Profiling Side-channel Analysis in the Restricted Attacker Framework

Stjepan Picek[1], Annelie Heuser[2], and Sylvain Guilley[3]

[1] Delft University of Technology, Mekelweg 2, Delft, The Netherlands
s.picek@tudelft.nl
[2] Univ Rennes, Inria, CNRS, IRISA, France
annelie.heuser@irisa.fr
[3] Secure-IC S.A.S
sylvain.guilley@secure-ic.com

Abstract. Profiling side-channel attacks represent the most powerful category of side-channel attacks. There, we assume that the attacker has access to a clone device in order to profile the device. Additionally, we assume the attacker to be unbounded in power in an effort to give the worst-case security analysis. In this paper, we start from a different premise and consider an attacker in a restricted setting where he is able to profile only a limited number of measurements. To that end, we propose a new framework for profiling side-channel analysis that we call the Restricted Attacker framework. With it, we enforce the attackers to really conduct the most powerful attack possible but also we provide a setting that inherently allows a more fair analysis among attacks. Next, we discuss the ramifications of having the attacker with unbounded power when considering neural network-based attacks. There, we are able to prove that the Universal Approximation Theorem can result in neural network-based attacks being able to break implementations with only a single measurement. Those considerations further strengthen the need for the Restricted Attacker framework.

Keywords: Side-channel analysis, Machine learning, Deep learning, Restricted Attacker framework

1 Introduction

Side-channel analysis (SCA) is a threat that exploits weaknesses in physical implementations of cryptographic algorithms rather than the algorithms themselves [1]. SCA exploits any unintentional leakage observed in physical channels like timing [2], power dissipation [3], electromagnetic (EM) radiation [4], etc. Profiling SCA performs the worst-case security analysis by considering the most powerful side-channel attacker that has access to an open (in the sense that the keys can be chosen by the attacker) clone device.

Usually, we consider an attacker in the setting where he has unbounded power, e.g., he can obtain any number of profiling or attacking traces and he

has unlimited computational power. Yet, it is clear that every attacker must be bounded and much less powerful than given by such assumptions. The difference between the assumed power of an attacker and his real power can introduce a serious problem into the profiling side-channel analysis framework and thus into the evaluation and classification of attacks. A common way to conduct a profiling attack is to use the template attack. Template attack is known to be the most powerful one from the information theoretic point of view when the attacker has an unbounded number of traces in the profiling phase [5, 6]. If the template attack is the most powerful one, why do we care about other attack techniques? The reason is that the template attack is the most powerful one only if some difficult constraints are fulfilled, which does not occur often in practice. In the last two decades, besides template attack and its variants [7, 8], the SCA community started using various machine learning techniques to conduct profiling attacks. The results with machine learning proved to be highly competitive when compared to template attack and actually in many scenarios such techniques surpassed template attack. Often in these scenarios, the number of profiling traces is arbitrarily limited and no clear guidelines on the limitation are given or discussed. In the last few years, the SCA community also started to experiment with various deep learning techniques where the performance of such techniques bested both template attack and other machine learning techniques. Again, no clear guideline on the number of profiling traces was given or investigated.

In the machine learning domain, there is a well-known theorem called the Universal Approximation Theorem, which informally states that a feed-forward neural network with a single hidden layer containing a finite number of neurons can approximate a wide range of functions with any desired level of error. With such a theorem, and considering a powerful ("unbounded") SCA attacker, we must assume he is able to always approximate any function describing the leakage of implementation. And since the theorem states that the approximation is done to any desired level of error, this would result in an attacker able to break any implementation.

In order to devise a reliable framework, we need to limit the power of an attacker to be able to draw meaningful conclusions. Still, it is important to limit the attacker in a sense that will make the framework more practical while giving the attacker enough power to obtain relevant results. Note, there is another reason why it would be beneficial to limit the power of the attacker. By setting the attacker in a scenario where he can obtain an unlimited number of measurements, instead of making the attacker as powerful as possible, we actually allow him to use weaker attacks. More precisely, he can use a larger set of measurements to compensate for less powerful profiling models. Consequently, a proper framework would consider an attacker that has as limited number of measurements as possible and is yet able to break the implementation. In this paper, the notion of a model is equivalent to the notion of mapping (i.e., a function), while the framework is the general setting that uses models in order to fulfill certain goals.

As far as we are aware, there are no previous works considering such restricted attacker frameworks. When the attacker is restricted, it is either set as one of a number of tested scenarios (e.g., testing the performance of a classifier with a different number of measurements in the training phase) or motivated with some limitation in the data acquisition. One example of a limitation would be a device with a counter on the number of measurements. Still, all those works have in common that they test restricted attackers as one test case and not as a "proper" framework for profiling SCA.

In this paper, we present two main contributions:

1. We propose a new framework for the profiling side-channel analysis where we restrict the ability of an attacker to obtain measurements in the profiling phase. Note, such attacker is still powerful from the computational perspective as well as from the perspective of the models he can build.
2. With some constraints on the type of SCA leakage, we show that a neural network can break the cryptographic implementation with a single measurement.

The rest of this paper is organized as follows. In Section 2, we discuss the available techniques to conduct profiling SCA. In Section 3, we present related works and in Section 4, the currently used frameworks for profiling SCA. Section 5 introduces our new framework – The Restricted Attacker framework. In Section 6, we further develop on the proposed framework and we show its relevance for neural network-based approaches. Additionally, by connecting the theoretical results from the machine learning domain and SCA, we show how neural networks can be even more powerful than one would intuitively think. In Section 7, we discuss the significance of our findings and the similarities/differences between profiling attacks and supervised learning. Finally, in Section 8, we conclude the paper.

2 On the Methods to Perform Profiling Side-channel Analysis

In this section, we formally introduce profiling side-channel analysis and supervised machine learning while maintaining the usual phrasing of the respective domain.

2.1 Setting

Let calligraphic letters ($\mathcal{X}$) denote sets, capital letters (X) denote random variables over $\mathcal{X}$, and the corresponding lowercase letters (x) denote their realizations. Let k^* be the fixed secret cryptographic key (byte), k any possible key hypothesis, and the random variable T the plaintext or ciphertext of the cryptographic algorithm, which is uniformly chosen.

We consider a scenario where a powerful attacker has a device (usually called the clone device) with knowledge about the secret key implemented and is able

to obtain a set of N profiling traces $X_1, \ldots, X_N$. Using the known secret key and N plaintexts or ciphertexts T_p, he calculates a leakage model $Y(T_p, k^*)$. In this phase, commonly known as the profiling phase, the attacker has available N pairs (X_i, Y_i) with $i = 1, \ldots, N$ which are used to build a profiled model f.

The attack can then be carried out on another device by using the mapping f. For this, the attacker measures additional Q traces $X_1, \ldots, X_Q$ from the device under attack in order to guess the unknown secret key k_a^*. The leakage model is now calculated for all possible key candidates $k \in \mathcal{K}$:

$$Y(T_a, k_1), \ldots, Y(T_a, k_{|\mathcal{K}|}), \tag{1}$$

given Q plaintexts or ciphertexts T_a.

