Template Attacks vs. Machine Learning Revisited
(and the Curse of Dimensionality in Side-Channel Analysis)

Liran Lerman[1], Romain Poussier[2], Gianluca Bontempi[1],
Olivier Markowitch[1] and François-Xavier Standaert[2]

[1] Département d'informatique, Université Libre de Bruxelles.
[2] ICTEAM/INGI, Université catholique de Louvain, Belgium.

Abstract. Template attacks and machine learning are two popular approaches to profiled side-channel analysis. In this paper, we aim to contribute to the understanding of their respective strengths and weaknesses, with a particular focus on their curse of dimensionality. For this purpose, we take advantage of a well-controlled simulated experimental setting in order to put forward two important intuitions. First and from a theoretical point of view, the data complexity of template attacks is not sensitive to the dimension increase in side-channel traces given that their profiling is perfect. Second and from a practical point of view, concrete attacks are always affected by (estimation and assumption) errors during profiling. As these errors increase, machine learning gains interest compared to template attacks, especially when based on random forests.

1 Introduction

In a side-channel attack, an adversary targets a cryptographic device that emits a measurable leakage depending on the manipulated data and/or the executed operations. Typical examples of physical leakages include the power consumption [15], the processing time [14] and the electromagnetic emanation [9].

Evaluating the security level of cryptographic implementations is an important concern, e.g. for modern smart cards. In this respect, profiled attacks are useful tools, since they can be used to approach their worst-case security level [24]. Such attacks essentially work in two steps: first a leakage model is estimated during a so-called profiling phase, then the leakage model is exploited to extract key-dependent information in an online phase. Many different approaches to profiling have been introduced in the literature. Template Attacks (TA), e.g. based on a Gaussian assumption [4], are a typical example. The stochastic approach exploiting Linear Regression (LR) is a frequently considered alternative [22]. More recently, solutions relying on Machine Learning (ML) have also been investigated [2, 11, 13, 12, 16, 17, 19]. These previous works support the claim that ML-based attacks are effective and lead to successful key recoveries. This is natural since they essentially exploit the same discriminating criteria as TA and LR (i.e. a difference in the mean traces corresponding to different intermediate computations if an unprotected implementation is targeted

– a difference in higher-order statistical moments if the device is protected with masking). By contrast, it remains unclear whether ML can lead to more efficient attacks, either in terms of profiling or in terms of online key recovery. Previous publications conclude in one or the other direction, depending on the implementation scenario considered, which is inherent to such experimental studies.

In this paper, we aim to complement these previous works with a more systematic investigation of the conditions under which ML-based attacks may outperform TA (or not)[1]. For this purpose, we start with the general intuition that ML-based approaches are generally useful in order to deal with high-dimensional data spaces. Following, our contributions are twofold. First, we tackle the (theoretical) question whether the addition of useless (i.e. non-informative) leakage samples in leakage traces has an impact on their informativeness if a perfect profiling phase is achieved. We show that the (mutual) information leakage estimated with a TA exploiting such a perfect model is independent of the number of useless dimensions if the useless leakage samples are independent of the useful ones. This implies that ML-based attacks cannot be more efficient than template attacks in the online phase if the profiling is sufficient. Second, we study the practical counterpart of this question, and analyze the impact of imperfect profiling on our conclusions. For this purpose, we rely on a simulated experimental setting, where the number of (informative and useless) dimensions is used as a parameter. Using this setting, we evaluate the curse of dimensionality for concrete TA and compare it with ML-based attacks exploiting Support Vector Machines (SVM) and Random Forests (RF). That is, we considered SVM as a popular tool in the field of side-channel analysis, and RF as an interesting alternative (since its random feature selection makes its behavior quite different than TA and SVM). Our experiments essentially conclude that TA outperform ML-based attacks whenever the number of dimensions can be kept reasonably low, e.g. thanks to a selection of Points of Interests (POI), and that ML (and RF in particular) become(s) interesting in "extreme" profiling conditions (i.e. with large traces and a small profiling sets) – which possibly arise when little information about the target device is available to the adversary.

As a side remark, we also observe that most current ML-based attacks rate key candidates according to (heuristic) scores rather than probabilities. This prevents the computation of probability-based metrics (such as the mutual/perceived information [20]). It may also have an impact on the efficiency of key enumeration [25], which is an interesting scope for further investigation.

The rest of the paper is organized as follows. Section 2 contains notations, the attacks considered, our experimental setting and evaluation metrics. Section 3 presents our theoretical result on the impact of non-informative leakage samples in perfect profiling conditions. Section 4 discusses practical (simulated) experiments in imperfect profiling conditions, in different contexts. Eventually, Section 5 concludes the paper and discusses perspectives of future work.

