DL-LA: Deep Learning Leakage Assessment
A modern roadmap for SCA evaluations

Felix Wegener*, Thorben Moos*, and Amir Moradi

Ruhr University Bochum, Horst Görtz Institute for IT Security, Germany
firstname.lastname@rub.de
* These authors contributed equally to this work

Abstract. In recent years, deep learning has become an attractive ingredient to side-channel analysis (SCA) due to its potential to improve the success probability or enhance the performance of certain frequently executed tasks. One task that is commonly assisted by machine learning techniques is the profiling of a device's leakage behavior in order to carry out a template attack. Very recently at CHES 2019, deep learning has also been applied to non-profiled scenarios, extending its reach within SCA beyond template attacks for the first time. The proposed method, called DDLA, has some tempting advantages over traditional SCA due to merits inherited from (convolutional) neural networks. Most notably, it greatly reduces the need for pre-processing steps when the SCA traces are misaligned or when the leakage is of a multivariate nature. However, similar to traditional attack scenarios the success of this approach highly depends on the correct choice of a leakage model and the intermediate value to target.

In this work we explore whether deep learning can similarly be used as an instrument to advance another crucial (non-profiled) discipline of SCA which is inherently independent of leakage models and targeted intermediates, namely leakage assessment. In fact, given the simple classification-based nature of common leakage assessment techniques, in particular distinguishing two groups fixed-vs-random or fixed-vs-fixed, it comes as a surprise that machine learning has not been brought into this context, yet. Our contribution is the development of a full leakage assessment methodology based on deep learning which gives the evaluator the freedom to not worry about location, alignment and statistical order of the leakages and that easily covers multivariate and horizontal patterns as well. We test our approach against a number of case studies based on FPGA measurements of the PRESENT block cipher, equipped with state-of-the-art hardware-based countermeasures. Our results clearly show that the proposed methodology and network structure (which remains unchanged between the experiments) outperform the classical detection approaches (t-test and χ^2-test) in all considered scenarios.

1 Introduction

In an ideal world, side-channel security evaluations would be able to provide a qualitative and confident answer (pass or fail) to the question whether the device under test (DUT) is vulnerable to physical attacks or not. However, history has shown that this expectation is indeed a utopia. An exhaustive verification of the security of a DUT against all possible attack vectors is simply infeasible. Instead, the concept of leakage assessment has been introduced in order to answer a slightly, but explicitly, less informative question, namely whether in general any kind of information can be extracted from side-channel measurements of the device under test. Clearly, in case this question is answered positively, no conclusions about the actual vulnerability of the device with respect to key recovery attacks can be drawn (although it is sometimes interpreted as an indication thereof). Yet, in case it is answered negatively (and no *false* negative occurs) the DUT should be sufficiently secure. In other words, leakage assessment is conceptually capable of providing the initially desired confidence in at least one of the two cases. This possibility inspired the quest for appropriate leakage assessment methods in academia and industry.

The most prominent approach is certainly distinguishing two groups of measurements, one for fixed and one for random inputs, by means of the Welch's t-test [5, 14]. Whenever these two groups are distinguishable with confidence one can conclude that the device reveals input-dependent information. However, this method has some severe limitations, especially when more sophisticated types of side-channel leakage need to be captured. First of all, since each point in time is evaluated independently, the approach inherently expects that any potential side-channel leakage is of a univariate nature and, more generally, that the detection of the leakage does not benefit from a combination of multiple points. Yet, many counterexamples to this assumption can be observed in reality. Although Schneider *et al.* [14] provide detailed information on how to perform the t-test at arbitrary order and variate, the performance of the test at higher variates either quickly runs into feasibility issues or its success depends highly on the expertise of the evaluator and the prior knowledge about the underlying implementation. On another note, a misalignment of the leaking samples between the individual traces leads to a significantly impaired detection as well. Thus, the Welch's t-test, as it is currently applied as a test vector leakage assessment (TVLA) methodology, is naturally unsuited to cover multivariate and horizontal leakages, as well as (heavily) misaligned traces. In addition to that, it was recently pointed out that the separation of statistical orders, which is often seen as a beneficial feature of the t-test when seeking the smallest key dependent moment for example, causes false negatives when masked implementations with (very) low noise levels are analyzed or when the leakage is distributed over multiple statistical moments (as it is common for hardware masking schemes like threshold implementations) [11, 16]. Moradi *et al.* [11] suggested Pearson's χ^2-test as a natural complement to the Welch's t-test to aggregate leakages distributed over multiple orders and to analyze the joint information. By combining the two approaches the risk of false negatives, especially in the previously described cases, can significantly be reduced. Yet, in the same manner as the t-test, the χ^2-test analyzes the individual points in a leakage trace independently and therefore suffers from the same shortcomings when it comes to multivariate or horizontal patterns and misalignments.

Deep learning has been brought into the context of side-channel analysis mainly in order to improve the effectiveness of template attacks [6]. In a template attack the adversary is in possession of a fully-controlled profiling device, learns the leakage function of a certain cryptographic operation and subsequently uses the acquired knowledge to reveal sensitive information on a structurally identical but distinct target device where the secrets are unknown. Apart from the general suitability of deep learning to build classifiers for profiled side-channel attacks, it has also been demonstrated that certain features and structures of the applied neural networks offer valuable advantages over classical template attacks. For example, it is shown in [3] that convolutional neural networks (CNNs) can lead to efficient classifiers even when the available side-channel traces suffer from a misalignment. Thus, due to their so-called translation invariance property, CNNs can be utilized to conquer jitter-based countermeasures. Very recently, the first non-profiled deep learning based side-channel attacks were demonstrated in literature [17]. The proposed method, called DDLA, is based on guessing a part of the key, using it to compute the targeted key-dependent intermediate value, applying a leakage model and labeling the training data according to its result. Assuming that under the correct key hypothesis the differences between the classes implied by the leakage model correlate with the measured leakage traces (and for the incorrect guesses they do not), the impact of the correct key guess is visible in the training loss and the training accuracy respectively and can easily be identified. Although, this approach depends on the correct choice of the targeted intermediate value and the applied leakage model just as much as traditional attacks do, it offers some tempting advantages. First of all, in case CNNs are used, the translation invariance property allows to analyze misaligned traces without any pre-processing. Secondly, when the leakage is of a multivariate nature or generally distributed over multiple points no recombination and no prior knowledge about the underlying implementation is required. Hence, deep learning is a powerful tool for non-profiled scenarios as well.

1.1 Our Contribution

For the first time in literature we evaluate whether deep learning is an eligible strategy for black box leakage detection. To this end, we have developed an approach that is based on the concept of supervised learning. We call it deep learning leakage assessment (DL-LA) in the following. Simply put, we train a neural network with a randomly interleaved sequence of labeled side-channel measurements that have been acquired while supplying the DUT with one of two distinct fixed inputs (fixed-vs-fixed). Afterwards, in the validation phase, the trained network is supplied with unlabeled measurements from both groups and supposed to correctly classify them. Of course, the training set and the validation set are disjoint. In case the network succeeds with a higher percentage of correct classifications than could be achieved by a randomly guessing binary classifier with a non-negligible probability, it can be concluded that indeed enough information was included in the training set to distinguish the two groups. In other words, given the percentage of correctly classified traces and the size of the validation set one can easily calculate a confidence value, i.e., a probability, that the correct classifications were not just a random statistical occurrence. In this way it is possible to directly compare the confidence values achieved by DL-LA with the confidence provided by classical leakage assessment approaches, such as the Welch's t-test and Pearson's χ^2-test.

Clearly, in order to qualify as a reasonable strategy for leakage assessment, it should be given that the network which is trained to become the classifier offers a fairly robust and universal performance, independent of the type of side-channel leakage to be detected and independent of exterior parameters such as the trace length. The evaluator should not be required to tune the hyper parameters of the network, since this easily becomes computationally expensive, especially when the result of the evaluation is not known a priori. Thus, one of the main concerns is whether it is possible to select a network structure that is performing robustly when faced with different types of side-channel leakage and characteristics of the traces. Interestingly, we found that a very simple network structure consisting of only a few fully-connected layers with a fixed number of neurons handles this task surprisingly well. We test the chosen network in a total of 7 FPGA-based case studies featuring the PRESENT ultra-lightweight block cipher with different kinds of countermeasures applied. The classification capability of our network does not only withstand misaligned and noisy traces, but is able to deal with univariate and multivariate higher-order leakage as well. In all 7 case studies we compare the success of our method to both the Welch's t-test and Pearson's χ^2-test and show that DL-LA outperforms the leakage assessment capabilities of the classical techniques in all considered scenarios (either by requiring less traces to achieve the same confidence or by providing a higher confidence for the same amount of traces). We also present one scenario where both the univariate and the multivariate versions of the t-test and the χ^2-test fail to detect leaked information with confidence, while DL-LA still succeeds with only half of the available traces. As an unintended byproduct of our practical case studies, we provide the most detailed practical comparison between the Welch's t-test and Pearson's χ^2-test that has been reported in the literature so far.

The most outstanding advantage of our approach is clearly that the network is free to combine as many points for the classification of the two groups as necessary. Thus, even in complex scenarios of purely multivariate or horizontal leakages, the traces can simply be fed as training data into the network without any pre-processing or manual selection of points. Accordingly, neither a high expertise is demanded from the evaluator, nor is it required to obtain any prior information about the underlying implementation or the type of leakage that is expected. Additionally, DL-LA entails a much lower risk of false positives than current approaches, as it provides a single confidence value to assess the dinstinguishability of the groups. The traditional point-wise methods would actually need to normalize their confidence values to the number of points in the traces to provide a correct confidence threshold. However, this inaccuracy is mostly disregarded in their respective methodologies. Even though, DL-LA provides only a single confidence value, the approach can still identify the points of interest in side-channel traces that contain leakage, by performing a Sensitivity Analysis (SA) on the trained network. Obviously, the average computation time to perform a DL-LA is significantly higher when compared to simple univariate tests. However, as soon as more complex types of side-channel leakage need to be analyzed, the additional run time quickly pays off, since the effort that otherwise has to be spend in order to make traditional methods recognize those complex patterns (if even possible) grows even bigger and contains several steps that are hard to automate.

1.2 Claims and Non-Claims

In order to avoid any potential confusion regarding our claims, or lack thereof, we explicitly list the most important statements below:

We *do not* claim that ...

- our chosen network is optimal for leakage detection in general or for any of the considered case studies in particular. We are certain that there is room

for improvement, as we intentionally optimized for robustness and simplicity instead of single case performance.
- our chosen network necessarily leads to a classifier that outperforms the t-test or the χ^2-test for any given side-channel traces.
- DL-LA should replace all other leakage detection techniques, but rather that it can be extremely helpful especially in cases where detection is non-trivial.
- DL-LA generally causes none or fewer false negatives than the classical approaches.

We *do* claim that ...

- the chosen network offers some basic universality and robustness. We have tested the network even beyond the case studies presented in this work (and also for extreme cases like a trace length of 1 or greater than 10 000) and it delivered reliable results in all of them.
- the chosen network is able to learn first-order, higher-order, univariate, multivariate and horizontal leakages without requiring any pre-processing or prior knowledge of the underlying implementation.
- DL-LA entails a much lower risk of false positives (when the same confidence threshold is chosen) since it provides one confidence per set of traces instead of one confidence per time sample in the trace set.