In Figure 1, we depict the profiling side-channel attacks scenario where we distinguish between the profiling phase in which we learn a model f using N measurements and the attacking phase where we use the model f and Q measurements to predict the secret key on the attacking device.

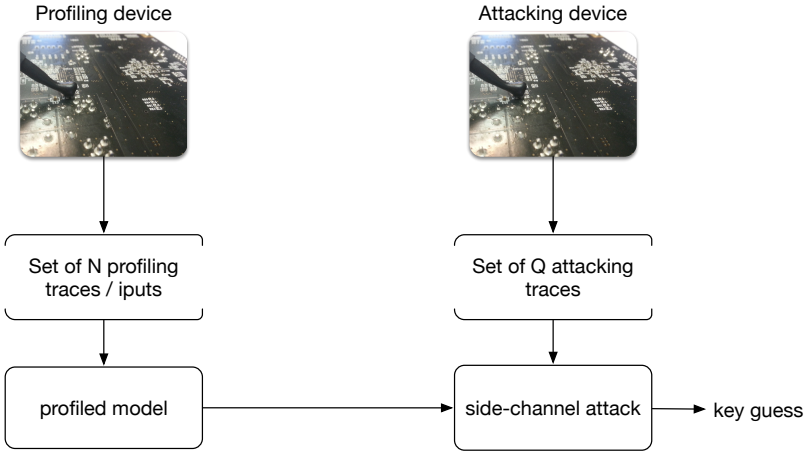

Fig. 1: The profiling side-channel analysis

2.2 Classical Profiling Side-channel Attacks

The best-known profiling attack is the template attack which is based on the Bayesian rule. It works under the assumption that the measurements are dependent among the D features given the target class. More precisely, given the vector of N observed attribute values for x, the posterior probability for each class value y is computed as:

$$p(Y = y|X = x) = \frac{p(Y = y)p(X = x|Y = y)}{p(X = x)}, \tag{2}$$

where $X = x$ represents the event that $X_1 = x_1 \wedge X_2 = x2 \wedge \ldots \wedge X_N = x_N$.

Note that the class variable Y and the measurement X are not of the same type: Y is discrete while X is continuous. So, the discrete probability $p(Y = y)$ is equal to its sample frequency where $p(X = x|Y = y)$ displays a density function. Mostly in the state-of-the art, $p(X = x|Y = y)$ is assumed to rely on a (multivariate) normal distribution and is thus parameterized by its mean $\bar{x}_y$ and covariance matrix Σ_y:

$$p(X = x|Y = y) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_y|}} e^{-\frac{1}{2}(x-\bar{x}_y)^T \Sigma_y^{-1}(x-\bar{x}_y)}. \tag{3}$$

In practice, the estimation of the covariance matrices for each class value y can be ill-posed mainly due to an insufficient number of traces for each class. The authors of [8] propose to use only one pooled covariance matrix to cope with statistical difficulties and thus a lower efficiency. Accordingly, Eq. (3) changes to

$$p(X = x|Y = y) = \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} e^{-\frac{1}{2}(x-\bar{x}_y)^T \Sigma^{-1}(x-\bar{x}_y)}. \tag{4}$$

The works in e.g., [8–10] showed that the pooled version can be indeed more efficient, in particular for a smaller number of traces in the profiling phase.

2.3 Supervised Machine Learning (in SCA)

Machine learning encompasses a number of methods used for classification, clustering, regression, feature selection, and other knowledge discovering methods [11]. A usual division of machine learning algorithms is into two fundamentally different approaches: supervised and unsupervised learning, based on the desired outcome of the algorithm. In the rest of this paper, we consider only the supervised learning paradigm but we note that also unsupervised learning (or their combination, the so-called semi-supervised learning) can be used in SCA, see e.g., [12].

The goal for machine learning algorithms is to learn a mapping f, such that $f : \mathcal{X} \to \mathcal{Y}$, given a training set of N pairs (X_i, Y_i). This phase is commonly known as the training phase. The function f is an element of the space of all possible functions $\mathcal{F}$. Since we are interested in scenarios where Y takes values from a finite set (discrete labels), we talk about classification. To classify new examples, we can use two families of algorithms for supervised learning. Generative algorithms try to model the class-conditional density $p(X|Y)$ by some unsupervised learning procedure. Discriminative algorithms do not estimate how X_i is generated, but instead estimate $p(Y|X)$.

Differing from the profiling SCA where there are only a handful of used algorithms, in supervised learning for SCA there is a plethora of algorithms that show good performance. For a more detailed list of different algorithms, we refer readers to Section 3. Next, we give details about multilayer perceptron since it has an important role for the Universal Approximation Theorem we use in the following sections.

The multilayer perceptron (MLP) is a feed-forward neural network that maps sets of inputs onto sets of appropriate outputs. MLP consists of multiple layers of nodes in a directed graph, where each layer is fully connected to the next one. To train the network, the backpropagation algorithm is used, which is a generalization of the least mean squares algorithm in the linear perceptron [13]. An MLP consists of three or more layers (since input and output represent two layers) of nonlinearly-activating nodes [14]. Note, if there is more than one hidden layer, the architecture can be already considered deep, i.e., we are using deep learning. We depict a multilayer perceptron in Figure 2. Here, every node is a perceptron unit as depicted in Figure 3. Note that the output of a perceptron is a weighted sum of N inputs x_i evaluated through an activation function A:

$$Output = A(\sum_{i=1}^{N} w_i \cdot x_i). \tag{5}$$

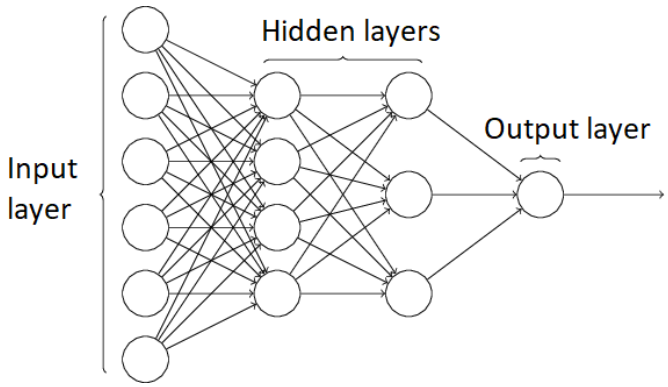

Fig. 2: Multilayer perceptron

3 Profiling Side-channel Evaluation and Techniques Used

Profiling side-channel attacks, especially those based on machine learning received a significant amount of attention in the SCA community in the last decade or so. There, researchers reported a number of scenarios where machine learning is able to achieve top results in attacking cryptographic implementations. Additionally, in the last few years, deep learning emerged as a powerful alternative where results surpassed both template attack and other machine learning techniques. In this section, we first give a brief overview of relevant works and then we give a more exhaustive list of works considering machine learning techniques and the supervised learning paradigm. The purpose of that list is twofold: 1) to demonstrate that machine learning techniques are of high interest to the SCA

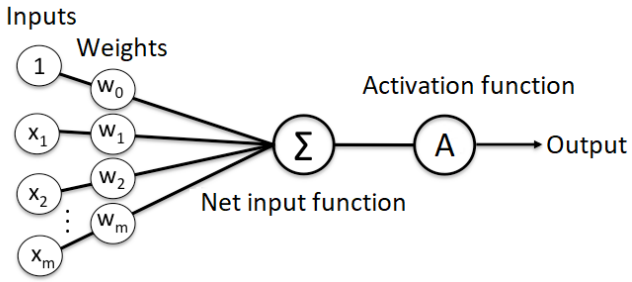

Fig. 3: Perceptron unit

community and 2) to facilitate future research by allowing easier identification of related works.