[1] Note that the gain of LR-based attacks over TA is known and has been analyzed, e.g. in [10, 23]. Namely, it essentially depends on the size of the basis used in LR.

2 Background

2.1 Notations

We use capital letters for random variables and small caps for their realizations. We use sans serif font for functions (e.g. F) and calligraphic fonts for sets (e.g. $\mathcal{A}$). We denote the conditional probabilitiy of a random variable A given B with $\Pr[A|B]$ and use the acronym SNR for the signal-to-noise ratio.

2.2 Template Attacks

Let $l_{x,k}$ be a leakage trace measured on a cryptographic device that manipulates a target intermediate value $v = \mathsf{f}(x, k)$ associated to a known plaintext (byte) x and a secret key (byte) k. In a TA, the adversary first uses a set of profiling traces $\mathcal{L}_p$ in order to estimate a leakage model, next denoted as $\hat{\Pr}_{\text{model}}\left[l_{x,k} \mid \hat{\theta}_{x,k}\right]$, where $\hat{\theta}_{x,k}$ represents the (estimated) parameters of the leakage Probability Density Function (PDF). The set of profiling traces is typically obtained by measuring a device that is similar to the target, yet under control of the adversary. Next, during the online phase, the adversary uses a set of new attack traces $\mathcal{L}_a$ (obtained by measuring the target device) and selects the secret key (byte) $\tilde{k}$ maximizing the product of posterior probabilities:

$$\tilde{k} = \underset{k^*}{\operatorname{argmax}} \prod_{l_{x,k} \in \mathcal{L}_a} \frac{\hat{\Pr}_{\text{model}}\left[l_{x,k} \mid \hat{\theta}_{x,k^*}\right] \cdot \Pr[k^*]}{\hat{\Pr}_{\text{model}}[l_{x,k}]}. \tag{1}$$

Concretely, the seminal TA paper suggested to use Gaussian estimations for the leakage PDF [4]. We will follow a similar approach and consider a Gaussian (simulated) experimental setting. It implies that the parameters $\hat{\theta}_{x,k}$ correspond to mean vectors $\hat{\mu}_{x,k}$ and covariance matrices $\hat{\Sigma}_{x,k}$. However, we note that any other PDF estimation could be considered by the adversary/evaluator [8]. We will further consider two types of TA: in the Naive Template Attack (NTA), we will indeed estimate one covariance matrix per intermediate value; in the Efficient Template Attack (ETA), we will pool the covariance estimates (assumed to be equal) across all intermediate values, as previously suggested in [5].

In the following, we will keep the $l_{x,k}$ and v notations for leakage traces and intermediate values, and sometimes omit the subscripts for simplicity.

2.3 Support Vector Machines

In their basic (two-classes) context, SVM essentially aims at estimating Boolean functions [6]. For this purpose, it first performs a supervised learning with labels (e.g. $v = -1$ or $v = 1$), annotating each sample of the profiling set. The binary SVM estimates a hyperplane $y = \hat{w}^\top l + \hat{b}$ that separates the two classes with

the largest possible margin, in the geometrical space of the vectors. Then in the attack phase, any new trace l will be assigned a label $\tilde{v}$ as follows:

$$\tilde{v} = \begin{cases} 1 & (\hat{w}^\top l + \hat{b}) \geq 1, \\ -1 & \text{otherwise.} \end{cases} \tag{2}$$

Mathematically, SVM finds the parameters $\hat{w} \in \mathbb{R}^{n_s}$ (where n_s is the number of time samples per trace) and $\hat{b} \in \mathbb{R}$ by solving the convex optimization problem:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}(w^\top w), \\ \text{subject to} \quad & v(w^\top \phi(l_v) + b) \geq 1, \end{aligned} \tag{3}$$

where ϕ denotes a projection function that maps the data into a higher (sometimes infinite) dimensional space usually denoted as the feature space. Our experiments considered a Radial Basis kernel Function ϕ (RBF), which is a commonly encountered solution, both in the machine learning field and the side-channel communities. The RBF kernel maps the traces into an infinite dimensional Hilbert space in order to find a hyperplane that efficiently discriminate the traces. It is defined by a parameter γ that essentially relates to the "variance" of the model. Roughly, the variance of a model is a measure on the variance of its output in function of the variance of the profiling set. The higher the value of γ, the lower the variance of the model is. Intuitively, the variance of a model therefore relates to its complexity (e.g. the higher the number of points per trace, the higher the variance of the model). We always selected the value of γ as one over the number of points per trace, which is a natural choice to compensate the increase of the model variance due to the increase of the number of points per trace. Future works could focus on other strategies to select this parameter, although we do not expect them to have a strong impact on our conclusions.