2 Background

In this section we introduce the necessary background with respect to the roots and the state-of-the-art of leakage assessment, as well as deep learning and its applications to side-channel analysis.

2.1 Leakage Assessment

Ever since the introduction of side-channel attacks in 1999 [8] the standard approach for assessing the physical vulnerability of a device has been a more or less exhaustive verification of its resistance against known attacks while attempting to cover a broad range of intermediate values and hypothetical leakage models. This approach, however, became less feasible over the years due to the increasing amount of new attack methods and the higher complexity of potential leakage models due to the introduction of countermeasures against physical attacks. Another concern regarding this procedure is that it entails a significant risk of reporting physical security in favor of the DUT while in reality merely a certain attack vector was missed in the process (by mistake or because it was unknown at time of evaluation) that could indeed enable key recovery [14]. Hence, the need for a robust and reliable standard leakage assessment method independent of concrete attack scenarios, targeted intermediates and hypothetical leakage models grew consistently over the years. In an attempt to gather and evaluate promising candidates, the National Institute of Standards and Technology (NIST) hosted a "Non-Invasive Attack Testing Workshop" in 2011. One of the most intriguing proposals at the workshop was the use of the non-specific Welch's t-test [5] for leakage detection. Leakage detection avoids any dependency on the choice of intermediates and leakage models by focusing on the detection of leakage only, without paying any attention to the possibility to exploit said leakage for key recovery. Simply put, the concept is based on supplying the device under test with different inputs, recording its leakage behavior and evaluating whether a difference can be observed. Thus, such a method is suitable for black box scenarios and allows certification of a device's physical security by third party evaluation labs without the need to test a multitude of different methods and parameter combinations. Seven years later, after some shortcomings of the moment-based nature of the t-test had been identified [16], another popular statistical hypothesis test was proposed for leakage detection purposes, namely the Pearson's χ^2-test [11]. Both hypothesis tests, the t-test and the χ^2-test, are applied in the field of statistics in order to answer the question whether two sets of data are significantly different from each other. To be more precise, the evaluation of the tests examines the validity of the null hypothesis, which constitutes that both sets of data were drawn from the same population (i.e., they are indistinguishable) [14]. In side-channel analysis contexts, it is usually evaluated whether two groups of measurements can be distinguished with confidence. Traditionally, those two groups are acquired by supplying the DUT either with random (group Q_0) or a fixed input (group Q_1), selected by coin toss. Later, it has been demonstrated that the careful choice of two distinct fixed inputs (instead of maintaining one group for random inputs) usually leads to a lower data complexity for the distinction [4]. We provide the details on how to conduct the Welch's t-test and Pearson's χ^2-test below.

Welch's t-test. We denote two sets of data by Q_0 and Q_1, their cardinality by n_0 and n_1, their respective means by μ_0 and μ_1 and their standard deviations by s_0 and s_1. The t-statistics and the degrees of freedom v can then be computed using the following formulas.

$$t = \frac{\mu_0 - \mu_1}{\sqrt{\frac{s_0^2}{n_0} + \frac{s_1^2}{n_1}}} \qquad v = \frac{\left(\frac{s_0^2}{n_0} + \frac{s_1^2}{n_1}\right)^2}{\frac{\left(\frac{s_0^2}{n_0}\right)^2}{n_0 - 1} + \frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1}}$$

Afterwards, the confidence p to accept the null hypothesis can be estimated via the Student's t probability density function, where $\Gamma(.)$ denotes the gamma function [14, 11].

$$p = 2 \int_{|t|}^{\infty} f(t, v) dt \qquad f(t, v) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{\pi v}\,\Gamma\left(\frac{v}{2}\right)} \left(1 + \frac{t^2}{v}\right)^{-\frac{v+1}{2}}$$

In practice, for the sake of simplicity, it is common to only evaluate the t-statistics and to set the confidence threshold for distinguishability to $|t| > 4.5$. The statistical background of this threshold is that for $|t| > 4.5$ and $v > 1000$ the confidence p to accept the null hypothesis is smaller than 0.00001 which is equivalent to a 99.999 % confidence that the two sets were *not* drawn from the same population. Of course, when the degrees of freedom v is not explicitly evaluated, it can occur that the assumption $v > 1000$ does not hold. However, practice has shown that this procedure rarely produces false positive results in side-channel analysis contexts. Yet, calculating the actual confidence p is certainly preferable, scientifically correct and can still be efficiently implemented [11]. Since the Welch's t-test is designed to distinguish the means of two distributions, it can only be applied to first-order univariate analyses in its simplest form. Schneider *et al.* [14] extended the methodology to arbitrary orders and variates and provide the required formulas for incremental one-pass computation of all moments.

Pearson's χ^2-test. In order to mitigate some of the limitations and shortcomings of the moment-based nature of the Welch's t-test, in particular for higher-order analyses of masked implementations, Moradi *et al.* [11] suggested the Pearson's χ^2-test. In contrast to the t-test this hypothesis test analyzes the full distributions and can capture information that lies is multiple statistical moments. Thus, it prevents false negatives when moment-based analyses become suboptimal [11]. In a first step a contingency table F has to be constructed from the two sets Q_0 and Q_1 (basically two histograms). We denote the number of rows by r (= 2, when two sets are compared) and the number of columns by c (number of bins of the histograms). The χ^2-statistics x and the degrees of freedom v can then be computed using the following formulas.

$$\mathrm{x} = \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} \frac{(F_{i,j} - E_{i,j})^2}{E_{i,j}} \qquad \mathrm{v} = (r-1) \cdot (c-1)$$

$E_{i,j}$ denotes the expected frequency for a given cell.

$$E_{i,j} = \frac{\left(\sum_{k=0}^{c-1} F_{i,k}\right) \cdot \left(\sum_{k=0}^{r-1} F_{k,j}\right)}{N}$$

Finally, the confidence p to accept the null hypothesis is estimated via the χ^2 probability density function, where $\Gamma(.)$ denotes the gamma function [11].

$$p = \int_{\mathrm{x}}^{\infty} f(x, \mathrm{v}) dx \qquad f(x, \mathrm{v}) = \begin{cases} \frac{x^{\frac{\mathrm{v}}{2}-1} e^{-\frac{x}{2}}}{2^{\frac{\mathrm{v}}{2}} \Gamma\left(\frac{\mathrm{v}}{2}\right)} & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

In contrast to the t-test this procedure can easily be extended to more than two sets of data ($r > 2$), which can be a valuable feature when used as distinguisher for key recovery attacks. Generally, it can be said that in cases where the χ^2-test provides a higher confidence to reject the null hypothesis than the t-test (on the same side-channel data), the analysis of the leakages requires some special attention. This is usually the case when masked implementations with low noise levels are analyzed or when hardware-masking schemes like threshold implementations cause leakages in multiple moments due to physical defaults such as glitches [11].

2.2 Deep Learning

We give a brief summary of the history and applications of deep learning and subsequently introduce definitions and explain the underlying principle.

History and Applications. Historically, the field of machine learning dealt with extracting meaningful information from data by applying relatively simple mathematical models, e.g., Bayes Classifiers, Support Vector Machines or Decision Trees to a sanitized version of the input data. This required manual and time-consuming *feature engineering* to predetermine which elements might be useful in a given set of raw data and how to best represent them, e.g., Canny edge detection as a first hard-coded step for image classification.

In contrast, deep learning methods are generally capable of learning from raw input data, thereby making the elaborate modeling process unnecessary. Since the breakthrough improvement of classification accuracy on the ImageNet data set in 2012 [9], deep learning has been successfully applied to many diverse tasks such as speech recognition, drug discovery, natural language processing, visual art style transfer, image classification, autonomous driving and strategy games.

More recently, the side channel community discovered deep learning as a tool to perform profiled attacks [3, 6, 10] with competitive results compared to classical modeling techniques, e.g., based on a multivariate normal distribution. Apart from our present work, only Timon at CHES 2019 has investigated the use of deep learning in the non-profiled case [17]. The author exploits the correlation between a correct key guess and a steep learning rate to enable key recovery. Unfortunately, the method is computationally intense as a separate model has to be trained for every key guess and its success is highly dependent on the correct labeling of the data which implies a suitable choice of leakage model and targeted intermediate, making it unsuitable as a starting point for efficient and confident leakage assessment.

Principle and Definitions. In the following we limit ourselves to sequential neural networks (without recurrent elements) used for the purpose of classification. The aim of this description is to give brief definitions for the standard terms in deep learning, while the explanation of principles is intentionally very shallow. A neural network is structured into multiple layers, each containing a matrix of learnable weights w that is linearly applied to its inputs x and a non-linear activation function applied to each coordinate of the result of this matrix multiplication. The output of this combined operation is taken as an input for the subsequent layer. Finally, the output layer of the neural network contains as many output coordinates as classes[1] (c) and uses softmax as an activation function

$$\text{softmax}(x_j) = \frac{e_j^x}{\sum_{i=1}^{c} e_i^x},$$

such that the sum over all outputs is always equal to one, thereby forming a probability distribution over the possible class labels.

Let us first assume the weights are initialized with some values before an evaluation of the network takes place by applying the function of the first layer to the input sample and subsequently propagating the computed values forward layer by layer until all layers have been evaluated. The prediction y' consists in the output coordinate of the final layer with the highest value.

In the beginning, the weights in a neural network are initialized with random values. To determine useful weights that achieve accurate prediction values, a *training phase* is necessary. First, the designer needs to define a metric to measure the distance between a prediction y' and the actual class label y. This metric is called a *loss function* which determines the *loss score*. To perform training, a data set with labeled inputs, i.e., a list of tuples (x, y), is separated into *batches* of a fixed size b. The neural network is evaluated simultaneously on all samples in a batch thereby producing loss scores. After each batch an optimization strategy based on *Backpropagation* is used to adjust all weights in the neural network dependent on the gradient of the loss function. Each iteration through the entire training set is called an *epoch*. To minimize the loss score, training over multiple epochs is performed in each of which the training data is randomly regrouped into new batches. For simplicity we assume that the training ends after a predetermined number of user-defined epochs.

[1] We limit ourselves to this variant called one-hot encoding.

To judge the quality of the classifier during and after training, the metrics *accuracy* and *validation accuracy* should be considered. While accuracy is related only to the training set, validation accuracy takes an entirely separate *validation set* into account, to ensure that the traits learned are actually generalizable opposed to rote learning of the specific training set (the latter phenomenon is called *overfitting*).

When choosing and training a deep learning model, the designer has to determine values for many so-called *hyper parameters*[2], these include the depth of the network, the types and sizes of layers, their activation function, the loss function and the optimizer strategy. We provide the hyper parameters we chose and maintained throughout all of our case studies in Section 3.3.

3 DL-LA: Deep Learning Leakage Assessment

We introduce Deep Learning Leakage Assessment (DL-LA), a novel leakage assessment methodology based on deep learning. Our method is simple to apply and outperforms classical leakage detection approaches such as the Welch's t-test and the more recently proposed Pearson's χ^2-test in many cases due to its intrinsically multivariate nature.

3.1 Core Idea of DL-LA

The aim of leakage assessment is to determine whether an attacker is able to extract information from side channel measurements. The current state of the art for non-profiled adversary models is based on univariate statistical distinction tests (Welch's t-test, Pearson's χ^2-test) which are applied to two groups of side-channel measurements collected for two distinct fixed inputs processed by the target implementation (alternatively one group for random inputs and the other one for fixed).