When considering profiling (or supervised) SCA, we see that the community started with a template attack and its variants. Afterward, Support Vector Machines (SVM) and Random Forest (RF) attracted most of the attention. Still, in the last few years, multilayer perceptron (MLP) is used more and more. Finally, in 2016, the SCA community started experimenting with convolutional neural networks and from that moment, most of the other machine learning techniques serve only as a means to compare. Besides that, we see sporadic attempts to extend the pool of used methods but for now, they do not seem to attract much attention.

In Table 1, we give a list of machine learning techniques used in profiling SCA and corresponding references. Naturally, despite the fact that we tried to be exhaustive, it is unlikely that these works are the only ones that exist. Note, if template attack is also used in those papers, we note that in the table but we do not list papers that do not use machine learning technique. First, one can observe a significant number of papers considering neural networks. Additionally, we see that Support Vector Machines are well explored but more careful inspection shows that in the last few years this technique represents the baseline machine learning case and not state-of-the-art. Finally, most of the works consider a different number of available traces in the profiling phase, which makes it difficult to compare.

4 Existing Frameworks for Side-channel Evaluation

In this section, we discuss the currently used frameworks for profiling side-channel analysis in scientific works and for certification.

4.1 Scientific Metrics

The most common evaluation metrics in the side-channel analysis are success rate (SR) and guessing entropy (GE) [42]. Success rate defines the estimated

Table 1: Overview of profiling side-channel attacks used in literature.

Algorithm	Reference
Naive Bayes and its variants	[9, 15–19]
Random Forest	[6, 15–17, 19–26]
Rotation Forest	[10, 17, 18, 27]
XGB	[18]
MultiBoost	[10]
Self-organizing maps	[22]
Support Vector Machines	[6, 10, 17, 19–25, 27–31]
Multivariate regression analysis	[23, 24, 32]
Multilayer Perceptron	[15, 16, 18–21, 33–39]
Convolutional Neural Networks	[18, 20, 21, 33, 39–41]
Autoencoders	[21]
Reccurent Neural Networks	[21]
Template Attack and its variant	[6, 9, 10, 17, 19–25, 27, 28, 30, 39–41]
Stochastic attack	[20, 23, 24]

averaged probability of success. The average key rank is given by the guessing entropy. More precisely, GE states the average number of key candidates an adversary needs to test in order to reveal the secret key after conducting a side-channel analysis. In particular, given Q amount of samples in the attacking phase, an attack outputs a key guessing vector $g = [g_1, g_2, \ldots, g_{|\mathcal{K}|}]$ in decreasing order of probability with $|\mathcal{K}|$ being the size of the keyspace. So, g_1 is the most likely and $g_{|\mathcal{K}|}$ the least likely key candidate. The guessing entropy is the average position of k_a^* in g over multiple experiments. The success rate is defined as the average empirical probability that g_1 is equal to the secret key k_a^*.

In practice one may consider leakage models $Y(\cdot)$ that are bijective functions, thus each output probability calculated from the classifiers for $Y(k)$ directly relates to one single key candidate k. In case $Y(\cdot)$ is not bijective, several key candidates k may get assigned with the same output probabilities, which is why on average a single trace attack ($Q = 1$) may not be possible in case of non-bijective leakage models. Further, to calculate the key guessing vector g over Q amount of samples, the (log-)likelihood principle is used.

SR and GE are used for practical evaluations in both non-profiling and profiling scenarios. Typically, they are given over a range of the number of traces used in the attacking phase (i.e., for $q = 1, 2, \ldots, Q$). In case these metrics are used in profiling scenarios, there are no clear guidelines on how to evaluate attacks. Most of the time, the number of training samples N in the profiling stage is (arbitrary) fixed which makes comparisons and meaningful conclusion on side-channel attacks or resistance of implementations hard and unreliable in most scenarios.

A more theoretical framework has been introduced by Whitnall and Oswald [43, 44] that aims at comparing distinguishing powers instead of estimators

of attacks. Accordingly, the size of the profiling dataset N does not play any role in this framework. The most popular metrics of the framework are the relative and absolute distinguishing margins in which the output score of the correct key and the value for the highest ranked alternative are compared.

Another approach to compare side-channel attacks uses closed-form expressions of distinguishers [45], which enables to make conclusions about distinguishers without the requirement of actual measurements. Unfortunately, only a few closed-form expressions of distinguishers have been achieved so far.

Typically, to assess the performance of the machine learning classifiers one uses accuracy:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{6}$$

TP refers to true positive (correctly classified positive), TN to true negative (correctly classified negative), FP to false positive (falsely classified positive), and FN to false negative (falsely classified negative) instances. TP, TN, FP, and FN are well-defined for hypothesis testing and binary classification problems. For multi-class classification, they are defined in one class–vs–all other classes manner. A more detailed comparison between accuracy and guessing entropy/success rate is given in [20], which details that accuracy may not always be a good metric for side-channel analysis.

While all these previous metrics are relevant in some contexts and scenarios, a different approach is required for profiling attacks. This becomes even more clear when looking at practical evaluation used in standardization processes as detailed next.

4.2 Practical Evaluation Testing

In practice, there are two main practical schemes: 1) Test-based schemes, such as NIST FIPS 140 [46] and ISO/IEC 17825 [47] and 2) Evaluation-based schemes, such as Common Criteria (CC, ISO/IEC 15408 [48]).

It is interesting that both FIPS 140 and CC pay attention to the limited amount of resources spent. When considering FIPS 140 / ISO/IEC 17825, the requirement is more on the attacking traces, but regarding CC, the evaluation of attacks is considered under two phases: identification (which matches with the training phase in the context of side-channel attacks) and exploitation (which matches with the attacking phase in the context of side-channel attacks). Strictly speaking, the distinction is for CC version 2, but it still implicitly holds for version 3. Several factors are considered for the quotation of attacks, namely: elapsed time, expertise, knowledge of the Target Of Evaluation (TOE), access to TOE, equipment, open samples. The first factor, elapsed time, has a direct connection with the acquisition of traces in the profiling phase. Indeed, according to the guidance "Application of Attack Potential to Smartcards" [49]), the score is considered:

– 0 if the profiling of the traces can be performed in less than one hour,

– 1 if the profiling of the traces can be performed in less than one day,
– 2 if the profiling of the traces can be performed in less than one week,
– 3 if the profiling of the traces can be performed in less than one month,
– 5 if the profiling of the traces cannot be performed in less than one month.

Accordingly, we see that the CC guidance favors attacks which are realized with as little profiling effort (besides time, this profiling effort could be also interpreted through the number of acquired measurements) as possible.

One reason is that the collection of side-channel traces becomes less reliable after a long period of time. Indeed, some trend noise must be added to the side-channel traces (due to temperature and environmental conditions evolution over time). This has for instance been characterized by Heuser et al. in [50], where it is proven that trend noise drastically impedes SCA. Similar findings are also confirmed by Cao et al. [51].