When the problem of Equation 3 is feasible with respect to the constraints, the data is said to be linearly separable in the feature space. As the problem is convex, there is a guarantee to find a unique global minimum. SVM can be generalized to multi-class problems (which will be useful in our context with typically 256 target intermediate values) and produce scores for intermediate values based on the distance to the hyperplane. In our experiments, we considered the "*one-against-one*" approach. In a one-against-one strategy, the adversary builds one SVM for each possible pair of target values. During the attack phase, the adversary selects the target value with a majority vote among the set of SVMs. Because of place constraints, we refer to [7] for a complete explanation.

2.4 Random Forests

Decision trees are classification models that use a set of binary rules to calculate a target value. They are structured as diagrams made of nodes and directed edges, where nodes can be of three types: root (i.e. the top node in the tree), internal (represented by a circle in Figure 1) and leaf (represented by a square

in Figure 1). In our side-channel context, we typically consider decision trees in which (1) the value associated to a leaf is a class label corresponding to the target to be recovered, (2) each edge is associated to a test on the value of a time sample in the leakage traces, and (3) each internal node has one incoming edge from a node called the parent node, as also represented in Figure 1.

In the profiling phase, learning data is used to build the model. For this purpose, the learning set is first associated to the root. Then, this set is split based on a time sample that most effectively discriminates the sets of traces associated to different target intermediate values. Each subset newly created is associated with a child node. The tree generator repeats this process on each derived subset in a recursive manner, until the child node contains traces associated to the same target value or the gain to split the subset is less than some threshold. That is, it essentially determines at which time sample to split, the value of the split, and the decision to stop or to split again. It then assigns terminal nodes to a class (i.e. intermediate value). Next, in the attack phase, the model simply predicts the target intermediate value by applying the classification rules to the new traces to classify. We refer to [21] for more details on decision trees.

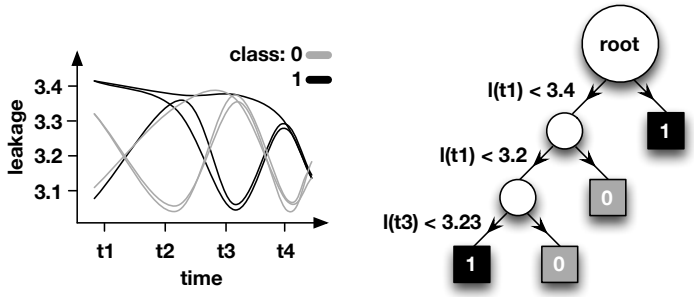


Fig. 1: Decision tree with two classes ($l(t1)$ is the leakage at time $t1$).

The Random Forests (RF) introduced by Breiman can be seen as a collection of classifiers using many (unbiased) decision trees as models [3]. It relies on model averaging (aka bagging) that leads to have a low variance of the resulting model. After the profiling phase, RF returns the most consensual prediction for a target value through a majority vote among the set of trees. RF are based on three main principles. First, each tree is constructed with a different learning set by re-sampling (with replacement) the original dataset. Secondly, the nodes of the trees are split using the best time sample among a subset of randomly chosen ones (by contrast to conventional trees where all the time samples are used). The size of this subset was set to the square of the number of time samples (i.e. $\sqrt{n_s}$) as suggested by Breiman. These features allow obtaining decorrelated trees, which improves the accuracy of the resulting RF model. Finally, and unlike conventional decision trees as well, the trees of a RF are fully grown and are not

pruned, which possibly leads to overfitting (i.e. each tree has a low bias but a high variance) that is reduced by averaging the trees. The main (meta-) parameters of a RF are the number of trees. Intuitively, increasing the number of trees reduces the instability (aka variance) of the models. We set this number to 500 by default, which was sufficient in our experiments in order to show the strength of this model compared to template attack. We leave the detailed investigation of these parameters as an interesting scope for further research.