DL-LA maintains the basic idea of distinguishing two groups of side channel traces from each other (fixed-vs-fixed). Hence, from an evaluator's perspective the entire measurement setup and tool-chain can remain unchanged when adopting our methodology. We apply deep learning to the concept of leakage assessment by training a neural network to serve as a distinguisher between the two groups. This is done in a supervised-learning-based approach by applying labeled data from both groups to the network. This set of data is then called the training set. Afterwards, the classification capabilities of the network are evaluated on a distinct validation set of labeled measurements without revealing the true labels to the network. The success rate of the classification on the validation set quantifies the amount of information that could be extracted from the *training* set by the neural network in order to provide a better-than-random guess which of the two fixed inputs was processed by the target. In case this classification succeeds with a higher percentage than it could be achieved by randomly guessing with a non-negligible probability, it gives clear evidence for the fact that informative side-channel leakage is present. In this context we present a simple metric to determine an exact p-value that quantifies the statistical confidence in the evidence.

Please note that the number of required traces to extract the information is only related to the training set. The size of the validation set can be chosen

[2] In distinction from the parameters, i.e., the concrete weights learned during training.

completely independent and influences the result of the detection only if generalizable features (i.e., side-channel leakage) could be extracted from the training set. Otherwise the percentage of correct classifications will never be significantly different from 50%, no matter how large the validation set is. We discuss the partition strategy into training set and validation set to decouple the number of traces available to the attacker from the statistical confidence the evaluator wants to obtain in Section 5.

Given an identical number of traces, our proposed DL-LA leads in many cases to a higher statistical confidence than the t-test and χ^2-test and is less prone to produce false positives. In the case of an implementation that does not show univariate, but only multivariate leakage DL-LA outperforms current leakage assessment methods by a large margin without requiring any knowledge about the underlying implementation.

3.2 Overall Methodology

We assume that the recorded traces have already been separated into a set of N training traces and a set of M validation traces, the latter of which should have an equal number of elements from both groups to maximize the statistical confidence value that can be obtained during the evaluation[3]. First, the evaluator has to pick a confidence level, i.e., an upper bound on the chance that a false positive occurs. We assume the common threshold in SCA evaluations of $p_{\text{th}} = 10^{-5}$. Now, let v be the validation accuracy obtained by the neural network, then the total number of correct classifications is computed as $s_M = v \cdot M$. Considering the null hypothesis $\mathcal{H}_0$ where the neural network did not learn anything and classifies randomly (coin flip model), this corresponds to modeling the total number of correct guesses as a random variable following a binomial distribution

$$\mathcal{H}_0 : X \sim Binom(M, 0.5).$$

The probability that at least s_M correct classifications occur in a purely random classifier is given by: $P(X \geq s_M)$. This probability is easily computed as

$$P(X \geq s_M) = \sum_{k=s_M}^{M} \binom{M}{k} 0.5^k 0.5^{M-k} = 0.5^M \sum_{k=s_M}^{M} \binom{M}{k}$$

Now, we say that the implementation leaks information about intermediate values if

$$P(X \geq s_M) < p_{\text{th}}.$$

In this case the exact location of leakage can be determined subsequently by Sensitivity Analysis (cf. Section 3.4).

In the following, we specify a neural network that showed excellent and robust results during our case studies. Further, we introduce Sensitivity Analysis as a method of leakage location and preempt several common pitfalls during adoption.

[3] We provide a discussion on the size of both sets in Section 5.

3.3 Proposed Network Structure

To perform DL-LA we suggest a generalizable network architecture that is free of any assumptions about the traces to be analyzed or about the underlying implementation of the DUT. We performed all case studies in Section 4 with the same network built within the Python library Keras using TensorFlow as the backend.

Specification. Initially, we determine the (point-wise) mean μ and standard deviation σ of the traces in the training set thereby standardizing both training and validation sets:

$$X_i^j := (X_i^j - \mu_i)/\sigma_i,$$

with j denoting the trace and i the time sample within the trace. This very lightweight pre-processing step is necessary to reach a homogeneous range between all input points and weights thus enabling efficient training.

The network consists of four fully-connected layers (*Dense*) of 120, 90, 50, respectively 2 output neurons. The input layer and each of the inner layers use a *Rectified Linear Unit* (*ReLU*) as an activation function, while the final layer uses *softmax*. The four Dense layers are each separated by a *BatchNormalization* layer. In summary, the model can be defined in Python as:

```
model = Sequential([
   Dense(120, activation = 'relu', input_shape= (tracelength,) ),
   BatchNormalization(),
   Dense(90, activation = 'relu'),
   BatchNormalization(),
   Dense(50, activation = 'relu'),
   BatchNormalization(),
   Dense(2, activation = 'softmax') ])
```

Further, we used the *mean squared error* as a loss function and *adam* as an optimizer with the default parameters provided by Keras[4]. We chose the batch size according to the memory restrictions of our Tesla K80 graphics card with 11 GB VRAM as 2 000 samples for traces of length 5 000 points and 20 000 samples for traces of length 500 points.

Justification. We chose *ReLU* defined as

$$relu(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

as an activation function over other common possibilities, e.g., tanh or sigmoid, because of better results regarding validation accuracy in initial tests as well as for better computational performance when operating on large datasets (which is highly relevant for the evaluation of protected implementations). We chose the softmax activation function of the final layer to create a probability distribution over both classes as explained in Section 2.2. The purpose of each *BatchNormalization*-layer is to decouple the learning process of all Dense layers from each other and additionally provide a means of regularization to prevent overfitting [7].

[4] $lr = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, decay = 0.0$

We confirmed the suitability for univariate and multivariate leakage located in different statistical orders and for traces as short as one point and as long as 10 000 points which may be encountered during a typical leakage evaluation of symmetric cryptographic primitives and provide extensive depth on the performance of our leakage assessment approach in different case studies in Section 4.

3.4 Extracting Temporal Information

If leakage is detected, the hardware designer or evaluator is usually interested in exactly pinpointing the leakage locations to report or alleviate the shortcoming, e.g., masking flaws. By applying *Sensitivity Analysis (SA)* [15, 17] we can exactly locate all points of interest by quantifying how much they contributed to the leakage function learned by the neural network. In short, SA determines the partial derivates of one output coordinate of the neural network with respect to the network inputs, thereby characterizing the effect of a slight change in each individual input on the classification outcome.

We perform SA on the final network after training has completed by averaging the gradients of one output coordinate (with respect to the network inputs) weighted with the network inputs for all samples in the validation set and subsequently take the absolute value. More precisely, let x_i denote the i-th input of our network, y_0 the first output coordinate of the network and X_i^j the value of the i-th input for trace j in the validation set. Then the sensitivity can be determined as:

$$s_i = \left| \sum_j \frac{\partial y_0}{\partial x_i} \cdot X_i^j \right|.$$

While the actual value of this expression has to be determined via the chain-rule over all network layers, this process is fully-automated by TensorFlow such that the remaining effort for the evaluator is a single function call.

3.5 Common Pitfalls

We discuss the most important differences between the classical detection approaches and DL-LA and aim to preempt common pitfalls an evaluator might encounter with our leakage assessment:

Group Imbalance. While the classical TVLA based on the Welch's t-test as well as the χ^2-test can handle groups imbalanced in mean, variance and size, we want to stress that an equalization of group sizes in the validation set is extremely important for DL-LA. If the groups are imbalanced, the test statistic no longer follows the distribution $X \sim Binom(M, 0.5)$. Instead, always assigning the label of the more common group leads to a classifier which outperforms random guessing, without actually being able to distinguish the groups based on their traces. This discrepancy between actual and theoretical distribution of the test statistic – given a sufficiently large validation set – will lead to false positives. We strongly advise pruning both groups in the validation set to an exact ratio of 50/50.

Probability Adaption. An obvious idea to counteract the issue just addressed is the adaption of the success probability in the Binomial distribution. Assume a slight (or even significant) imbalance ϵ in group sizes

$$\frac{|G_0|}{|G_0| + |G_1|} = 0.5 + \epsilon.$$

The evaluator could simply adapt the distribution of the test statistic to

$$X \sim Binom(M, 0.5 + \epsilon).$$

While this might even show satisfactory results in the case of low noise and unprotected or severely flawed implementations, which in turn lead to a high validation accuracy, we caution against any alteration of the distribution. In all practically relevant cases (protected implementation, moderate noise) a change of the success probability severely lowers the confidence of the statistical test. More specifically, consider a validation set of size 500 000 traces over which a validation accuracy of $v = 0.506$ has been achieved. In case of a balanced validation set this event is highly statistically significant (10^{-17}). However, if the validation set contains a small bias of $\epsilon = 0.004$ no significance can be concluded as the remaining likelihood for this event is only 10^{-3}. It is obvious that the adapted test looses its statistical power in all interesting cases; hence, false negatives might occur. Therefore, we want to reinforce the previous point to prune the validation set to an exact 50/50 ratio.

Overfitting. We caution against using an overly complicated neural network as it might lead to overfitting, which is defined by a continuous rise of the training accuracy over the number of epochs while the validation accuracy begins to fall. The underlying cause of this effect is the memorization of the training set as opposed to learning generalizable features of the entire set. Hence, it can be prevented by using a network with a simple structure which does not contain excessively many weights and optionally includes Normalization, Regularization or Dropout layers (cf. Section 3.3).

4 Experimental Results

In the following we provide an experimental verification of the suitability of DL-LA as a black box leakage assessment strategy. In contrast to related works we do not test our approach under circumstances of purely theoretical relevance, such as noise-free simulations or Sbox-only measurements with an extremely high signal-to-noise ratio (SNR). Instead, we strive for a realistic benchmark of our approach with a clear real-world impact. For this reason we chose hardware implementations of the full PRESENT-80 ultra lightweight block cipher [2] as the common target in our case studies. PRESENT has been developed for ubiquitous and resource constrained computing environments, which exactly constitutes the type of application that commonly requires side-channel security as a design goal and may be certified by third-party evaluations labs. We target a very compact serialized hardware implementation of the cipher and two protected versions of it which both feature provable first-order security. One of them even provides security at any order against univariate-only attacks. In all scenarios, we compare the leakage assessment capability of DL-LA to the previously introduced state-of-the-art methods Welch's t-test and Pearson's χ^2-test. We conclude that DL-LA

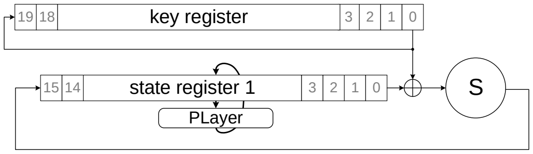

Fig. 1. Unprotected serialized PRESENT architecture with a 4-bit data path.

is able to confidently detect leakage with fewer traces or a higher confidence on the same amount of traces compared to the conventional methods which makes it an extremely valuable tool for leakage assessment. Especially in the case studies where only multivariate leakage is present DL-LA outperforms the state of the art by a large margin.