5 The Restricted Attacker Framework

As already stated, current frameworks for profiling SCA assume the attacker to be unbounded in his power. By doing this, we aim to provide the worst-case scenario for the designer, which helps in the proper assessment of the risk. Naturally, despite the fact that the attacker is considered unbounded, he is always bounded but those bounds are set ad-hoc and there are no clear directions one should follow when modeling the realistic attacker.

We start by examining the three components of a successful attack. The worst-case (strongest) attacker will be unbounded in all three components. At the same time, fulfilling only one or two of them accounts for more realistic settings one can encounter in practice:

1. Quantity - there must be sufficient measurements in the profiling/testing phase to conduct the attack, i.e., to build a reliable model that generalizes to the unseen data.
2. Quality - the measurements need to be of sufficient quality to conduct the attack. This could be translated into the condition that the SNR should be sufficiently high, or that the data need to have all information required to correctly model the leakage. Finally, this component also includes the quality of the leakage model, i.e., that the considered leakage model provides sufficient information as well as the distribution of leakages.
3. Learnability - the attacker needs to be able to learn the model. This perspective also accounts for learning the best possible parameters of the model. The learnability is naturally connected with the quantity and quality parameters.

Obviously, we should not influence the quality parameter: if the attacker is able to obtain measurements, those measurements should be of the best possible quality. Similarly, we cannot limit the attacker's power from the perspective of learnability. The attacker must be able (at least in theory) to learn the model (and its optimal parameters) and he should also possess enough computational

power to do so. Finally, when discussing the quantity parameter, we see that we can (and we must) limit the attacker in the number of measurements he is able to obtain in the profiling phase. There are two reasons for that:

1. By giving the attacker the option to use as much as possible measurements, we "allow" him to design a weaker model. Indeed, additional measurements will sometimes help the attacker to reach good results despite the fact that he uses a sub-optimal attack. Differing from that, if we limit the attacker in the number of traces he has on his disposal, we require from him to develop a strong attack if he wants to succeed.
2. On a more general level, the attacker who is unbounded in his capabilities is able to break cryptographic implementations with a single measurement under certain assumptions. This suggests that ultimately, there is nothing the designer can do to stop the attacker. Naturally, being able to mount such attack is often not possible in practice since all the components from Listing 5 need to be fully fulfilled.

Limiting the number of measurements is also a realistic occurrence in practical scenarios, as the attacker may be limited by time, resources, and also face implemented countermeasures which prevent him from taking an arbitrarily large amount of side-channel measurements, while knowing the secret key of the device.

How do we limit the attacker, i.e., the quantity of data? We need to consider an attacker who is able to perform a successful attack with the smallest possible number of measurements N, where success is defined over a performance metric ρ with a threshold δ. For example, ρ could be reaching GE < 10, or SR > 0.9, which are common threshold values in the side-channel analysis, see, e.g., [20].

Recall, the goal for machine learning is to learn a mapping (model) f from $\mathcal{X}$ to $\mathcal{Y}$, i.e., $Y \leftarrow f(X, \theta)$ where X are samples drawn i.i.d. from set $\mathcal{X}$ and where the cardinality of X equals N. Let $theta$ be the parameters of the model that result in the best possible approximation, X_p is the input to the model (measurements), Y_a labels associated with X_a, and $c(\theta, X_a, Y_a)$ is the cost used to train the model. Additionally, let $\mathbf{g}_{Q,f} = [g_1, g_2, \ldots, g_{|\mathcal{K}|}]$ be the guessing vector from the profiling side-channel attack using Q measurement traces in the attacking phase and the model f build in the profiling phase as an input. Then, $\rho(g_Q, k_a^*)$ represents the performance metric of the profiling side-channel attack using the secret key k_a^* to evaluate the success.

The Restricted Attacker framework aims at minimizing the number of profiling traces N to model the function f, such that the performance metric is still below (or above) a certain threshold δ:

$$\min\{N \mid \rho(g_{Q,f}, k_a^*) < \delta\}, where\ N \geq 1. \tag{7}$$

Algorithm 1 gives the pseudocode of the evaluation in the Restricted Attacker framework and an example is given Example 1.

Example 1. A common performance metric used in the side-channel analysis is, for example, the guessing entropy with a threshold $\delta = 10$. Therefore, in the

Input : Profiling and attacking device to collect traces from
Output : Minimum number of profiling traces N

Capture a testing dataset (with secret key k_a^*). Its size Q depends on the expected performance of the attack. For instance, this test dataset can be as small as one trace!
Select a performance metric ρ and a threshold value δ, e.g., GE < 10
Training_set $\leftarrow \varnothing$
while *True* **do**
    Capture one trace `// A speed-up can be obtained by advancing faster, e.g., 10 by 10 traces`
    Append them to Training_set, $N = N + 1$
    Perform Training (which yields a model f)
    Make a key guess k from the Testing_set with Q measurements
    if $\rho < \delta$ **then**
        break `// model is good enough`
return *Minimum number of profiling traces N*

Algorithm 1: The Restricted Attacker framework

Restricted Attacker framework, one would compute the minimum number of profiling traces N to reach a guessing entropy below 10 for a fixed number of Q attacking traces. Typically, Q is ranging over a set of values. Experimental results are discussed in Section 7.

Notice that so far the cost of training the model $c(\theta, X_a, Y_a)$ is not limited but we assume that X and Y are of sufficient quality (inherently) and quantity (by the Restricted Attacker framework) for a model to generalize well to the unseen data. Once the attacker is restricted, we have a framework enabling us to conduct a more fair comparison. Indeed, to improve on the results, now one needs to reach at least the same level of performance as measured with ρ but by having smaller sets X and Y. It is easy to see that now we require our attacker to build a more powerful model if he wants to surpass the performance as obtained up to that moment.

Next, one could also add the computational power of the attacker to our framework. Naturally, this leaves the possibility to leverage whether computational power or the number of measurements in the profiling phase is more important. We believe it is easier to estimate the number of measurements and consequently consider this as the most important criterion. Still, two models resulting in the same performance can and must be compared with respect to the used computational power to obtain such results. The model that spent fewer resources performs better. What happens if two models exhibit very similar performance but use a radically different amount of resources. In this case, a Pareto front of solutions (i.e., a set of nondominated solutions) needs to be given where the designer is then able to decide on a proper trade-off.

Finally, we reiterate that our framework is not designed to force the attacker to use a small number of measurements in the profiling phase. Rather, it forces the attacker to evaluates what is the smallest number of traces he requires in order to conduct a successful attack.

6 The Restricted Attacker Framework in Neural Network-based Attacks

In this section, we investigate machine learning techniques in SCA and we show the relevance of the Restricted Attacker framework when considering neural network approaches. Next, we present several results proving that neural networks are able to provide the most powerful attacks for profiling SCA, provided certain assumptions are fulfilled.