2.5 Experimental setting

Let $l_{p,k}(t)$ be the t-th time sample of the leakage trace $l_{p,k}$. We consider contexts where each trace $l_{p,k}$ represents a vector of n_s samples, that is:

$$l_{p,k} = \{l_{p,k}(t) \in \mathbb{R} \mid t \in [1; n_s]\}. \tag{4}$$

Each sample represents the output of a leakage function. The adversary has access to a profiling set of N_p traces per target intermediate value, in which each trace has d informative samples and u uninformative samples (with $d + u = n_s$). The informative samples are defined as the sum of a deterministic part representing the useful signal (denoted as δ) and a random Gaussian part representing the noise (denoted as ϵ), that is:

$$l_{p,k}(t) = \delta_t(p,k) + \epsilon_t, \tag{5}$$

where the noise is independent and identically distributed for all t's. In our experiments, the deterministic part δ corresponds to the output of the AES S-box, iterated for each time sample and sent through a function G, that is:

$$\delta_t(p,k) = \mathsf{G}\left(\mathsf{SBox}^t(p \oplus k)\right), \tag{6}$$

where:

$$\mathsf{SBox}^1(p \oplus k) = \mathsf{SBox}(p \oplus k),$$
$$\mathsf{SBox}^t(p \oplus k) = \mathsf{SBox}\left(\mathsf{SBox}^{t-1}(p \oplus k)\right).$$

Concretely, we considered a function G that is a weighted sum of the S-box output bits. However, all our results can be generalized to other functions (preliminary experiments did not exhibit any deviation with highly non-linear leakage functions – which is expected in a first-order setting where the leakage informativeness essentially depends on the SNR [18]). We set our signal variance to 1 and used Gaussian distributed noise variables ϵ_t with mean 0 and variance σ^2 (i.e. the SNR was set to $\frac{1}{\sigma^2}$). Eventually, uninformative samples were simply generated with only a noisy part. This simulated setting is represented in Figure 2 and its main parameters can be summarized as follows:

- Number of informative points per trace (denoted as d),
- Number of uninformative points per trace (denoted as u),

– Number of profiling traces per intermediate value (denoted as N_p),
– Number of traces in the attack step (noted N_a),
– Noise variance (denoted as σ^2) and SNR.

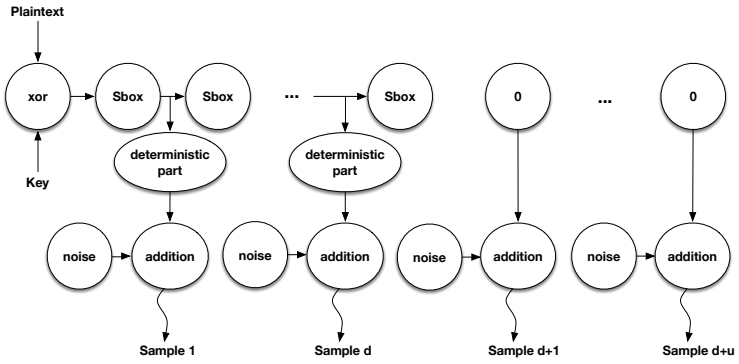


Fig. 2: Simulated leaking implementations.

2.6 Evaluation metrics

The efficiency of side-channel attacks can be quantified according to various metrics. We will use information theoretic and security metrics advocated in [24].

Success rate (SR). For an attack targeting a subkey (e.g. a key byte) and allowing to sort the different candidates, we define the success rate of order o as the probability that the correct subkey is ranked among the first o candidates. The success rate is generally computed in function of the number of attack traces N_a (given a model that has been profiled using N_p traces). In the rest of this paper, we focus on the success rate of order 1 (i.e. the correct key rated first).

Perceived/Mutual information (PI/MI). Let X, K, L be random variables representing a target key byte, a known plaintext and a leakage trace. The perceived information between the key and the leakage is defined as [20]:

$$\hat{\mathrm{PI}}(K; X, L) = \mathrm{H}(K) + \sum_{k \in \mathcal{K}} \Pr[k] \sum_{x \in \mathcal{X}} \Pr[x] \sum_{l \in \mathcal{L}} \Pr_{\mathrm{chip}}[l|x,k] \cdot \log_2 \hat{\Pr}_{\mathrm{model}}[k|x,l].$$

The PI measures the adversary's ability to interpret measurements coming from the true (unknown) chip distribution $\Pr_{\mathrm{chip}}[l|x,k]$ with an estimated model $\hat{\Pr}_{\mathrm{model}}[l|x,k]$. $\Pr_{\mathrm{chip}}[l|x,k]$ is generally obtained by sampling the chip distribution (i.e. making measurement). Of particular interest for the next section will be the context of *perfect profiling*, where we assume that the adversary's model and the chip distribution are identical (which, strictly speaking, can only happen

in simulated experimental settings since any profiling based on real traces will at least be imperfect because of small estimation errors [8]). In this context, the estimated PI will exactly correspond to the (worst-case) estimated MI.