4.1 Measurement Setup

We have implemented the different instances of the PRESENT block cipher on a SAKURA-G board [1] which has specifically been designed for SCA evaluations. The board features two Spartan-6 FPGAs, one as a target and the other as a control interface. We measured the voltage drop over a 1 Ω shunt resistor in the V$_{dd}$ path of the target FPGA, which is amplified through the built-in AC amplifier, with a digital sampling oscilloscope at a sampling rate of 1 GS/s for the first 5 case studies and 100 MS/s for the last 2. The targets were clocked at a frequency of 6 MHz, with the exception of the clock-randomized case study which is detailed later. For all case studies we measured side-channel traces in a fixed-vs-fixed manner for two arbitrarily selected fixed inputs. We have taken care to follow all rules that have to be considered to avoid false positives in leakage assessment [14], e.g., the measurements of the two groups are randomly interleaved and in the masked cases the communication between the control and the target FPGA is performed in a shared manner (in our case the same holds for the communication with the measurement PC).

Case Study 1: Unprotected PRESENT, aligned Traces

In this first case study we target an unprotected serialized implementation of the PRESENT block cipher. The architecture can be seen in Figure 1 and is similar to profile 1 introduced in [13]. As a first step we evaluate the confidence to distinguish the two groups of measurements (fixed-vs-fixed) by conventional methods. The results of the first-order t-test and the χ^2-test can be seen in Figure 2. In both cases we plot the confidence values p instead of relying on the common (and less precise) approach of defining a threshold for the intermediate statistics (e.g., $|t| > 4.5$). The t-test succeeds in providing a confidence higher than 99.999 % for the distinguishability of the two groups after only 20 traces since it shows a probability below 10^{-5} to accept the null hypothesis. The χ^2-test requires approximately 90 traces to overcome the desired confidence threshold. In conclusion, none of the two methods faces any problems to distinguish the leakage distributions with a high confidence when 1 000 traces are considered.

When applying DL-LA to the same traces, the results in Figure 3 are achieved. We have to state here that a plot as depicted in Figure 3(b) is rather unnatural

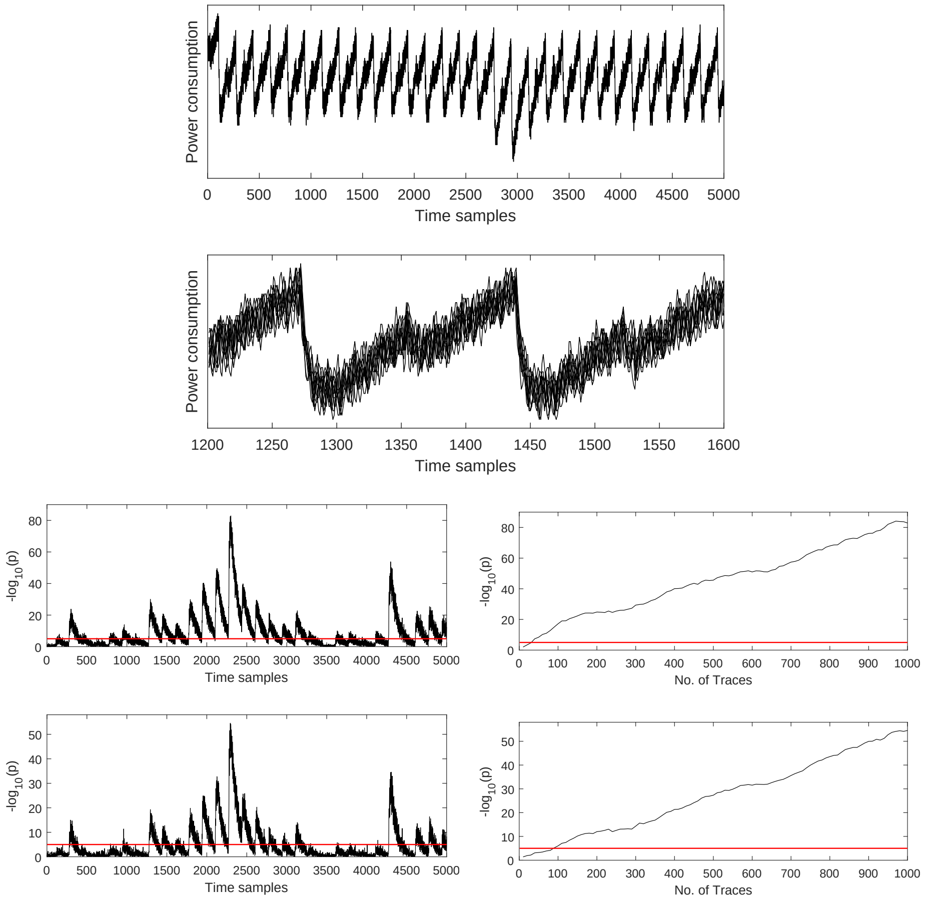

Fig. 2. Univariate leakage assessment using 1 000 traces (step size 10) of an unprotected serialized PRESENT-80 implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) t-test results, 4) χ^2-test results.

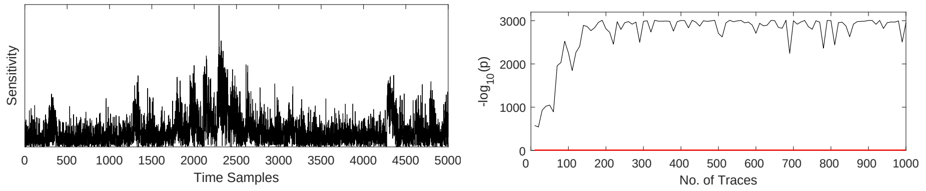

Fig. 3. DL-LA and Sensitivity Analysis using 1 000 traces (step size 10) of an unprotected serialized PRESENT-80 implementation. For each p value 30 epochs and a validation set of 10 000 traces are considered.

to obtain using DL-LA. Normally, training and validating the network results in a confidence value after each epoch. Thus, it would be more natural to train the network on a training set of fixed size and to show the p values over the number of epochs to determine how many are required to overcome the threshold. However, in order to offer the best possible comparison between the leakage assessment approaches we repeated this process 100 times for a fixed number of epochs (30) and a training set that increases by 10 traces per step and plotted the maximum confidence over the number of traces. The result shows that a network which is trained on only 10 traces is already capable of providing an extremely high confidence that the two groups are distinguishable (since large $-\log_{10}(p)$ values give confidence to reject the null hypothesis). By increasing the size of the training set the confidence is boosted significantly until the p values stagnate in a corridor between 10^{-2300} to 10^{-3011}. Please note that, as the validation set has a size of 10 000 traces, the maximum achievable p value is $0.5^{10\,000} = 10^{-3011}$. Thus, the stagnation in the corridor is simply caused by the fact that (almost) all of the traces in the validation set were classified correctly. By using a larger validation set the $-\log_{10}(p)$ values would rise even beyond 3011. We also perform a Sensitivity Analysis on the network to determine the points of interest and obtain a spatial resolution comparable to the univariate tests (cf. Figure 3(a)). The absolute values of the SA are not meaningful and cannot be compared to any threshold. Thus, they are omitted here. In summary, DL-LA outperforms the classical detection approaches in terms of required number of traces and absolute confidence provided. Of course, for the evaluation of DL-LA as performed in this case study, a validation set is required on top of the training set. However, please note that we only chose a validation set of 10 000 traces here in order to show the extremely high magnitude of achievable confidence values, even when considering very small training sets[5]. In fact, the indication of distinguishability relates only to the training set, and, in case the network learned generalizable features from it, the confidence can be arbitrarily boosted by increasing the validation set. If no generalizable features were learned (e.g., because no leakage is present) the percentage of correct classifications will not be significantly different from 0.5. The advantages of decoupling the confidence from the number of traces are discussed in Section 5.

Case Study 2: Unprotected PRESENT, misaligned Traces

This case study is an exact replication of the previous one apart from the fact that we artificially created a misalignment of the traces, as apparent in Figure 4(b). This misalignment was achieved by forcing the oscilloscope to trigger the acquisition of the power traces close to the peak of the rising edge of the trigger signal (in our case at 2.48 V while the peak is at 2.5 V) as opposed to the more stable part in the middle of the edge. Thus, due to the electronic noise, the acquisition is in some cases triggered earlier than in others and the traces are not perfectly aligned anymore. Figure 4 shows that the t- and χ^2-test results do not seem to significantly suffer from this misalignment when considering the absolute magnitude of the $-\log_{10}(p)$ values. However, the number of traces to overcome the threshold is increased in comparison to the previous case study in both tests. DL-LA also performs similar as before, as apparent from Figure 5 and outscores the classical detection approaches in required traces and provided confidence. It seems that the slight misalignment of the traces does not significantly affect the detection capabilities of any of the leakage assessment techniques when unprotected implementations are considered and the number of available traces is not chosen to be extremely small.

[5] The minimum size of the validation set in order to be able to overcome the detection threshold is 17, since $-\log_{10}(0.5^{17}) > 5$. However, this assumes a 100% correct classification by the network, otherwise a larger set needs to be considered.

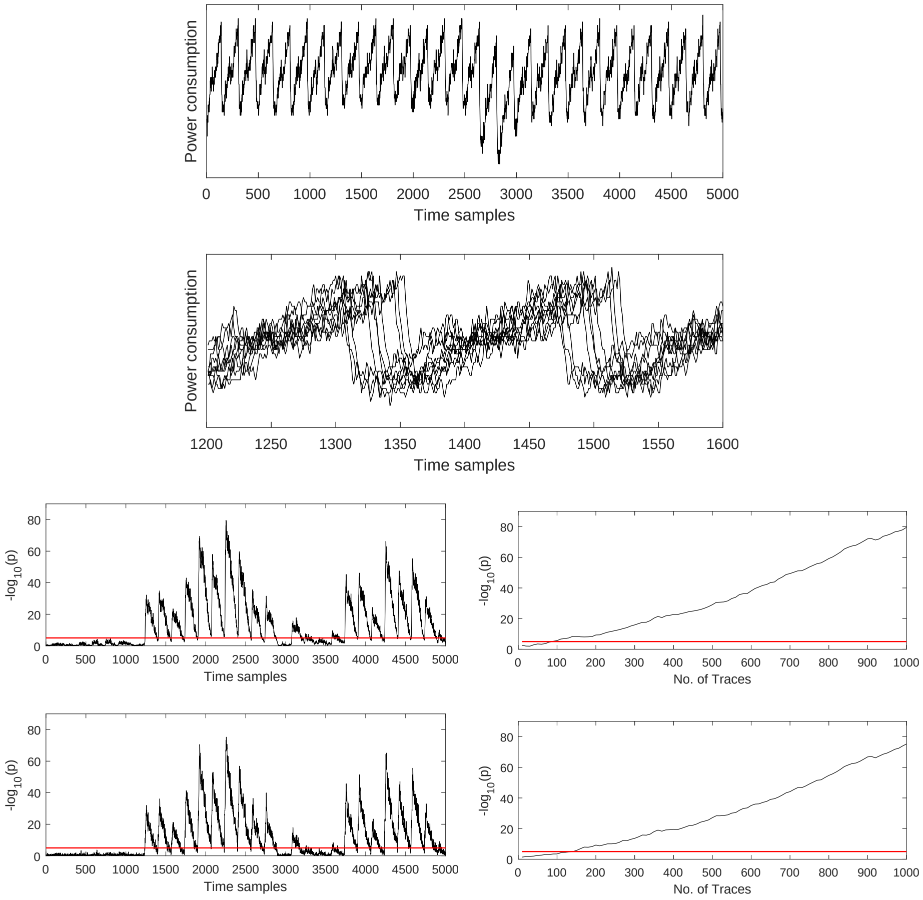

Fig. 4. Univariate leakage assessment using 1 000 misaligned traces (step size 10) of an unprotected serialized PRESENT-80 implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) t-test results, 4) χ^2-test results.

