Leakage Detection with the χ^2-Test

Amir Moradi[1], Bastian Richter[1], Tobias Schneider[2]
and François-Xavier Standaert[2]

[1] Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Germany
[2] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium

Abstract. We describe how Pearson's χ^2-test can be used as a natural complement to Welch's t-test for black box leakage detection. In particular, we show that by using these two tests in combination, we can mitigate some of the limitations due to the moment-based nature of existing detection techniques based on Welch's t-test (e.g., for the evaluation of higher-order masked implementations with insufficient noise). We also show that Pearson's χ^2-test is naturally suited to analyze threshold implementations with information lying in multiple statistical moments, and can be easily extended to a distinguisher for key recovery attacks. As a result, we believe the proposed test and methodology are interesting complementary ingredients of the side-channel evaluation toolbox, for black box leakage detection and non-profiled attacks, and as a preliminary before more demanding advanced analyses.

Keywords: χ^2-test · t-test · SCA evaluation · SCA distinguisher · statistical moments

1 Introduction

Motivation. Welch's t-test is commonly used in the side-channel community as a leakage detection tool. In brief, the goal of leakage detection is to provide a *qualitative* answer to the question: are side-channel measurements informative (i.e., reveal information about the data manipulated, independent of whether this information is exploitable)? In its most popular form – usually denoted as the Test Vector Leakage Assessment (TVLA) methodology – it works by comparing the leakages of a cryptographic (e.g., block cipher) implementation with fixed plaintexts (and key) to the leakages of the same implementation with random plaintexts (and fixed key) [GJJR11b, CMG+]. If a significant difference of means is observed between the leakages, it is concluded that the device leaks. As shown by Schneider and Moradi, such a methodology can be extended to the analysis of higher-order and/or multivariate leakages (by testing higher-order and/or mixed statistical moments) [SM15].

Informally, the main advantages of leakage detection are its simplicity, its efficiency (in time and data complexity) and its ability to be used with minimum implementation knowledge. These advantages are due to two main factors, both coming with natural drawbacks: (*i*) a reduction of the number of classes for which the leakages have to be estimated (typically from 256 in the case of an 8-bit target sensitive variable to only 2 classes corresponding to the fixed and random inputs), and (*ii*) a simple statistical treatment based on the estimation and comparison of statistical moments.

As discussed in [DS16], the main drawback of the first factor (i.e., the reduction of the number of classes) is a risk of false positives and false negatives. False positives correspond to the detection of samples (or tuples of samples in the higher-order multivariate case) that are not exploitable in a simple "divide and conquer" side-channel attack (e.g., because these samples correspond to plaintext variations, or intermediate values in the middle

IACR Transactions on Cryptographic Hardware and Embedded Systems ISSN 2569-2925,
Vol. 2018, No. 1, pp. 209–237
DOI:10.13154/tches.v2018.i1.209-237

rounds of a cipher that are hard to guess). False negatives correspond to cases where the two classes of the TVLA methodology have too similar leakages for being detectable (despite exploitable signal would be detected with more classes). Concretely though, these risks vanish with the number of samples tested (i.e., the size of the leakage traces).

More critically, and as discussed in [Sta17], the main drawback of the second factor (i.e., the use of a moment-based statistical treatment) is another risk of false negative typically happening when moment-based side-channel attacks (e.g., higher-order DPAs using a combination function or an estimation of moments to distinguish [PRB09, MS16]) become suboptimal compared to their counterparts using the full leakage distribution (e.g., higher-order DPAs using a Gaussian mixture model [SVO+10]) or an approximation thereof [SMSG16]. The latter risk becomes increasingly relevant (and difficult to anticipate) as the number of shares and security order of a masked implementation increases. For illustration, a masked implementation with more than 8 shares could require millions of traces for a moment-based detection, despite being breakable with a single (noise-free) trace [Sta17].