Information theoretic metrics such as the MI/PI are especially interesting for the comparison of profiled side-channel attacks as we envision here. This is because they can generally be estimated based on a single plaintext (i.e. with $N_a = 1$) whereas the success rate is generally estimated for varying N_a's. In other words, their scalar value provides a very similar intuition as the SR curves [23]. Unfortunately, the estimation of information theoretic metrics requires distinguishers providing probabilities, which is not the case of ML-based attacks[2]. As a result, our concrete experiments comparing TA, SVM and RF will be based on estimations of the success rate for a number of representative parameters.

3 Perfect profiling

In this section, we study the impact of useless samples in leakage traces on the performances of TA with perfect profiling (i.e. the evaluator perfectly knows the leakages' PDF). In this context, we will use Pr for both $\Pr_{\text{model}}$ and $\Pr_{\text{chip}}$ (since they are equal) and omit subscripts for the leakages l to lighten notations.

Proposition 1. *Let us assume two TA with perfect models using two different attack traces l_1 and l_2 associated to the same plaintext x: l_1 is composed of d samples providing information and $l_2 = [l_1||\epsilon]$ (where $\epsilon = [\epsilon_1, ..., \epsilon_u]$ represents noise variables independent of l_1 and the key.). Then the mutual information leakage $\mathrm{MI}(K; X, L)$ estimated with their (perfect) leakage models is the same.*

Proof. As clear from the definitions in Section 2.6, the mutual/perceived information estimated thanks to TA only depend on $\Pr[k|l]$. So we need to show that these conditional probabilities $\Pr[k|l_2]$ and $\Pr[k|l_1]$ are equal. Let k and k' represent two key guesses. Since ϵ is independent of l_1 and k, we have:

$$\begin{aligned}\frac{\Pr[l_2|k']}{\Pr[l_2|k]} &= \frac{\Pr[l_1|k'] \cdot \Pr[\epsilon|k']}{\Pr[l_1|k] \cdot \Pr[\epsilon|k]}, \\ &= \frac{\Pr[l_1|k'] \cdot \Pr[\epsilon]}{\Pr[l_1|k] \cdot \Pr[\epsilon]}, \\ &= \frac{\Pr[l_1|k']}{\Pr[l_1|k]}. \end{aligned} \tag{7}$$

[2] There are indeed variants of SVM and RF that aim to remedy to this issue. Yet, the "probability-like" scores they output are not directly exploitable in the estimation of information theoretic metrics either. For example, we could exhibit examples where probability-like scores of one do not correspond to a success rate of one.

This directly leads to:

$$\frac{\sum_{k' \in \mathcal{K}} \Pr[l_2|k']}{\Pr[l_2|k]} = \frac{\sum_{k' \in \mathcal{K}} \Pr[l_1|k']}{\Pr[l_1|k]},$$
$$\frac{\Pr[l_2|k]}{\sum_{k' \in \mathcal{K}} \Pr[l_2|k']} = \frac{\Pr[l_1|k]}{\sum_{k' \in \mathcal{K}} \Pr[l_1|k']},$$
$$\Pr[k|l_2] = \Pr[k|l_1], \tag{8}$$

which concludes the proof.

Quite naturally, this proof does not hold as soon as there are dependencies between the d first samples in l_1 and the u latter ones. This would typically happen in contexts where the noise at different time samples is correlated (which could then be exploited to improve the attack). Intuitively, this simple result suggests that in case of perfect profiling, the detection of POI is not necessary for a TA, since useless points will not have any impact on the attack's success. Since TA are optimal from an information theoretic point-of-view, it also means that the ML-based approaches cannot be more efficient in this context.

Note that the main reason why we need a perfect model for the result to hold is that we need the independence between the informative and non-informative samples to be reflected in these models as well. For example, in the case of Gaussian templates, we need the covariance terms that corresponds to the correlation between informative and non-informative samples to be null (which will not happen for imperfectly estimated templates). In fact, the result would also hold for imperfect models, as long as these imperfections do not suggest significant correlation between these informative and non-informative samples. But of course, we could not state that TA necessarily perform better than ML-based attacks in this case. Overall, this conclusion naturally suggests a more pragmatic question. Namely, perfect profiling never occurs in practice. So how does this theoretical intuition regarding the curse of dimensionality for TA extend to concrete profiled attack (with bounded profiling phases)? We study it in the next section.