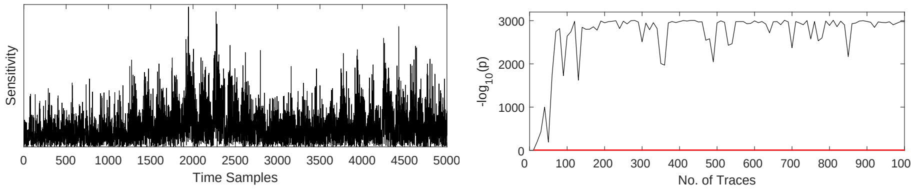

Fig. 5. DL-LA and Sensitivity Analysis using 1 000 misaligned traces (step size 10) of an unprotected serialized PRESENT implementation. For each p value 30 epochs and a validation set of 10 000 traces are considered.

Case Study 3: (Unprotected) PRESENT, randomized Clock

Since the artificial delay in the previous case study only slightly increased the data complexity of a leakage detection we now try to test a countermeasure that leads to much more heavily misaligned and noisy traces. In particular, we randomize the clock that drives the targeted PRESENT implementation. This is done by clocking the cipher with the output of a 64-bit LFSR. Hence, in each encryption (and therefore also in the power traces) the same intermediate computations are executed at different times, since the rising edges of the LFSR output occur in a random sequence. The input frequency of the LFSR was set to 24 MHz so that the number of rising edges in a certain frame of time is on average similar to being clocked by a stable 6 MHz clock. In this case the *t*- and χ^2-test struggle significantly more to detect leakage than in the previous experiments, as apparent in Figure 6. While the *t*-test requires about 2 000 traces for a detectable breach of side-channel security, the χ^2-test barely overcomes the threshold at all. DL-LA on the other hand is able to confidently state distinguishability after about 150 traces (cf. Figure 7). Although all three approaches suffer significantly from the misalignment and the added noise, DL-LA is still able to perform detection on a much smaller amount of traces. Please note that, if desired by the

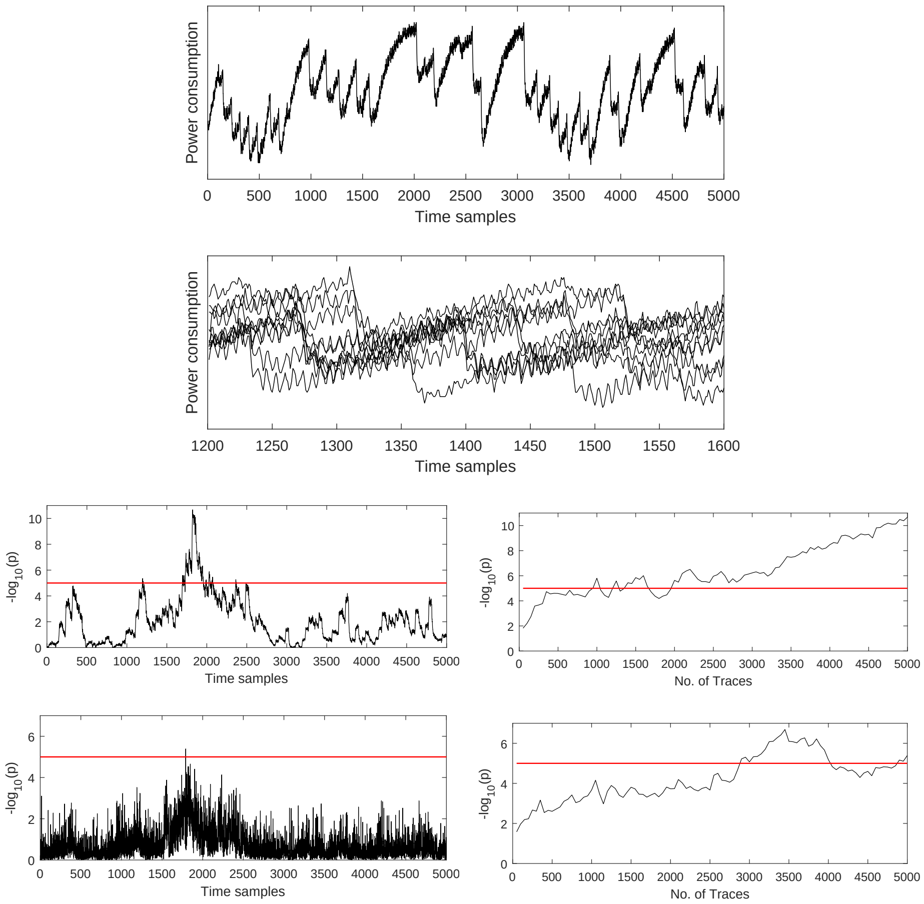

Fig. 6. Univariate leakage assessment using 5 000 traces (step size 50) of a serialized PRESENT-80 implementation with clock randomization. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) *t*-test results, 4) χ^2-test results.

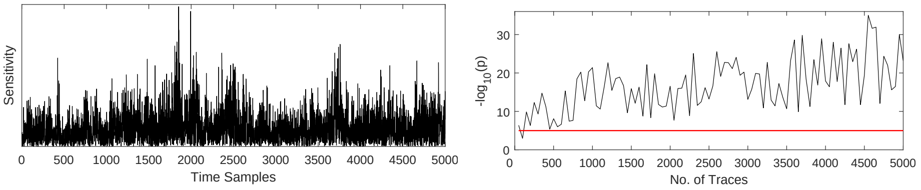

Fig. 7. DL-LA and Sensitivity Analysis using 5 000 misaligned traces (step size 50) of an unprotected serialized PRESENT implementation with clock randomization. For each p value 30 epochs and a validation set of 10 000 traces are considered.

evaluator, the confidence can be made arbitrarily larger by increasing the size of the validation set.

Case Study 4: PRESENT Threshold Implementation, aligned Traces

In this case study we target a serialized threshold implementation (TI) [12] of the PRESENT block cipher. The architecture can be seen in Figure 8 and is similar to profile 2 introduced in [13]. The PRESENT Sbox is decomposed into two quadratic functions F and G. Both of those decompositions are split into three component functions each according to the concepts of *correctness*, *non-completeness* and *uniformity* [12]. As apparent from Figure 8 the three shares in the computation of the decomposed Sbox are evaluated in parallel. Thus, no first-order, but univariate higher-order (especially second- and third-order) leakage is expected. We evaluate this assumption in Figure 9. As expected the first-order t-test does not indicate detectable leakage, but the second- and third-order test do. Interestingly, we can confirm the statements made by the authors of the χ^2-test proposal [11] regarding the shortcomings of the moment-based nature of the t-test. Unlike the situation in the previous case studies, the χ^2-test outperforms the t-test here. While the second-order and the third-order t-test require 3 000 000 and 1 500 000 traces for the detection respectively, the χ^2-test succeeds after only 500 000 traces and results in a much higher confidence over all.

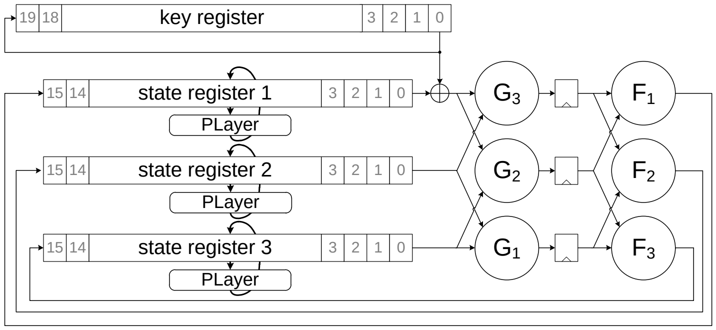

Fig. 8. Serialized PRESENT threshold implementation architecture with 3 shares and a decomposed Sbox.

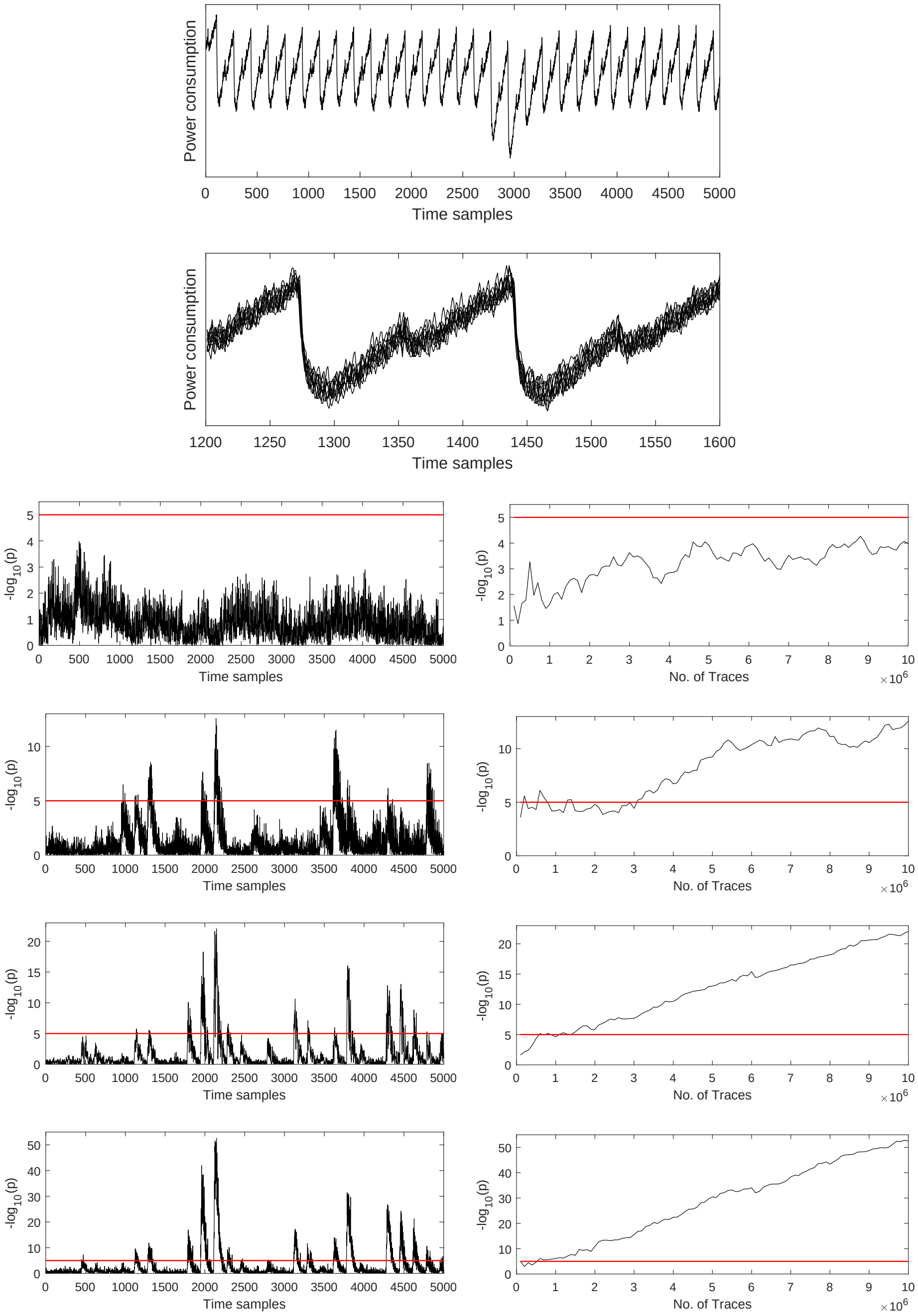

Fig. 9. Univariate leakage assessment using 10 000 000 traces (step size 100 000) of a serialized PRESENT threshold implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) first-order t-test results, 4) second-order t-test results, 5) third-order t-test results, 6) χ^2-test results.