In order to mitigate this second drawback, one straightforward direction is to move from a qualitative detection test to a *quantitative* information theoretic analysis of the leakages [SMY09]. Yet, such an approach is more expensive (since it requires analyzing multiple classes) and requires access to implementation details. Therefore, it also cancels the interesting "separation of duties" between simple leakage detection tests used for preliminary / black box (qualitative) assessments, and complete (quantitative) information theoretic evaluations used to predict/bound attack complexities [DZFL14, LPR+14, DFS15].

Our contribution. Motivated by this state-of-the-art, we describe how to extend leakage detection in order to maintain an efficient (qualitative) analysis based on a limited number of classes, while making it possible to detect problematic leakages that cannot be efficiently spot by a moment-based analysis with Welch's t-test for some specific cases.

For this purpose, we start by arguing that the χ^2-test is a natural candidate for various reasons: (*i*) as Welch's t-test, it is conceptually simple and enables efficient implementations, (*ii*) as Welch's t-test, it directly allows evaluating the confidence in a detection test thanks to p-values, (*iii*) as Welch's t-test, it can be used in a black box manner (i.e., without knowing implementation details), and (*iv*) contrary to Welch's t-test, it can capture complex distributions with information lying in multiple statistical moments.

Next, we apply the proposed methodology to different settings: first univariate and multivariate higher-order simulated leakages in order to gain understanding about the proposed method, second univariate higher-order leakages corresponding to state-of-the-art Threshold Implementations (TIs) [NRS11] in order to confirm its concrete relevance. We additionally explain how to use the χ^2-test as a side-channel distinguisher, in order to perform key recovery attacks, which sometimes improve the state-of-the-art.

Based on these experiments, our most important conclusion is that Welch's t-test and the χ^2-test are nicely *complementary* in the context of leakage detection. This can be explained by observing that the aforementioned cases where leakage detection based on Welch's t-test is not sufficient typically happen in two contexts, namely: either when the noise in a masked implementation is too low (and in particular, lower than required by masking security proofs [DFS15]) – this is in fact exactly the scenario analyzed in [Sta17]; or when the information leakages are spread over several statistical moments due to physical defaults such as glitches – this is what frequently happens in the analysis of Threshold Implementations (TIs) (see for example [SM15, MS16]). As a result, running Welch's t-test and the χ^2-test naturally leads to better intuition about the type of leakages faced by the evaluator, in particular regarding the main (independence and noise) hypotheses required for masking. Typically, detection based on the χ^2-test requiring less samples than detection based on Welch's t-test should raise a warning flag. In this case, one can quite safely conclude that the analysis of the leakages requires special care (e.g., because of a

too low noise level or because of physical defaults such as glitches). Otherwise, one gains confidence that the leakages observed are "simple" (i.e., that the noise level is sufficient for masking to deliver its promises and that a single statistical moment captures most of the exploitable information). Yet, the conclusion is admittedly less definitive because the advantage of the t-test can then be due to its simpler nature, in particular in the context of multivariate distributions where detection based on the χ^2 statistic may suffer more from the increase of the leakages' dimensionality. The latter is a natural price to pay for black box evaluation and non-profiled attack methods.

The combination of these tools therefore provides a useful preliminary assessment of a masked implementation's leakages, before carrying out more elaborate evaluations and attacks.[1] This is particularly true in the context of univariate leakages where the estimation the χ^2 statistic is simple and leads to powerful detection and attacks.