4 Experiments with imperfect profiling

We now consider examples of TA, SVM- and RF-based attacks in order to gain intuition about their behavior in concrete profiling conditions. As detailed in Section 2, we will use a simulated experimental setting with various number of informative and uninformative samples in the leakage traces for this purpose.

4.1 Nearly perfect profiling

As a first experiment, we considered the case where the profiling is "sufficient" – which should essentially confirm the result of Proposition 1. For this purpose, we analyzed simulated leakage traces with 2 informative points (i.e. $d = 2$), $u = 0$ and $u = 15$ useless samples, and a SNR of 1, in function of the number of

traces per intermediate value in the profiling set N_p. As illustrated in Figure 3, we indeed observe that (e.g.) the PI is independent of u if the number of traces in the profiling set is "sufficient" (i.e. all attacks converge towards the same PI in this case). By contrast, we notice that this "sufficient" number depends on u (i.e. the more useless samples, the larger N_p needs to be). Besides, we also observe that the impact of increasing u is stronger for NTA than ETA, since the first one has to deal with a more complex estimation. Indeed, the ETA has 256 times more traces than the NTA to estimate the covariance matrice. So overall, and as expected, as long a the profiling set is large enough and the assumptions used to build the model capture the leakage samples sufficiently accurately, TA are indeed optimal, independent of the number of samples they actually profile. So there is little gain to expect from ML-based approaches in this context.

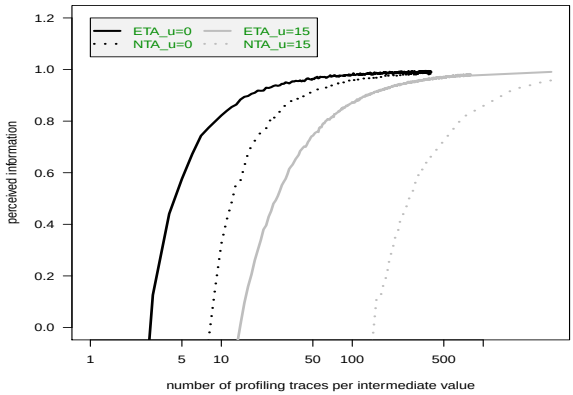


Fig. 3: Perceived information for NTA and ETA in function of N_p with SNR=1.

4.2 Imperfect profiling

We now move to the more concrete case were profiling is imperfect. In our simulated setting, imperfections naturally arise from limited profiling (i.e. estimation errors): we will investigate their impact next and believe they are sufficient to put forward some useful intuitions regarding the curse of dimensionality in (profiled) side-channel attacks. Yet, we note that in general, assumption errors can also lead to imperfect models, that are more difficult to deal with (see, e.g. [8]) and are certainly worth further investigations. Besides, and as already mentioned, since we now want to compare TA, SVM and RF, we need to evaluate and compare them with security metrics (since the two latter ones do not output the probabilities required to estimate information theoretic metrics).

In our first experiment, we set again the number of useful dimensions to $d = 2$ and evaluated the success rate of the different attacks in function of the number of non-informative samples in the leakages traces (i.e. u), for different sizes of the

profiling set. As illustrated in Figure 4, we indeed observe that for a sufficient profiling, ETA is the most efficient solution. Yet, it is also worth observing that NTA provides the worst results overall, which already suggests that comparisons are quite sensitive to the adversary/evaluator's assumptions. Quite surprisingly, our experimental results show that up to a certain level, the success rate of RF increases with the number of points without information. The reason is intrinsic to the RF algorithm in which the trees need to be as decorrelated as possible. As a result, increasing the number of points in the leakage traces leads to a better independence between trees and improves the success rate. Besides, the most interesting observation relates to RF in high dimensionality, which remarkably resists the addition of useless samples (compared to SVM and TA). The main reason for this behavior is the random feature selection embedded into this tool. That is, for a sufficient number of trees, RF eventually detects the informative POI in the traces, which makes it less sensitive to the increase of u. By contrast, TA and SVM face a more and more difficult estimation problem in this case.

