the global variance. Examples of aggregation methods are bagging (discussed by Breiman in [3]) and arcing (discussed by Breiman in [4]). Random Forest [6] represents an example of aggregation methods. The main disadvantage of the aggregation approach represents the interpretability of the model. Another (more interpretable) alternative is the support vector machine [8] based on the radial basis function kernel that allows to vary its variance according to the value of a parameter γ. This parameter controls the regularization of the learning algorithm and consequently can reduce the variance (as well as the error rate) of the model. We refer to [21] for the interested readers about aggregation methods as it falls outside the scope of this paper.

Other interesting approaches in order to reduce the variance are: (1) increasing the size of the learning set, (2) reducing the number of irrelevant points, (3) increasing the number of relevant points, (4) reducing the complexity of the model, and (5) reducing the level of noise in the dataset.

As far as the bias term is concerned, we can reduce it by (1) increasing the size of the learning set, (2) reducing the number of irrelevant points, (3) increasing the number of relevant points, (4) increasing the complexity of the model and (5) applying a boosting approach. The boosting approach regroups several algorithms such as Adaboost [16]. Most boosting algorithms combines several bias models. Each model is weighted in function to its accuracy. Furthermore, the data in the learning set are also weighted in function of the complexity to learn them by the set of models. We refer to [43] for a deeper view on boosting as it is beyond the scope of this work.

An interesting observation is that the leakage model influences the bias term of stochastic attack and not the bias term of template attack. It is due to the high complexity of template attack that leads to a low bias.

The number of classes should influence the variance term for template attack. Indeed, the higher the number of classes, the higher is the number of parameters to estimate. As a result, increasing the number of classes should lead to an increase of the variance for template attack. Therefore, the size of the target value should be selected also in function of its impact on the variance term when analyzing an embedded device. Concerning stochastic attack, the impact of the number of classes is less obvious: the higher the number of classes, the higher the maximal degree of the stochastic attack. However, an adversary can vary the complexity of the function *h* (unlike template attack).

Finally, the bias–variance decomposition can be applied on the process that selects the best attack against a cryptographic device from a set of strategies. A common approach to select the best model from a set of candidates is to evaluate each strategy on a validation set and then select the attack that minimizes the error rate on the validation set. For example, we can train multiple stochastic attacks with different degrees and then select the model that minimizes the error rate on the validation set. However, when the validation set is noisy and the set of tested attacks is large, a big danger of overfitting exists. This overfitting comes from having tested too many strategies with a fixed validation set (see for example [40]). As a result, we can underestimate the security level of a cryptographic device when testing a large set of attacks. From a bias–variance decomposition point of view, we say that the process that selects the best attack increases its variance term by increasing the size of the tested attacks. In order to reduce the risk of overfitting, several approaches exist such as (1) increasing the size of the validation set, (2) imposing a restricted number of attacks when analyzing a cryptographic device and (3) using another test set. Concerning the first and the latter approaches, the problem does not disappear because the estimation of the security level depends on a (set of) specific test set(s) collected in a specific context: another test/validation set could give another estimation of the success rate. In a mathematical term, we say that the estimation of the MIME (see Eq. 19) has a bias term and a variance term. The main issue is intrinsic to the procedure: the evaluator estimates the security level of a cryptographic device and, as a result, this estimation (that has a variance term and a bias term) can differ from the true value leading to underestimate the security level (when considering the success rate of a profiled attack). Note that increasing the number of test traces allows to reduce its variance.

5 Conclusion and perspectives

The error rate represents a common criterion to evaluate profiled attacks. We presented a bias and variance decomposition of this metric. We applied the bias–variance diagnostic on several contexts using template attack and stochastic attack. We studied in depth how and why several parameters can affect each component of the error rate. Examples of parameters are the size of the learning set, the number of interested points, the number of irrelevant points, the leakage model and the noise level.

Briefly, the bias–variance decomposition allows to understand why some simple model (such as stochastic attack of degree 1) can outperform complex model (such as template attack) in a specific context while a different result can occur in a different context. As a result, there is no strategy that can be entitled to be the universally best one as formalized by the no-free-lunch theorem. Therefore, looking for the best model against a specific setting remains a challenging task in a realistic setting. At the same time, the bias–variance decomposition explains the reasons of a high or low error rate of a model and, as a result, provides insight into how to improve the attacks (e.g., by reducing the number of parameters of template attacks while keeping the ability to target non-linear leakage models).