Please note that for this case study and the upcoming ones, where we analyze protected implementations, we change the visualization of the DL-LA results.

Due to the large data sets involved it is not feasible to train many different classifiers with a steadily increasing size of the training set over many steps. Instead we visualize the $-\log_{10}(p)$ values over the number of epochs (instead of over training traces). We do this twice, once for the minimum number of traces required by the classical assessment method (here χ^2-test with 500 000) and once for a larger training set in order to show that much larger confidence values can be achieved in the protected cases as well. Those results are presented in Figure 10. In case of a training set of size 500 000, the DL-LA succeeds only just in overcoming the confidence threshold. However, barring the possibility of a false positive results (which is highly unlikely), the confidence could be increased by an evaluator either by considering more epochs or by increasing the validation set. In the case of a training set including 3 000 000 measurements, the confidence that side-channel leakage is present becomes extremely large.

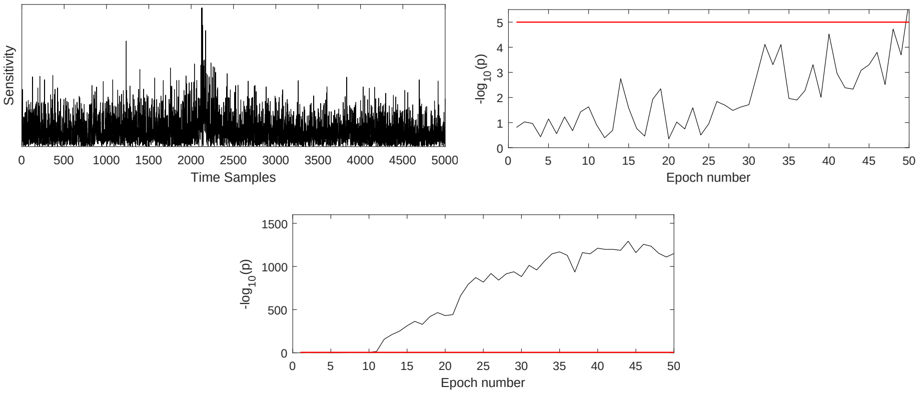

Fig. 10. DL-LA and Sensitivity Analysis using 500 000 (Fig. 10(b)) and 3 000 000 (Fig. 10(a), 10(c)) traces of a serialized PRESENT threshold implementation respectively. For each p value 50 epochs and a validation set of 1 500 000 traces are considered.

Case Study 5: PRESENT Threshold Implementation, misaligned Traces

This case study is equivalent to the previous one apart from the fact that we artificially created a misalignment of the traces, as it was already done for case study 2. As a result of this misalignment the leakage detection approaches require slightly more traces to overcome the confidence threshold than in the aligned case. In particular, as shown in Figure 11, the second-order and the third-order t-test require 3 600 000 and 1 500 000 traces for the detection respectively, while the χ^2-test succeeds after only 800 000 traces and again results in a much higher confidence. The DL-LA is again the most powerful leakage detection mechanism and succeeds for both sizes of the training set (800 000 and 3 000 000) with much higher confidence values than any of the classical approaches (cf. Figure 12).

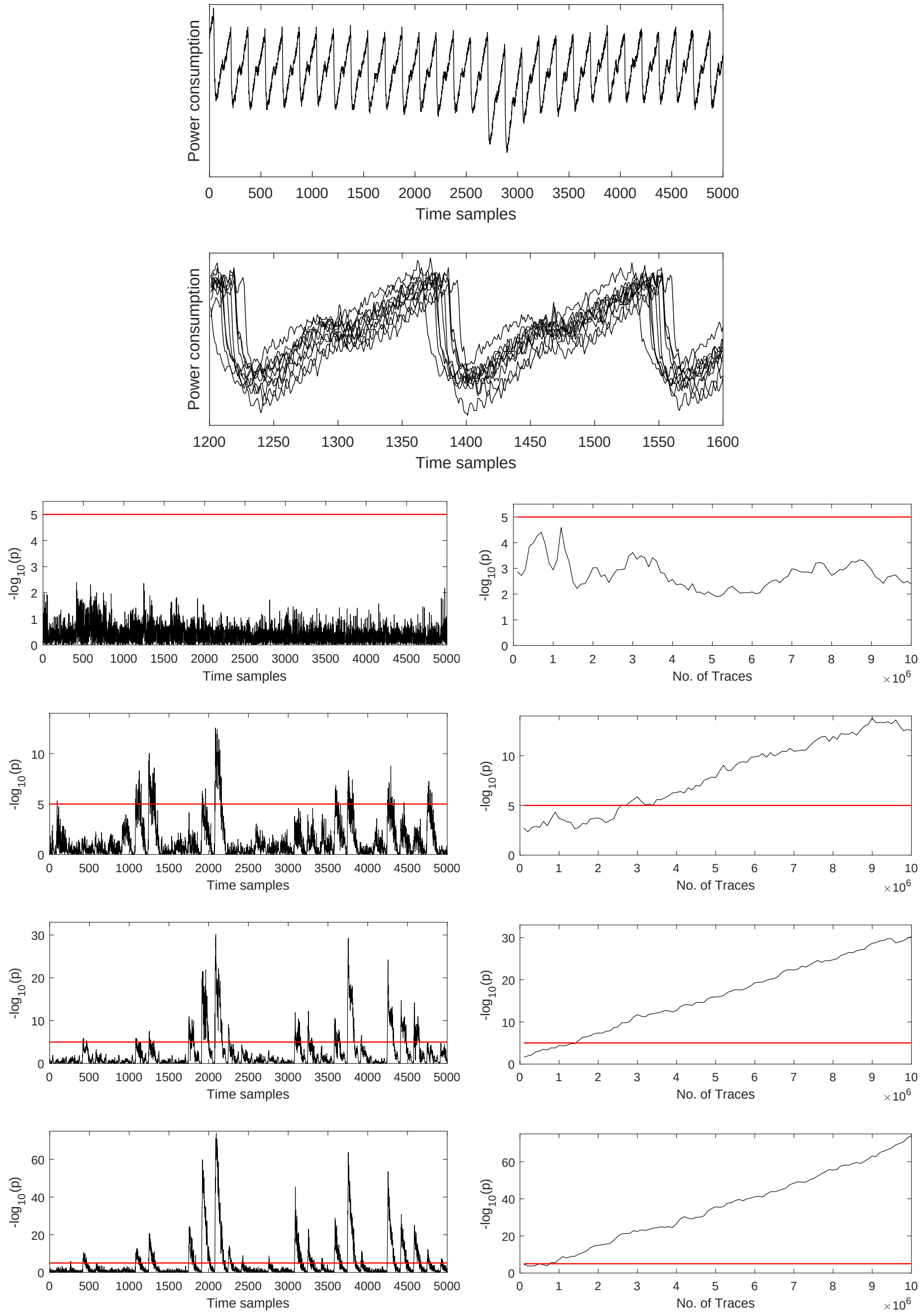

Fig. 11. Univariate leakage assessment using 10 000 000 misaligned traces (step size 100 000) of a serialized PRESENT threshold implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) first-order t-test results, 4) second-order t-test results, 5) third-order t-test results, 6) χ^2-test results.

Case Study 6: PRESENT Multivariate Threshold Implementation, aligned Traces

In the final two case studies we concentrate on scenarios where the classical univariate detection approaches are naturally unsuited to detect leakage, namely

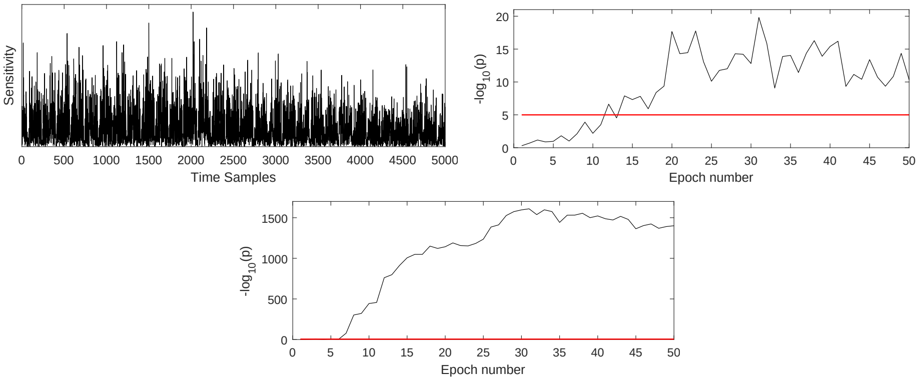

Fig. 12. DL-LA and Sensitivity Analysis using 800 000 (Fig. 10(b)) and 3 000 000 (Fig. 10(a), 10(c)) misaligned traces of a serialized PRESENT threshold implementation respectively. For each p value 50 epochs and a validation set of 1 500 000 traces are considered.

purely multivariate higher-order leakages. These cases are the primary motivation to apply DL-LA in reality, as all currently known methods fail to capture the whole amount of present side-channel leakage in these scenarios. We provide evidence for this statement in the following.

We constructed a special version of the PRESENT threshold implementation architecture depicted in Figure 8, that does not offer univariate side-channel leakage. To this end we had to ensure that all six component function (G_1, G_2, G_3, F_1, F_2 and F_3) are evaluated sequentially and not in parallel. We did this by gating their respective inputs with `AND` gates which are controlled by a finite state machine (FSM). In addition to that we had to make sure that none of the state registers are clocked at the same time. Thus a single Sbox computation takes 7 clock cycles in the resulting hardware design. As expected, our univariate leakage assessment using the classical detection approaches does not indicate the presence of any side-channel leakage (cf. Figure 13). However, a multivariate investigation could still find higher order leakage if performed at the correct offsets. This requires either white-box knowledge about the implementation or must be determined by exhausting all possibilities. The results for the best offset leading to detectable multivariate leakage is illustrated in Figure 14 which shows leakage in the third order after more than 45 million traces. Please note that we have performed each multivariate second-order t-test, third-order t-test and χ^2 with the correct offsets (as we know all the implementation details) and none of them was able to detect leakage with fewer traces.

In stark contrast, as apparent in Figure 15, DL-LA provides a very high confidence level of $-\log_{10}(p) > 150$ for the presence of side-channel leakage after training on only 20 million traces. We still retained the same network architecture as before and made absolutely no assumptions about the leakage and required no white-box knowledge about the offset of the individual evaluation of TI shares. Due to memory and time restrictions we pruned the trace to a length of 500 sample points (1281-1780). Validation took place on 5 million traces. Leakage becomes apparent after 25 epochs and continuously increases until our chosen threshold of 50 epochs has been reached.