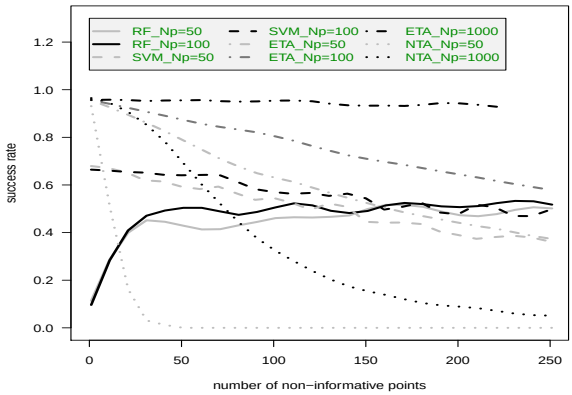
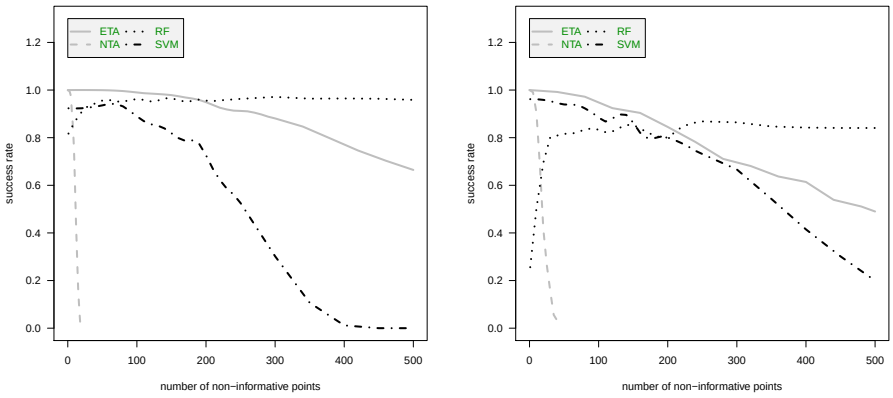


Fig. 4: Success rate for NTA, ETA, SVM and RF in fct. of the number of useless samples u, for various sizes of the profiling set N_p, with $d = 2$, SNR=1, $N_a = 15$.

Another noticeable element of Figure 4 is that SVM and RF seem to be bounded to lower success rates than TA. But this is mainly an artifact of using the success rate as evaluation metric. As illustrated in Figure 5 increasing either the number of informative dimensions in the traces d or the number of attack traces N_a leads to improved success rates for the ML-based approaches as well. For the rest, the latter figure does not bring significantly new elements. We essentially notice that RF becomes interesting over ETA for very large number of useless dimensions and that ETA is most efficient otherwise.

Eventually, the interest of the random feature selection in RF-based models raises the question of the time complexity for these different attacks. That is, such a random feature selection essentially works because there is a large



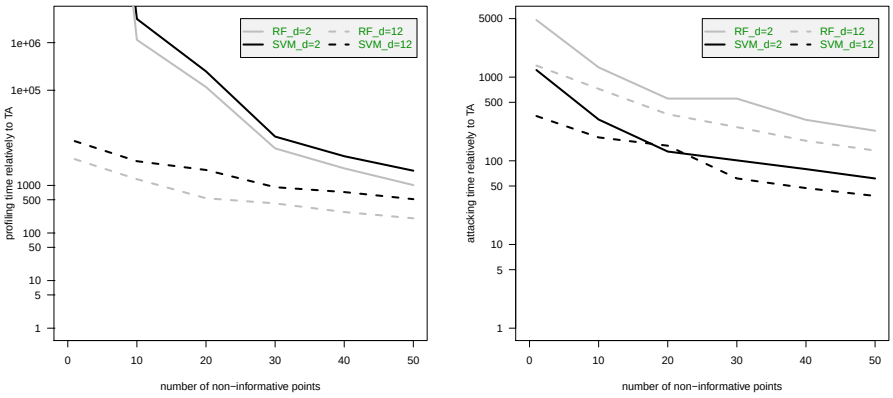
(a) (b)

Fig. 5: (a) Success rate for NTA, ETA, SVM and RF in function of the number of useless samples u, with parameters $N_p = 25$, $d = 5$, SNR=1 and $N_a = 15$. (b) Similar experiment with parameters $N_p = 50$ $d = 2$, SNR=1 and $N_a = 30$.

enough number of trees in our RF models. But increasing this number naturally increases the time complexity of the attacks. For this purpose, we report some results regarding the time complexity of our attacks in Figure 6. As a preliminary note, we mention that those results are based on prototype implementations in different programming languages (C for TA, R for SVM and RF). So they should only be taken as a rough indication. Essentially, we observe an overhead for the time complexity of ML-based attacks, which vanishes as the size of the leakage traces increases. Yet, and most importantly, this overhead remains comparable for SVM and RF in our experiments (mainly due to the fact that the number of trees was set to a constant 500). So despite the computational cost of these attacks is not negligible, it remains tractable for the experimental parameters we considered (and could certainly be optimized in future works).