In the future we plan to investigate the decomposition of the error rate of several other profiled and non-profiled attacks. Another interesting future research perspective concerns the error rate of a profiled attack using several traces to find the secret key. Finally, we envisage to decompose the error rate of (non-profiled and profiled) attacks targeting a masking scheme.

References

1. Bartkewitz, T., Lemke-Rust, K.: Efficient template attacks based on probabilistic multi-class support vector machines. In: Mangard, S. (ed.) CARDIS, LNCS, vol. 7771, pp. 263–276. Springer (2012)
2. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer, New York (1993)
3. Breiman, L.: Bagging predictors. Technical report, Department of Statistics, University of California (1995)
4. Breiman, L.: Arcing classifiers. Technical report, Department of Statistics, University of California (1996)
5. Breiman, L.: Randomizing outputs to increase prediction accuracy. Mach. Learn. **40**(3), 229–242 (2000)
6. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
7. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (ed.) CHES, LNCS, vol. 2523, pp. 13–28. Springer (2002)
8. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
9. Dietterich, T.G., Kong, E.B.: Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Technical report, Department of Computer Science, Oregon State University (1995)

10. Doget, J., Prouff, E., Rivain, M., Standaert, F.-X.: Univariate side channel attacks and leakage modeling. J. Cryptogr. Eng. **1**(2), 123–144 (2011)
11. Domingos, P.: A unified bias-variance decomposition and its applications. In: Langley, P. (ed.) ICML, pp. 231–238. Morgan Kaufmann, San Francisco (2000)
12. Domingos, P.: A unified bias-variance decomposition for zero-one and squared loss. In: Kautz, H.A., Porter, B.W. (eds.) AAAI/IAAI, pp. 564–569. AAAI Press/The MIT Press, New York (2000)
13. Durvaux, F., Standaert, F.-X., Veyrat-Charvillon, N.: How to certify the leakage of a chip? In: EUROCRYPT, LNCS, vol. 8441, pp. 459–475. Springer (2014) (to appear)
14. Elaabid, M.A., Guilley, S.: Portability of templates. J. Cryptogr. Eng. **2**(1), 63–74 (2012)
15. Fei, Y., Ding, A.A., Lao, J., Zhang, L.: A statistics-based fundamental model for side-channel attack analysis. Cryptology ePrint Archive, Report 2014/152 (2014). http://eprint.iacr.org/. Accessed 1 July 2014
16. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
17. Friedman, J.H.: On bias, variance, 0/1-loss, and the curse-of-dimensionality. Data Min. Knowl. Discov. **1**(1), 55–77 (1997)
18. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: concrete results. In: Koç, Ç.K., Naccache, D., Paar, C. (ed.) CHES, LNCS, vol. 2162, pp. 251–261. Springer (2001)
19. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. Neural Comput. **4**(1), 1–58 (1992)
20. Gierlichs, B., Lemke-Rust, K., Paar, C.: Templates vs. stochastic methods. In: Goubin, L., Matsui, M. (ed.) Cryptographic Hardware and Embedded Systems—CHES 2006, 8th International Workshop, Yokohama, Japan, 10–13 October 2006, Proceedings, LNCS, vol. 4249, pp. 15–29. Springer (2006)
21. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference and Prediction, 2nd edn. Springer, New York (2009)
22. Heskes, T.: Bias/variance decompositions for likelihood-based estimators. Neural Comput. **10**(6), 1425–1433 (1998)
23. Heuser, A., Rioul, O., Guilley, S.: A theoretical study of kolmogorov-smirnov distinguishers—side-channel analysis vs. differential cryptanalysis. In: Prouff, E. (ed.) Constructive Side-Channel Analysis and Secure Design—5th International Workshop, COSADE 2014, Paris, France, 13–15 April 2014. Revised Selected Papers, LNCS, vol. 8622, pp. 9–28. Springer (2014)
24. Heuser, A., Zohner, M.: Intelligent machine homicide - breaking cryptographic devices using support vector machines. In: Proceedings of the Third international conference on Constructive Side-Channel Analysis and Secure Design, LNCS, vol. 7275, pp. 249–264. Springer, Berlin, Heidelberg (2012)
25. Hospodar, G., Gierlichs, B., Mulder, E.D., Verbauwhede, I., Vandewalle, J.: Machine learning in side-channel analysis: a first study. J. Cryptogr. Eng. **1**(4), 293–302 (2011)