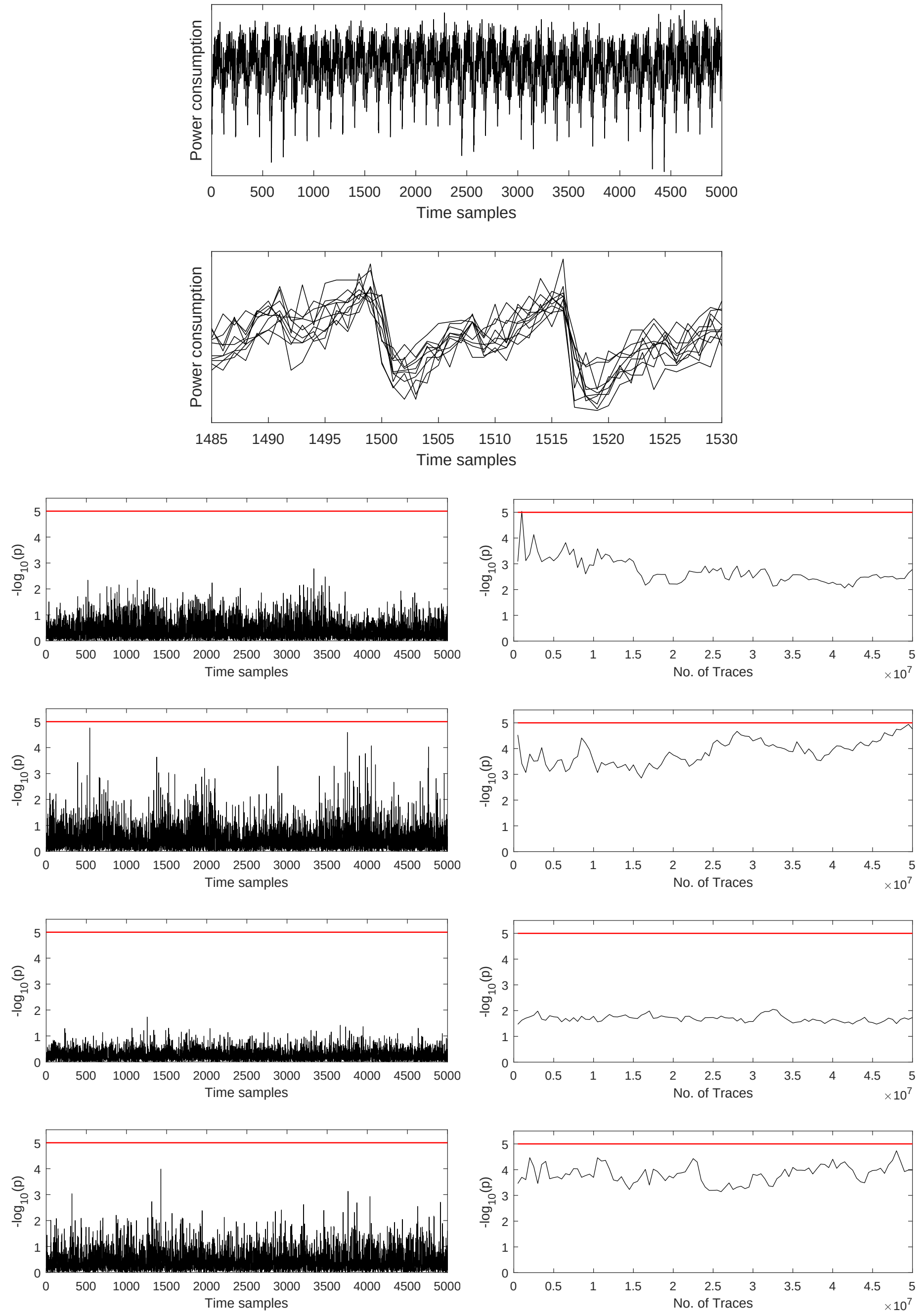

Fig. 13. Univariate leakage assessment using 50 000 000 traces (step size 500 000) of a serialized multivariate PRESENT threshold implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) first-order t-test results, 4) second-order t-test results, 5) third-order t-test results, 6) χ^2-test results.

Case Study 7: PRESENT Multivariate Threshold Implementation, misaligned Traces

Our final case study is a replication of the previous one, but again we misaligned the traces through bad triggering. As shown in Figure 16 no univariate detec-

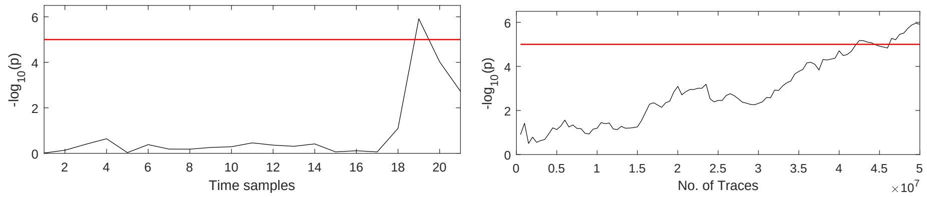

Fig. 14. Multivariate third-order t-test using 50 000 000 traces (step size 500 000) of a serialized multivariate PRESENT threshold implementation.

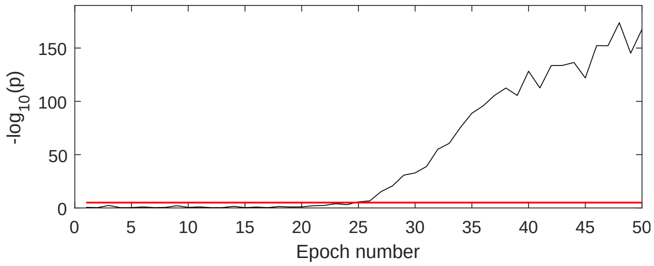

Fig. 15. DL-LA using 20 000 000 traces of a serialized multivariate PRESENT threshold implementation. For each p value 50 epochs and a validation set of 5 000 000 traces are considered.

tion of leakage succeeds. In this case however, even the multivariate third-order analysis with the best possible offset for leakage detection in the previous case study does not succeed (cf. Figure 17). In other words, the acquired set of traces does not allow detection of any leakage using conventional methods, at least in case the traces are not re-aligned before the analysis.

DL-LA however detects leakage with high confidence ($-\log_{10}(p) > 60$) after training on only half of the available traces (25 000 000). This result is depicted in Figure 18.

5 Discussion

False Positives. False positives commonly appear as a problem in classical leakage evaluations. This is due to their point-wise independent nature. A threshold of $p_{\text{th}} = 10^{-5}$ for each individual point will lead to an aggregation of the error probability over the length of the entire trace, thereby lowering the confidence. More formally, the likelihood that a false positive occurs at least once in a trace of length K can be described as:

$$P(\text{false positive}) = 1 - (1 - p_{\text{th}})^K.$$

For the typical value of $K = 5\,000$ in our case studies and the common threshold of $p_{\text{th}} = 10^{-5}$ this formula equates to 0.0488, thereby it only provides an effective confidence of roughly 5%. Hence, a manual investigation of the individual leakage points is often necessary when performing classical leakage detection to exclude false positives.

In contrast, our deep learning based methodology does not produce several individual univariate statistical tests, but produces a single decision metric based

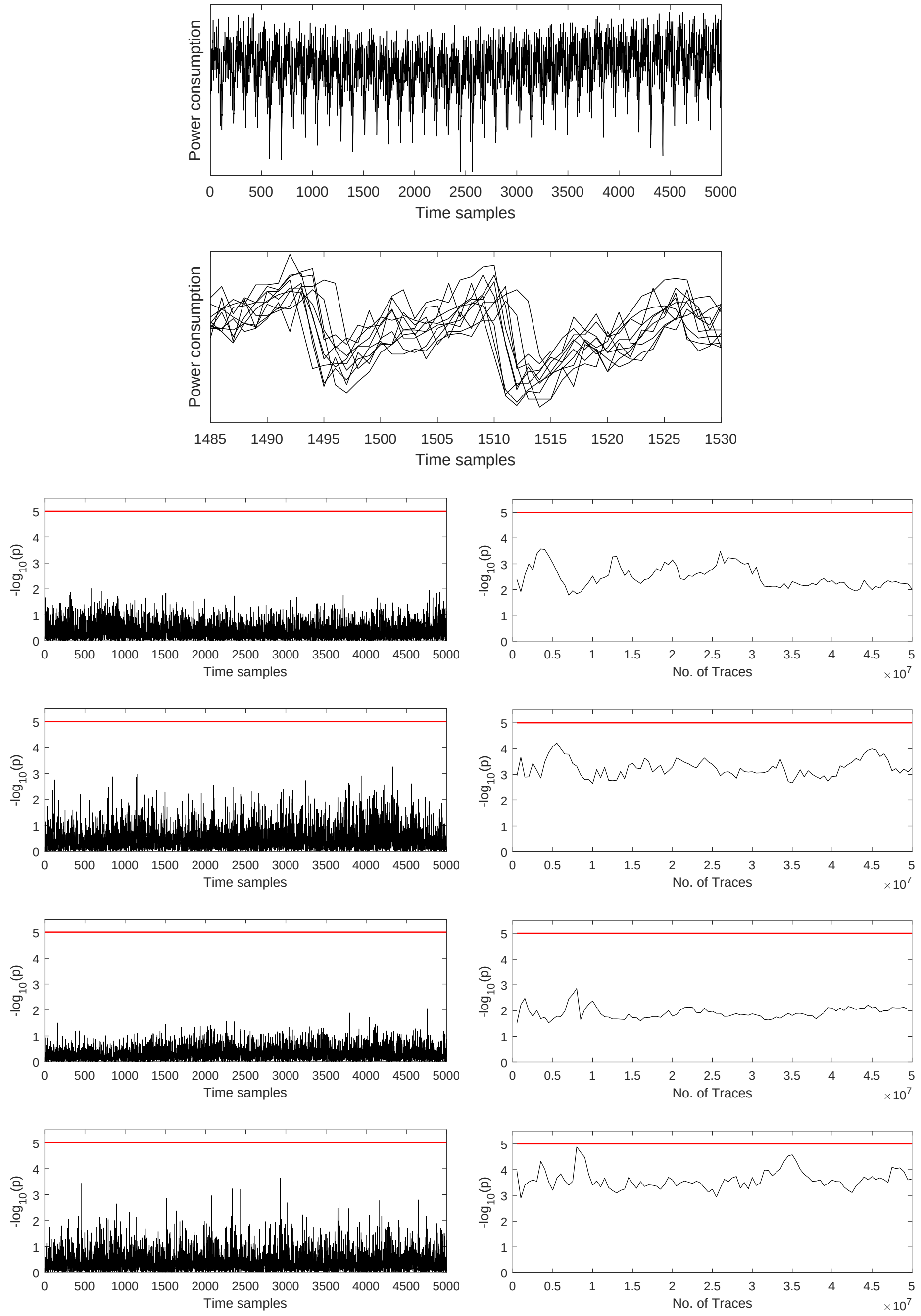

Fig. 16. Univariate leakage assessment using 50 000 000 misaligned traces (step size 500 000) of a serialized multivariate PRESENT threshold implementation. From top to bottom: 1) Sample trace, 2) Overlay of 10 sample traces, 3) first-order t-test results, 4) second-order t-test results, 5) third-order t-test results, 6) χ^2-test results.

on the entirety of points[6]. To practically verify the resilience against false pos-

[6] Note that pinpointing leakage in the time dimension is still possible due to sensitivity analysis as demonstrated in our case studies.