6.1 Universal Approximation Theorem

The Universal Approximation Theorem proves that for any Borel measurable function f (where Borel measurable mapping $f : \mathcal{X} \to \mathcal{Y}$ between two topological spaces has the property that $f^{-1}(\mathcal{A})$ is a Borel set for any open set $\mathcal{A}$), there exists a feed-forward neural network, having only a single hidden layer with a finite number of neurons, which uniformly approximates f within an arbitrary nonzero amount of error ϵ [52, 53].

For this theorem to hold, we require only mild assumptions on activation functions (such that saturate for both very negative and very positive arguments) and naturally, the network needs to have enough hidden units. Note, the theorem was also proved for a wider class of activation functions, including rectified linear unit [54].

As a consequence of the Universal Approximation Theorem, we know that a multilayer perceptron that has enough nodes will be able to represent any Borel measurable function. Naturally, there is no guarantee that the machine learning algorithm will be actually able to learn that function. Indeed, if this theorem is correct, the question is why it is still difficult (in many practical applications) to obtain even a decent performance of a classifier, let alone approximation to an arbitrary ϵ. For instance, as given in Section 3, there are numerous works using multilayer perceptron (which is a feed-forward network) where more than a single hidden layer is used and yet the results are far from optimal.

The main problem is that the Universal Approximation Theorem does not consider the algorithmic learnability of feed-forward networks. The theorem says that the number of nodes in the network is finite, but does not specify how many nodes do we actually need. There are some results on bounds on the number of nodes, but unfortunately, in the worst case scenario, an exponential number of

[3] A Borel set is any set in a topological space that can be formed from open sets (a set S is open if every point in S has a neighborhood lying in the set) through the operations of countable union, countable intersection, and relative complement.

nodes is needed [55]. Additionally, from a practical perspective, the learnability of the model also heavily depends on the quality and quantity of data we have at our disposal. Here, by quality, we consider that our data need to have all information needed to correctly model the function f and by the quantity that we need to have sufficient information to build a reliable model that will generalize to the unseen data.

Up to now, we mentioned only feed-forward networks and how they fit the Universal Approximation Theorem. Yet, we stated in Section 1 that convolutional neural networks were recently used to achieve state-of-the-art performance in the SCA domain. Consequently, a natural question is to ask whether the Universal Approximation Theorem is also valid for convolutional neural networks. We give a small example. Let us consider a feed-forward network with a single hidden layer that has A inputs and B outputs. To realize such an architecture, we require a weight matrix $W \in \mathbb{R}^{B \times A}$. If we assume that the convolution is applied only to the input and there is no padding, it is rather straightforward to see that we can simulate this feed-forward network with only two convolutional layers. In the first layer, we have $B \times A$ filters of shape A. The element a of filter b, a is equal to $W_{b,a}$ with the rest being zeros. This layer transforms the input into BA-dimensional intermediate space where every dimension represents a product of weight and its corresponding input. The second layer contains B filters of shape BA. The elements $bA \dots (b+1)A$ of filter b are ones while the rest are zeros. This layer performs the summation of products from the previous layer. Naturally, for this construction, we assumed some conditions that are not realistic but we show this as a motivating example that the Universal Approximation Theorem can be applied for other types of neural networks also. We emphasize that various functions can be more efficiently approximated by architectures that have greater depth, which is a reason why deep learning is able to exhibit such performance.

More formally, D. Yarotsky showed that any translation equivariant function can be approximated arbitrarily well by a convolutional neural network given that it is sufficiently wide [56]. This has a direct analogy to the Universal Approximation Theorem.

6.2 From the Universal Approximation Theorem to Optimal Side-channel Attack

Conjecture 1. A side-channel leakage can be modeled by Borel measurable function.

Recall, Borel measurable function is a mapping $f : \mathcal{X} \to \mathcal{Y}$ between two topological spaces with the property that $f^{-1}(\mathcal{A})$ is a Borel set. A Borel set is any set in a topological space that can be formed from open sets (a set $\mathcal{S}$ is open if every point in $\mathcal{S}$ has a neighborhood lying in the set) through the operations of countable union, countable intersection, and relative complement for any open set $\mathcal{A}$.

Clearly, all continuous functions (i.e., functions defined on $\mathbb{R}$) are Borel functions (but not all Borel functions are continuous). Unfortunately, in SCA one uses an oscilloscope in the acquisition process and they have a finite precision, which makes the resulting function a discrete one.

Let us consider power or electromagnetic side-channel. As mentioned, the oscilloscope samples only a discrete time and quantifies the measurements. Such measurements are a series of finite values, which may not be Borel measurable as such. However, before sampling and quantization, the signal was a physical quantity, which is Borel measurable. Indeed, it is obtained from the RLC-filtering of some physical quantity [57, Figure 2], itself obtained as the resolution of differential equations of electronics/mechanisms. It is therefore possible, as a pre-processing step, to interpolate and smooth the SCA measurements to make them continuous, hence eligible to be Borel measurable. More intuitively, there are infinitely many continuous functions that can describe a finite number of samples.

Additionally, we can make use of Lusin's theorem, which states that every measurable function is continuous on nearly all its domain [58]. More formally, a function $f : \mathcal{X} \to \mathbb{R}$ is measurable if for every real number a, the set $x \in \mathcal{X} : f(x) > a$ is measurable. Practically, this means that any function that can be described is measurable.

Lemma 1. *If a side-channel leakage is Borel measurable (see Conjecture 1)), then a feed-forward neural network with a single hidden layer consisting of a finite number of neurons can approximate any side-channel leakage to a desired nonzero error.*

Proof. Straightforward from the Universal Approximation Theorem. □

Theorem 1. *A profiling side-channel attack where the Universal Approximation Theorem holds (i.e., where Lemma 1 holds), can succeed in breaking an implementation with only a single measurement.*

Proof. Trivial. If SCA leakage can be approximated to a desired (nonzero) amount of error, it means (provided that we use the appropriate leakage model) that we need only a single measurement to obtain the key information. □

There is also a simple alternative proof. Since we know that in the ideal case, template attack can break an implementation with a single measurement, then it is enough for neural networks to be able to approximate such a model (template) built by the template attack. If a neural network can approximate a template with a desired nonzero amount of error, then such a network can simulate the behavior of a template attack. We emphasize that for the template attack to be able to break an implementation in a single measurement some conditions must be met. Similar as we discussed the quality and quantity for machine learning, we can extend it to template attack. There, in the quality, we need to account for the level of noise, leakage model, and distribution of leakage. In the quantity, we assume to have a sufficient number of measurements for template attack to work (and break the implementation with a single measurement).

One could ask if neural networks are able to (theoretically and under some assumptions) break the implementation in a single measurement, how is that aligned with what we know about template attack. Indeed, we already said that the template attack is the most powerful one from the information theoretic point of view and yet, now we claim that neural networks are able to display the same performance. We believe this not to be in contradiction due to heavy assumptions on both techniques. For template attack, we require an unlimited number of traces, which is naturally impossible to have. On the other side, for neural networks, we do not consider the algorithmic learnability, where the learning process can fail from several reasons [13]. Still, we note that the breaking of cryptographic implementations in a single measurement is not something that is only possible in theory, see e.g., [40] where convolutional neural networks and template attack are able to break different implementations in a single measurement.