5 Conclusion

Our results provide interesting insights on the curse of dimensionality for side-channel attacks. From a theoretical point of view, we first showed that as long as a limited number of POI can be identified in leakage traces and contain most of the information, TA are the method of choice. Such a conclusion extends to any scenario where the profiling can be considered as "nearly perfect". By contrast, we also observed that as the number of useless samples in leakage traces increases and/or the size of the profiling set becomes too limited, ML-based attacks gain interest. In our simulated setting, the most interesting gain is exhibited for RF-based models, thanks to their random feature selection. Interestingly, the recent work of Banciu *et al.* reached a similar conclusion in a different context, namely, Simple Power Analysis and Algebraic Side-Channel Analysis [1].



(a) (b)

Fig. 6: Time complexity for ETA, SVM and RF in fct. of the number of useless samples, for $d = [2, 12]$ and $N_p = 25$. (a) Profiling phase. (b) Attack phase.

Besides, and admittedly, the simulated setting we investigated is probably most favorable to TA, since only estimation errors can decrease the accuracy of the adversary/evaluator models in this case. One can reasonably expect that real devices with harder to model noise distributions would improve the interest of SVM compared to ETA – as has been suggested in previously published works. As a result, the extension of our experiments towards other distributions is an interesting avenue for further research. In particular, the study of leakage traces with correlated noise could be worth additional investigations in this respect. Meanwhile, we conclude with the interesting intuition that TA are most efficient for well understood devices, with sufficient profiling, as they can approach the worst-case security level of an implementation in such context. By contrast, ML-based attacks (especially RF) are promising alternative(s) in black box settings, with only limited understanding of the target implementation.

Acknowledgements. F.-X. Standaert is a research associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in parts by the European Commission through the ERC project 280141 (CRASH).

References

1. Valentina Banciu, Elisabeth Oswald, and Carolyn Whitnall. Reliable information extraction for single trace attacks. *IACR Cryptology ePrint Archive*, 2015:45, 2015.
2. Timo Bartkewitz and Kerstin Lemke-Rust. Efficient template attacks based on probabilistic multi-class support vector machines. In Stefan Mangard, editor, *CARDIS*, volume 7771 of *Lecture Notes in Computer Science*, pages 263–276. Springer, 2012.
3. Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

4. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.
5. Omar Choudary and Markus G. Kuhn. Efficient template attacks. In Aurélien Francillon and Pankaj Rohatgi, editors, *CARDIS*, volume 8419 of *Lecture Notes in Computer Science*, pages 253–270. Springer, 2013.
6. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
7. Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2010.
8. 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, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 459–476. Springer, 2014.
9. Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.
10. Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. stochastic methods. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2006.
11. Annelie Heuser and Michael Zohner. Intelligent machine homicide - breaking cryptographic devices using support vector machines. In Werner Schindler and Sorin A. Huss, editors, *COSADE*, volume 7275 of *Lecture Notes in Computer Science*, pages 249–264. Springer, 2012.
12. Gabriel Hospodar, Benedikt Gierlichs, Elke De Mulder, Ingrid Verbauwhede, and Joos Vandewalle. Machine learning in side-channel analysis: a first study. *J. Cryptographic Engineering*, 1(4):293–302, 2011.
13. Gabriel Hospodar, Elke De Mulder, Benedikt Gierlichs, Joos Vandewalle, and Ingrid Verbauwhede. Least Squares Support Vector Machines for Side-Channel Analysis. In *Second International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 99–104. Center for Advanced Security Research Darmstadt, 2011.
14. Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In Neal Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.
15. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.
16. Liran Lerman, Gianluca Bontempi, and Olivier Markowitch. Side-Channel Attacks: an Approach Based on Machine Learning. In *Second International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 29–41. Center for Advanced Security Research Darmstadt, 2011.
17. Liran Lerman, Gianluca Bontempi, and Olivier Markowitch. Power analysis attack: an approach based on machine learning. *IJACT*, 3(2):97–115, 2014.
18. 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.
19. Hiren Patel and Rusty O. Baldwin. Random forest profiling attack on advanced encryption standard. *IJACT*, 3(2):181–194, 2014.

20. Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In Kenneth G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 109–128. Springer, 2011.
21. Lior Rokach and Oded Maimon. *Data Mining with Decision Trees: Theory and Applications*. Series in machine perception and artificial intelligence. World Scientific Publishing Company, Incorporated, 2008.
22. Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.
23. François-Xavier Standaert, François Koeune, and Werner Schindler. How to compare profiled side-channel attacks? In Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud, editors, *ACNS*, volume 5536 of *Lecture Notes in Computer Science*, pages 485–498, 2009.
24. François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.
25. Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2012.