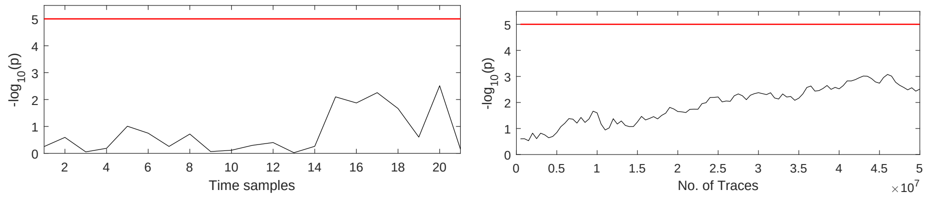

Fig. 17. Multivariate third-order t-test using 50 000 000 misaligned traces (step size 500 000) of a serialized multivariate PRESENT threshold implementation.

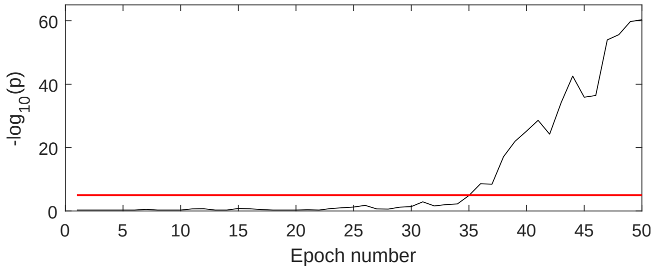

Fig. 18. DL-LA using 25 000 000 misaligned traces of a serialized multivariate PRESENT threshold implementation. For each p value 50 epochs and a validation set of 5 000 000 traces are considered.

itives, we trained our network several times on randomly generated data with random group assignments. In these evaluations we never observed any confidence exceeding $p = 10^{-2}$. Hence, we have a high confidence, that false positives are far less likely to occur with our methodology if a reasonable threshold is chosen, e.g., $p_{\text{th}} = 10^{-5}$.

False Negatives. Any leakage evaluation methodology should primarily aim to prevent false negatives. While we cannot possibly compare our methodology to all existing attacks and most certainly do not claim the impossibility of false negatives, we want to stress the importance to adopt DL-LA as one new element besides the t- and the χ^2-test in the evaluator's toolbox to severely reduce false negatives when considering multivariate information leakage.

Confidence Boosting. In a common leakage evaluation a statistical test (t-test, χ^2-test) is performed on the entirety of collected traces. Thereby, two different metrics, (i) the number of traces required to extract meaningful information, and (ii) the level of confidence the evaluator wants to achieve are tightly intertwined. More specifically, under realistic noise conditions t-test and χ^2-test are fundamentally unable to answer the question: Given a very high confidence threshold of $p = 10^{-50}$ can the attacker extract information given only very few traces? In stark contrast, our deep learning based methodology operates on two sets: One training set of size N and one validation set of size M. Here, N and M can be chosen independently from each other. While N represents the actual amount of traces available to an attacker, M should be chosen sufficiently large to reach the desired level of statistical confidence, e.g. note that the maximum level of statistical confidence that can be achieved with a given validation set equals 0.5^M and might be much lower under realistic noise conditions.

Sensitivity Analysis. As seen in Section 4 computing the gradient of an output component of the neural network with respect to the input values can provide an insight into the dependence of the classification result on each individual time sample. While this seems similar to the result of classical univariate hypothesis tests, which illustrate independent statistical tests on each point in time, there are some crucial differences: DL-LA learns a function depending on the inputs in some way, that minimizes the given loss function. This leads to two effects: (1) Points that do not contribute to leakage may still receive a non-zero component in the gradient, (2) Points that contribute to leakage, but correlate heavily with other points contributing to leakage might not be learned, as there is no intrinsic incentive for the neural net to learn redundant information. However, all of our practical case studies show, that the highest values in the Sensitivity Analysis always correspond to leakages which are also found by point-wise analyses. Yet, we want to caution against the idea that all leakage locations can be found with a single SA. Instead, the process is more iterative: After a design flaw has been identified and fixed, the DL-LA of the next design iteration might reveal new leakage locations of flaws that already persisted in the initial evaluation, but were simply not learned by the classifier.

Validation Accuracy in Isolation. Commonly, neural networks are applied to classification tasks in which the user is actually interested in obtaining a good classifier, e.g., obtain a network to distinguish cat pictures from dog pictures. In those cases a very high validation accuracy ($0.99 + \epsilon$) is expected from a suitable neural network as each individual sample is noise free and can easily be assigned to one specific group. In contrast, when evaluating side-channel traces, especially of masked implementations, the randomized intermediate values lead to an impossibility to precisely assign each individual sample to a group with high accuracy.[7] In contrast, the aim of the attacker can only be to distinguish different intermediate values statistically, i.e., *on average.* This leads to a very different expectation (compared to the image classification problem): The aim is to find a network that works better than chance (validation accuracy > 0.5) and does so consistently over a large validation set. Hence, we caution the evaluator to disregard seemingly small values for the validation accuracy, e.g. 0.505. Instead, the size of the (perfectly balanced) validation set should always be taken into account by computing the correct p-value according to the Binomial distribution.

Test and Validation Set. In deep learning, there is a common distinction between the validation set, used as a feedback mechanism to adjust the hyper parameters and the test set, another completely independent set that is used to access the accuracy of the final network. This approach is used to prevent implicit information leakage from the validation set into the trained model (through the adjustment of hyper parameters). For our case studies this distinction is not needed, because we performed all evaluations on networks with identical hyper parameters and the chosen network architecture is not special by any means, e.g., we verified that (small) variations in the size of layers will lead to comparable results.

[7] In fact, if we find a neural network with validation accuracy equal to 1.0 an attacker would most likely be able to not only succeed with DPA, but mount a successful Simple Power Analysis (SPA).

Misalignment. As seen in our case studies, DL-LA is resilient against slight misalignment through bad triggering. However, this robustness is shared with the t- and χ^2-test. When operating on severely misaligned traces due to clock randomization both DL-LA and classical tests lose orders of magnitude of confidence compared to an aligned evaluation. Fortunately, this can be partially offset by increasing the validation set to perform *Confidence Boosting*. While we performed initial experiments with CNNs which indeed provide more resilience against misalignment, we chose not to include them into our network suggestion as the usage of convolutions severely hindered the detection of higher-order (univarate or multivariate) leakages. We leave the design of a neural network combining convolutions and fully-connected layers to resist misalignment while detecting higher-order leakages as future work.

Availability. A sample implementation of our evaluation methodology based on Keras and TensorFlow, including an arbitrary precision computation of the classifier will be made freely available upon publication.

6 Conclusion

We introduced Deep Learning Leakage Assessment (DL-LA), a novel methodology to evaluate leakages in an intrinsically multivariate and horizontal way by training a classifier based on deep neural networks. In the case of univariate leakages our method outperforms the non-specific t-test as well as the χ^2-test in several case studies: We detect leakages with fewer traces and with a confidence orders of magnitudes higher compared to classical leakage assessment. However, as we cannot provide an indication suggesting the general superiority of our approach over the classical hypothesis tests in these scenarios, we suggest DL-LA as a complement to the t-test and χ^2-test and not a replacement thereof.

In the case of multivariate leakage, DL-LA effortlessly learns an accurate classifier, while multivariate extensions of the t- and χ^2-test require (i) exhaustive search over all time offsets or (ii) expert-level domain knowledge to choose the correct offset. Most importantly, we demonstrate a case study in which the classical hypothesis tests cannot detect any leakage despite having white-box knowledge about the underlying implementation while DL-LA indicates the insecurity with overwhelming confidence in a black box setting, requiring only a part of the available trace.

Our method unifies horizontal and vertical side-channel attacks, is simple to use, broadly applicable and produces results with high statistical confidence. It should be adopted as a valuable addition to the evaluator's toolbox to severely reduce false negatives in multivariate and horizontal settings.

Acknowledgments

The work described in this paper has been supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the project 271752544 "NaSCA: Nano-Scale Side-Channel Analysis".

References

1. Side-channel AttacK User Reference Architecture. http://satoh.cs.uec.ac.jp/SAKURA/index.html
2. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4727, pp. 450–466. Springer (2007), https://doi.org/10.1007/978-3-540-74735-2_31
3. Cagli, E., Dumas, C., Prouff, E.: Convolutional neural networks with data augmentation against jitter-based countermeasures - profiling attacks without preprocessing. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10529, pp. 45–68. Springer (2017), https://doi.org/10.1007/978-3-319-66787-4_3
4. Durvaux, F., Standaert, F.: From improved leakage detection to the detection of points of interests in leakage traces. In: Fischlin, M., Coron, J. (eds.) Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9665, pp. 240–262. Springer (2016), https://doi.org/10.1007/978-3-662-49890-3_10
5. Goodwill, G., Jun, B., Jaffe, J., Rohatgi, P.: A testing methodology for side channel resistance validation. In: NIST non-invasive attack testing workshop (2011)
6. Hospodar, G., Gierlichs, B., Mulder, E.D., Verbauwhede, I., Vandewalle, J.: Machine learning in side-channel analysis: a first study. J. Cryptographic Engineering **1**(4), 293–302 (2011), https://doi.org/10.1007/s13389-011-0023-x
7. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
8. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer (1999), https://doi.org/10.1007/3-540-48405-1
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
10. Masure, L., Dumas, C., Prouff, E.: A comprehensive study of deep learning for side-channel analysis. Cryptology ePrint Archive, Report 2019/439 (2019)
11. Moradi, A., Richter, B., Schneider, T., Standaert, F.: Leakage detection with the x2-test. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2018**(1), 209–237 (2018), https://doi.org/10.13154/tches.v2018.i1.209-237
12. Nikova, S., Rechberger, C., Rijmen, V.: Threshold implementations against side-channel attacks and glitches. In: Ning, P., Qing, S., Li, N. (eds.) Information and Communications Security, 8th Int. Conf., ICICS 2006, Raleigh, NC, USA, Dec, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4307, pp. 529–545. Springer (2006)
13. Poschmann, A., Moradi, A., Khoo, K., Lim, C., Wang, H., Ling, S.: Side-channel resistant crypto for less than 2, 300 GE. J. Cryptology **24**(2), 322–345 (2011)
14. Schneider, T., Moradi, A.: Leakage Assessment Methodology - A Clear Roadmap for Side-Channel Evaluations. In: CHES 2015. Lecture Notes in Computer Science, vol. 9293, pp. 495–513. Springer (2015)
15. Simonyan, Zisserman, V.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
16. Standaert, F.: How (not) to use welch's t-test in side-channel security evaluations. In: Smart Card Research and Advanced Applications - 16th International Conference, CARDIS 2018, Montpellier, France, November 2018 (2018)

17. Timon, B.: Non-profiled deep learning-based side-channel attacks with sensitivity analysis. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(2), 107–131 (2019)