Finally, although in this paper we do not discuss machine learning techniques except neural networks, other machine learning techniques would also benefit from the Restricted Attacker framework. Indeed, we do not require the theoretical promise of being able to break implementation with a single measurement. Rather, we require a setting that limits the attacker's power in the profiling phase, which is independent of the considered attack techniques.

7 Discussion

In this section, we first provide insights from our new framework. Next, we discuss the difference between the profiling attacks and attacks based on supervised learning. Finally, we give directions for future work.

7.1 Insights from the Restricted Attacker Framework

The Restricted Attacker framework does not only enable us to compare side-channel attacks but also gives a fair comparison between leakage models. For profiling side-channel attacks it is often assumed to consider the most accurate leakage model, i.e., using the intermediate value as class variables which results into 2^b classes where b is the number of considered bits. In an unsupervised setting, using the Hamming weight or Hamming distance model is a common choice which results in only $b + 1$ classes. Clearly, using only $b + 1$ Hamming weight/distance classes to guess a key value in $\{0, \ldots, 2^b\}$ cannot result in a single trace attack on average. However, it is often overlooked that using the Hamming weight/distance models may require fewer traces in the profiling phase to gain good quality estimates of the models. It is therefore not straightforward to determine what leakage model is more suitable. Consequently, to fairly give a comparison one should include a dependency on the number of traces in the profiling phase as in the Restricted Attacker framework.

Figure 4 illustrates an example using the AES software implementation. We use the publicly available traces of the DPAcontest v4, which is a masked AES

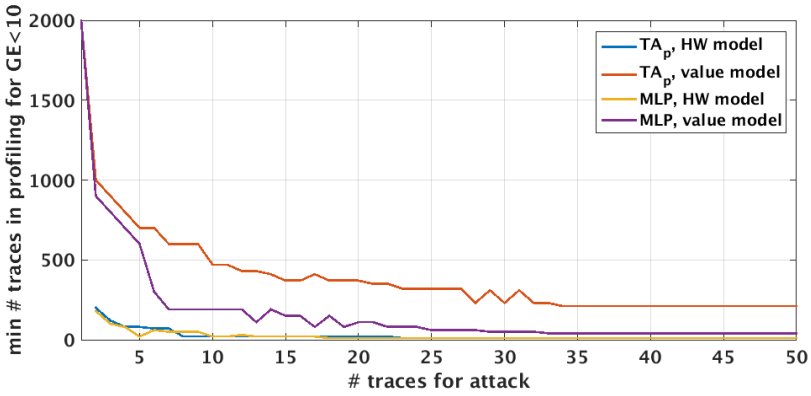

Fig. 4: TA pooled and MLP evaluation for HW and value model

software implementation [59]. We assume the mask is known and thus turn it into an unprotected scenario. As a metric, we utilize the guessing entropy (GE) and in particular, give the minimum number of profiling traces in order to reach GE < 10. As a side-channel attack, we use the pooled template attack (TA_p) and the multilayer perceptron (MLP) consisting of 5 hidden layers with the number of neurons equal to $(50, 25, 10, 25, 50)$ that uses $ReLU$ activation function. As a leakage model, we use the intermediate value model (resulting in 256 classes) and the Hamming weight (HW) model (resulting into 9 classes) as AES operates on $b = 8$ bits.

One can observe that only when using the intermediate value model it is possible to reach a GE < 10 with a single trace. In our experiments, we could not reach this threshold using the HW model with $Q = 1$. On the other side, it is clear that if the HW model is able to succeed (i.e., number of attacking traces > 1) then the HW model requires much less profiling traces. Comparing TA_p and MLP we see that when using the HW model no real difference is observable. For the intermediate value model, it can be seen that for $Q = 1, 2, \ldots, 5$, MLP and TA_p require approximately the same amount of profiling traces N to reach GE < 10. When $Q > 5$, we observe that MLP requires less profiling traces to reach the same performance as TA_p, which is very relevant information, for example, for evaluation labs. Moreover, one can choose a trade-off between profiling traces N and attacking traces Q while still being able to perform a successful attack.

Note that, the example given in Figure 4 comes from a low-noise implementation setup where one does not expect interesting and decisive results as the classification task is "too easy". Still, with the Restricted Attacker framework, one can clearly define the strength of both leakage models and make conclusions on N and Q even with a high SNR.

7.2 Profiling SCA vs Supervised Learning

It is quite common to consider profiling SCA to be the same as supervised machine learning. Indeed, such a perspective is true for most of the published works but there can be certain differences. From the machine learning perspective, supervised learning represents a scenario where a certain number of examples with a known label is obtained and used to train the machine learning model. For the training phase in machine learning, we do not generally assume how the measurements are obtained. We only assume there are reasonably accurate so we are able to build a model that generalizes to the unseen data.

Differing from that, in SCA in the profiling phase we assume that the same (type of) device is used to acquire knowledge about the device. Such knowledge can be translated into a template or a model that is used in the attack phase. Recently, B. Timon introduced a concept of non-profiled deep learning (or more generally, machine learning). There, he does not build a model from the clone device but actually estimates the behavior of a device with a non-profiled approach and then uses supervised machine learning to build a model and verify it [33]. With this approach, he does not conduct a profiling phase but is conducting supervised learning. Consequently, the name of no-profiled deep learning could be regarded as a misnomer since machine learning (in the strict sense) cannot be profiling or non-profiling but supervised or unsupervised. To conclude, every profiling attack is a supervised attack but not all supervised attacks are profiling ones (since non-profiling attacks can be supervised).

Afterward, the results from the profiling/training are used to test the measurements where the label is not known. The testing phase (as in machine learning) can be the same as the attack phase (in SCA) but again, this is not mandatory. We consider the test phase to be the same as the attack phase if we are able to use the results to break the implementation. Considering the metrics in SCA and machine learning, attacking and testing are the same if and only if the machine learning metric gives a direct insight into the key guessing performance. This is a common occurrence in, for instance, the intermediate value model.

7.3 Future Directions

Besides using the Restricted Attacker framework to compare profiling side-channel attacks on various implementations, we emphasize as a future direction the comparison of attacks in the presence of various side-channel countermeasures. Such a study would highlight if different types of side-channel countermeasures differ in their complexity of profiling where it may be of particular interest to increase the complexity of the profiling phase more than the attacking phase or to find a suitable trade-off to protect against powerful attackers.

Another direction that could be interesting to include in the Restricted Attacker framework is the analysis of the complexity of side-channel attacks. Naturally, some attacks are less complex in terms of resources than others. For example, in most cases, the template attack is less complex than a deep neural network. Taking required resources (time, memory, etc.) will give more detailed

information for the Restricted Attacker framework, such that reasonable trade-off, in particular for evaluation labs, can be made.

Recently, there were several papers proposing to conduct data augmentation techniques in order to construct additional, synthetic measurements to be used in the profiling phase [20, 41, 60]. Our framework does not limit the use of such synthetic examples in the sense that those measurements are not counted in the profiling set N since they are built from the original measurements. It would be interesting to investigate what are the limits of efficiency for such data augmentation techniques in SCA, i.e., can we construct good, synthetic examples even from a limited number of real measurements.