26. Hospodar, G., Mulder, E.D., Gierlichs, B., Vandewalle, J., Verbauwhede, I.: Least squares support vector machines for side-channel analysis. In: Second International Workshop on Constructive Side Channel Analysis and Secure Design, pp. 99–104. Center for Advanced Security Research Darmstadt (2011)
27. James, G., Hastie, T.: Generalizations of the bias/variance decomposition for prediction error. Technical report, Department of Statistics, Standford University (1996)
28. Kocher, P.C.: Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO, LNCS, vol. 1109, pp. 104–113. Springer (1996)
29. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: CRYPTO, LNCS, pp. 388–397. Springer (1999)
30. Kohavi, R., Wolpert, D.: Bias plus variance decomposition for zero-one loss functions. In: Saitta, L. (ed.) ICML, pp. 275–283. Morgan Kaufmann, San Francisco (1996)
31. Lerman, L., Bontempi, G., Markowitch, O.: Side channel attack: an approach based on machine learning. In: Second International Workshop on Constructive Side Channel Analysis and Secure Design, pp. 29–41. Center for Advanced Security Research Darmstadt (2011)
32. Lerman, L., Bontempi, G., Markowitch, O.: Power analysis attack: an approach based on machine learning. Int. J. Appl. Cryptogr. **3**(2), 97–115 (2014)
33. Lerman, L., Bontempi, G., Markowitch, O.: A machine learning approach against a masked aes. J. Cryptogr. Eng. **5**(2), 123–139 (2015)
34. Lerman, L., Bontempi, G., Ben Taieb, S., Markowitch, O.: A time series approach for profiling attack. In: Gierlichs, B., Guilley, S., Mukhopadhyay, D. (ed.) SPACE, LNCS, vol. 8204, pp. 75–94. Springer (2013)
35. Lerman, L., Fernandes Medeiros, S., Bontempi, G., Markowitch, O.: A machine learning approach against a masked AES. In: Francillon, A., Rohatgi, P. (ed.) Smart Card Research and Advanced Applications—12th International Conference, CARDIS 2013, Berlin, Germany, 27–29 November 2013. Revised Selected Papers, LNCS, vol. 8419, pp. 61–75. Springer (2013)
36. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks—Revealing the Secrets of Smart Cards. Springer, New York (2007)
37. Matsui, M.: Linear cryptoanalysis method for des cipher. In: Helleseth, T. (ed.) EUROCRYPT, LNCS, vol. 765, pp. 386–397. Springer (1993)
38. Meier, W., Staffelbach, O.: Nonlinearity criteria for cryptographic functions. In: Quisquater, J-J., Vandewalle, J. (ed.) EUROCRYPT, LNCS, vol. 434, pp. 549–562. Springer (1989)
39. Montminy, D.P., Baldwin, R.O., Temple, M.A., Laspe, E.D.: Improving cross-device attacks using zero-mean unit-variance normalization. J. Cryptogr. Eng. **3**(2), 99–110 (2013)
40. Ng, A.Y.: Preventing “overfitting” of cross-validation data. In: Fisher, D.H. (ed.) ICML, pp. 245–253. Morgan Kaufmann, San Francisco (1997)
41. Prouff, E.: DPA attacks and S-boxes. In: Gilbert, H., Handschuh, H. (ed.) Fast Software Encryption, LNCS, vol. 3557. pp. 424–441. Springer, Berlin, Heidelberg (2005)
42. Rivain, M., Dottax, E., Prouff, E.: Block ciphers implementations provably secure against second order side channel analysis. In: Nyberg, K. (ed.) FSE, LNCS, vol. 5086, pp. 127–143. Springer (2008)
43. Schapire, R. E.: The boosting approach to machine learning: an overview. In: MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA, USA (2001)
44. Schindler, W., Lemke, K., Paar, C.: A stochastic model for differential side channel cryptanalysis. In: Rao, J.R., Sunar, B. (ed.) CHES, LNCS, vol. 3659, pp. 30–46. Springer (2005)
45. Standaert, F-X., Malkin, T., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Joux, A. (ed.) EUROCRYPT, LNCS, vol. 5479, pp. 443–461. Springer (2009)
46. Sugawara, T., Homma, N., Aoki, T., Satoh, A.: Profiling attack using multivariate regression analysis. IEICE Electron. Express **7**(15), 1139–1144 (2010)
47. Tibshirani, R.: Bias, variance, and prediction error for classification rules. Technical report, Statistics Department, University of Toronto, Toronto (1996)
48. Weisberg, S.: Applied Linear Regression. Wiley Series in Probability and Statistics, Wiley, New York (2005)
49. Whitnall, C., Oswald, E.: Profiling DPA: efficacy and efficiency trade-offs. In: Bertoni, G., Coron, J-S. (ed.) CHES, LNCS, vol. 8086, pp. 37–54. Springer (2013)