Finally, discussing the unbounded attacker can be problematic if not considering what the concept of unbounded means for machine learning. For instance, how realistic is to expect to have a neural network either large enough or deep enough to (almost) perfectly approximate a function. Or, since the attacker is unbounded in his power, we must assume he will be able to learn a model, i.e., to find the best parameters of the model. In machine learning, we usually assume i.i.d. samples, but what happens if the attacker has the capability to take only those measurements that would improve his attack, i.e., to increase the quality of his measurements. In practice, we do not observe that the scenarios given here appear often but we still need to consider them if the attacker is unbounded in his power. In future work, we aim to address issues arising there and provide recommendations on how to further limit the attacker.

8 Conclusions

In this paper, we discuss how to limit the power of the attacker when considering the profiling side-channel analysis. We argue that having the unbounded attacker, while not being realistic can also have negative effects on the way how side-channel analysis is performed. We propose a new framework, called the Restricted Attacker framework where we limit the amount of the measurements that the attacker can acquire in the profiling phase. Next, we connect the notion of the unbounded attacker with the Universal Approximation Theorem and we show that because of it, the attacker could be able to break implementations with only a single measurement, provided that some conditions are met.

Naturally, this does not occur often in practice but we still consider the "race" for the most powerful attacks meaningless if the theory indicates that breaking an implementation in a single measurement is possible. We consider our new framework not only more realistic but also more adept for experimental evaluations since it allows to compare different results in a more unified way.

References

1. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer (December 2006) ISBN 0-387-30857-1, `http://www.dpabook.org/`.

2. Kocher, P.C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Proceedings of CRYPTO'96. Volume 1109 of LNCS., Springer-Verlag (1996) 104–113
3. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology. CRYPTO '99, London, UK, UK, Springer-Verlag (1999) 388–397
4. Quisquater, J.J., Samyde, D.: Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In Attali, I., Jensen, T., eds.: Smart Card Programming and Security, Berlin, Heidelberg, Springer Berlin Heidelberg (2001) 200–210
5. Heuser, A., Rioul, O., Guilley, S.: Good is Not Good Enough — Deriving Optimal Distinguishers from Communication Theory. In Batina, L., Robshaw, M., eds.: CHES. Volume 8731 of Lecture Notes in Computer Science., Springer (2014)
6. Lerman, L., Poussier, R., Bontempi, G., Markowitch, O., Standaert, F.: Template Attacks vs. Machine Learning Revisited (and the Curse of Dimensionality in Side-Channel Analysis). In: COSADE 2015, Berlin, Germany, 2015. Revised Selected Papers. (2015) 20–33
7. Schindler, W., Lemke, K., Paar, C.: A Stochastic Model for Differential Side Channel Cryptanalysis. In LNCS, ed.: CHES. Volume 3659 of LNCS., Springer (Sept 2005) 30–46 Edinburgh, Scotland, UK.
8. Choudary, O., Kuhn, M.G.: Efficient template attacks. In Francillon, A., Rohatgi, P., eds.: Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers. Volume 8419 of LNCS., Springer (2013) 253–270
9. Picek, S., Heuser, A., Guilley, S.: Template attack versus Bayes classifier. Journal of Cryptographic Engineering **7**(4) (Nov 2017) 343–351
10. Picek, S., Heuser, A., Jovic, A., Ludwig, S.A., Guilley, S., Jakobovic, D., Mentens, N.: Side-channel analysis and machine learning: A practical perspective. In: 2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017. (2017) 4095–4102
11. Mitchell, T.M.: Machine Learning. 1 edn. McGraw-Hill, Inc., New York, NY, USA (1997)
12. Picek, S., Heuser, A., Jovic, A., Legay, A., Knezevic, K.: Profiled sca with a new twist: Semi-supervised learning. Cryptology ePrint Archive, Report 2017/1085 (2017) `https://eprint.iacr.org/2017/1085`.
13. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016) `http://www.deeplearningbook.org`.
14. Collobert, R., Bengio, S.: Links Between Perceptrons, MLPs and SVMs. In: Proceedings of the Twenty-first International Conference on Machine Learning. ICML '04, New York, NY, USA, ACM (2004) 23–
15. Heuser, A., Picek, S., Guilley, S., Mentens, N.: Side-channel analysis of lightweight ciphers: Does lightweight equal easy? In: Radio Frequency Identification and IoT Security - 12th International Workshop, RFIDSec 2016, Hong Kong, China, November 30 - December 2, 2016, Revised Selected Papers. (2016) 91–104
16. Heuser, A., Picek, S., Guilley, S., Mentens, N.: Lightweight ciphers and their side-channel resilience. IEEE Transactions on Computers **PP**(99) (2017) 1–1
17. Picek, S., Heuser, A., Jovic, A., Legay, A.: Climbing down the hierarchy: Hierarchical classification for machine learning side-channel attacks. In Joye, M., Nitaj, A., eds.: Progress in Cryptology - AFRICACRYPT 2017: 9th International Conference on Cryptology in Africa, Dakar, Senegal, May 24-26, 2017, Proceedings, Cham, Springer International Publishing (2017) 61–78

18. Picek, S., Samiotis, I.P., Kim, J., Heuser, A., Bhasin, S., Legay, A.: On the performance of convolutional neural networks for side-channel analysis. In Chattopadhyay, A., Rebeiro, C., Yarom, Y., eds.: Security, Privacy, and Applied Cryptography Engineering, Cham, Springer International Publishing (2018) 157–176
19. Picek, S., Heuser, A., Alippi, C., Regazzoni, F.: When theory meets practice: A framework for robust profiled side-channel analysis. Cryptology ePrint Archive, Report 2018/1123 (2018) `https://eprint.iacr.org/2018/1123`.
20. Picek, S., Heuser, A., Jovic, A., Bhasin, S., Regazzoni, F.: The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(1) (2019) 209–237
21. Maghrebi, H., Portigliatti, T., Prouff, E.: Breaking cryptographic implementations using deep learning techniques. In: Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings. (2016) 3–26
22. Lerman, L., Bontempi, G., Markowitch, O.: Power analysis attack: An approach based on machine learning. Int. J. Appl. Cryptol. **3**(2) (June 2014) 97–115
23. Lerman, L., Medeiros, S.F., Bontempi, G., Markowitch, O.: A Machine Learning Approach Against a Masked AES. In: CARDIS. Lecture Notes in Computer Science, Springer (November 2013) Berlin, Germany.
24. Lerman, L., Bontempi, G., Markowitch, O.: A machine learning approach against a masked AES - Reaching the limit of side-channel attacks with a learning model. J. Cryptographic Engineering **5**(2) (2015) 123–139
25. Lerman, L., Bontempi, G., Ben Taieb, S., Markowitch, O.: A time series approach for profiling attack. In Gierlichs, B., Guilley, S., Mukhopadhyay, D., eds.: Security, Privacy, and Applied Cryptography Engineering, Berlin, Heidelberg, Springer Berlin Heidelberg (2013) 75–94
26. Najm, Z., Jap, D., Jungk, B., Picek, S., Bhasin, S.: On comparing side-channel properties of AES and chacha20 on microcontrollers. In: 2018 IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2018, Chengdu, China, October 26-30, 2018, IEEE (2018) 552–555
27. Lerman, L., Medeiros, S.F., Veshchikov, N., Meuter, C., Bontempi, G., Markowitch, O.: Semi-supervised template attack. In Prouff, E., ed.: Constructive Side-Channel Analysis and Secure Design, Berlin, Heidelberg, Springer Berlin Heidelberg (2013) 184–199
28. Heuser, A., Zohner, M.: Intelligent Machine Homicide - Breaking Cryptographic Devices Using Support Vector Machines. In Schindler, W., Huss, S.A., eds.: COSADE. Volume 7275 of LNCS., Springer (2012) 249–264
29. Hospodar, G., Gierlichs, B., De Mulder, E., Verbauwhede, I., Vandewalle, J.: Machine learning in side-channel analysis: a first study. Journal of Cryptographic Engineering **1** (2011) 293–302 10.1007/s13389-011-0023-x.
30. Bartkewitz, T., Lemke-Rust, K.: Efficient template attacks based on probabilistic multi-class support vector machines. In Mangard, S., ed.: Smart Card Research and Advanced Applications, Berlin, Heidelberg, Springer Berlin Heidelberg (2013) 263–276
31. Hospodar, G., De Mulder, E., Gierlichs, B.: Least squares support vector machines for side-channel analysis. Center for Advanced Security Research Darmstadt (01 2011) 99–104
32. Sugawara, T., Homma, N., Aoki, T., Satoh, A.: Profiling attack using multivariate regression analysis. IEICE Electron. Express **7**(15) (2010) 1139–1144
33. Timon, B.: Non-profiled deep learning-based side-channel attacks. Cryptology ePrint Archive, Report 2018/196 (2018) `https://eprint.iacr.org/2018/196`.

34. Pfeifer, C., Haddad, P.: Spread: a new layer for profiled deep-learning side-channel attacks. Cryptology ePrint Archive, Report 2018/880 (2018) https://eprint.iacr.org/2018/880.
35. Gilmore, R., Hanley, N., O'Neill, M.: Neural network based attack on a masked implementation of AES. In: 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). (May 2015) 106–111
36. Martinasek, Z., Hajny, J., Malina, L.: Optimization of power analysis using neural network. In Francillon, A., Rohatgi, P., eds.: Smart Card Research and Advanced Applications, Cham, Springer International Publishing (2014) 94–107
37. Yang, S., Zhou, Y., Liu, J., Chen, D.: Back propagation neural network based leakage characterization for practical security analysis of cryptographic implementations. In Kim, H., ed.: Information Security and Cryptology - ICISC 2011, Berlin, Heidelberg, Springer Berlin Heidelberg (2012) 169–185
38. Martinasek, Z., Zeman, V.: Innovative method of the power analysis. Radioengineering **22**(2) (2013)
39. Hettwer, B., Gehrer, S., Güneysu, T.: Profiled power analysis attacks using convolutional neural networks with domain knowledge. In Cid, C., Jr., M.J.J., eds.: Selected Areas in Cryptography - SAC 2018 - 25th International Conference, Calgary, AB, Canada, August 15-17, 2018, Revised Selected Papers. Volume 11349 of Lecture Notes in Computer Science., Springer (2018) 479–498
40. Kim, J., Picek, S., Heuser, A., Bhasin, S., Hanjalic, A.: Make some noise: Unleashing the power of convolutional neural networks for profiled side-channel analysis. Cryptology ePrint Archive, Report 2018/1023 (2018) https://eprint.iacr.org/2018/1023.
41. Cagli, E., Dumas, C., Prouff, E.: Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures - Profiling Attacks Without Preprocessing. In: Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. (2017) 45–68
42. Standaert, F.X., Malkin, T., Yung, M.: A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In: EUROCRYPT. Volume 5479 of LNCS., Springer (April 26-30 2009) 443–461 Cologne, Germany.
43. Whitnall, C., Oswald, E.: A Fair Evaluation Framework for Comparing Side-Channel Distinguishers. J. Cryptographic Engineering **1**(2) (2011) 145–160
44. Whitnall, C., Oswald, E.: A Comprehensive Evaluation of Mutual Information Analysis Using a Fair Evaluation Framework. In Rogaway, P., ed.: CRYPTO. Volume 6841 of Lecture Notes in Computer Science., Springer (2011) 316–334
45. Guilley, S., Heuser, A., Rioul, O.: A Key to Success - Success Exponents for Side-Channel Distinguishers. In Biryukov, A., Goyal, V., eds.: Progress in Cryptology - INDOCRYPT 2015 - 16th International Conference on Cryptology in India, Bangalore, India, December 6-9, 2015, Proceedings. Volume 9462 of Lecture Notes in Computer Science., Springer (2015) 270–290
46. Easter, R.J.: Text for ISO/IEC 1st WD 17825 – Information technology – Security techniques – Non-invasive attack mitigation test metrics for cryptographic modules (January 19 2012) Prepared within ISO/IEC JTC 1/SC 27/WG 3. (Online).
47. ISO/IEC JTC 1/SC 27 IT Security techniques: ISO/IEC 17825:2016 Information technology – Security techniques – Testing methods for the mitigation of non-invasive attack classes against cryptographic modules (January 2016) https://www.iso.org/standard/60612.html.

48. ISO/IEC JTC 1/SC 27 IT Security techniques: ISO/IEC 15408-1:2009 Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model (January 2014) https://www.iso.org/standard/50341.html.
49. Common Criteria: Supporting Document Mandatory Technical Document Application of Attack Potential to Smartcards (2013) https://www.commoncriteriaportal.org/files/supdocs/CCDB-2013-05-002.pdf.
50. Heuser, A., Kasper, M., Schindler, W., Stöttinger, M.: A New Difference Method for Side-Channel Analysis with High-Dimensional Leakage Models. In Dunkelman, O., ed.: CT-RSA. Volume 7178 of Lecture Notes in Computer Science., Springer (2012) 365–382
51. Cao, Y., Zhou, Y., Yu, Z.: On the negative effects of trend noise and its applications in side-channel cryptanalysis. Chinese J. Electron. **23** (2014) 366–370
52. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems **2**(4) (Dec 1989) 303–314
53. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5) (July 1989) 359–366
54. Leshno, M., Schocken, S.: Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Networks **6** (1993) 861–867
55. Barron, A.R.: Universal approximation bounds for superpositions of a sigmoidal function. IEEE Transactions on Information Theory **39**(3) (May 1993) 930–945
56. Yarotsky, D.: Universal approximations of invariant maps by neural networks. CoRR **abs/1804.10306** (2018)
57. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining Smart-Card Security under the Threat of Power Analysis Attacks. IEEE Trans. Computers **51**(5) (2002) 541–552
58. Feldman, M.B.: A proof of lusin's theorem. The American Mathematical Monthly **88**(3) (1981) 191–192
59. TELECOM ParisTech SEN research group: DPA Contest (4^{th} edition) (2013–2014) http://www.DPAcontest.org/v4/.
60. Pu, S., Yu, Y., Wang, W., Guo, Z., Liu, J., Gu, D., Wang, L., Gan, J.: Trace Augmentation: What Can Be Done Even Before Preprocessing in a Profiled SCA? In Eisenbarth, T., Teglia, Y., eds.: Smart Card Research and Advanced Applications, Cham, Springer International Publishing (2018) 232–247