Related works. In a paper from Asiacrypt 2013, Mather et al. initiated the use of tests based on the estimation of the mutual information as an alternative to Welch's t-test for leakage detection [MOBW13]. The latter therefore has similar goals as to the proposed χ^2-test, yet with two drawbacks: (*i*) the mutual information does not have a simple sampling distribution allowing the easy extraction of p-values (as with the χ^2-test); (*ii*) the mutual information is more expensive to compute (and may therefore require dedicated hardware to perform large scale analyzes). Other references discussing the exploitation of multiple statistical moments in leakage distributions include the works of Bruneau et al. (about Taylor expansions for maximum likelihood side-channel attacks) [BGH+16] and Cagli et al.(about the use of Kernel Discriminant Analysis (KDA) against masked implementations) [CDP16]. Yet, they have quite different objectives than ours. Namely, profiled attacks in the first case and dimensionality reduction in the second one. Besides, other authors also used the χ^2-test in the context of side-channel analysis, for different purposes than ours. For example, Thiebeauld et al. presented a pre-processing technique to mitigate jitter and random delay countermeasures by compressing multiple points into histograms in [TGWC17]. They used the χ^2-test to compare the generated histograms as part of a distinguisher (rather than for evaluating the detection capabilities of the χ^2-test). Linge et al. [LDL14] applied the χ^2-test (among other statistics) to compare distributions generated by algorithmic models of the attacked cipher to the observed distributions. Finally, Wagner et al. [WH17] used a χ^2 analysis to identify points of interest for a template attack. The method used is more different from our χ^2-test since their function relies on the means of different classes making it moment-dependent rather than distribution-dependent (as revealed in [WHZZ16, footnote on p. 8]).

2 Background

2.1 Welch's t-test

Statistical tests generally provide a quantitative value (i.e., a confidence level) to accept (or reject) an underlying hypothesis. In the following – considering two sets of samples – we consider the *null hypothesis* as the case where the samples in both sets are drawn from the same population (i.e., the two sets are not distinguishable).[2] Welch's t-test, where the test statistic follows a Student's t distribution, accepts (or rejects) the null hypothesis by means of comparing the estimated means (averages) of the two populations.

[1] Note that the noise condition of a masked implementation could also be verified by analyzing its shares' leakages. Yet, the latter typically requires knowing (or controlling) the implementation's randomness, which is usually not possible in the black box evaluation scenario that is most relevant to leakage detection.

[2] We call this a qualitative test (despite it produces a quantitative value to accept (or reject) the null hypothesis because it answers a binary question, and therefore it does not quantify the *amount* of information leakage nor the actual security level of an implementation (measured as a number of traces needed to perform a successful key recovery).

Let us denote the two sets by $\mathcal{Q}_0$ and $\mathcal{Q}_1$, and their corresponding cardinality, sample mean and sample variance by n_0, μ_0, and ${s_0}^2$ (resp. n_1, μ_1, and ${s_1}^2$). To this end, the t-test statistic and the degrees of freedom v are computed as

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

Based on the two-tailed Welch's t-test, the confidence level to accept the null hypothesis is estimated by means of the Student's t probability density function as

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

where $\Gamma(.)$ denotes the gamma function. As a result, small p values (alternatively, large t values) give evidence to reject the null hypothesis and conclude that the sets were drawn from different populations. It is noteworthy that the degree of freedom is sometimes ignored in the exploitation of Welch's t-test for leakage detection, and a threshold of 4.5 for the t statistic is frequently considered as a condition of detection [SM15, CRB+16, CBR+16, BGN+14]. We refer to [ZDD+17] for a recent discussion on how to set this threshold.

2.2 Leakage detection with Welch's t-test

Welch's t-test has been frequently used in the areas of Side-Channel Analysis (SCA), both as a distinguisher (e.g., classical Kocher DPA attack [KJJ99]) and as a detection tool. In the popular context of the TVLA methodology [CDG+13, GJJR11a], the Device Under Test (DUT) which contains a fixed key is supplied with fixed or random inputs (in a non-deterministic order) and the measurements (or leakage traces) are collected for those two classes. By splitting the traces into two sets $\mathcal{Q}_{\text{fixed}}$ and $\mathcal{Q}_{\text{random}}$, Welch's t-test can be conducted independently for each sample point of the measured traces. The latter typically allows assessing the leakage of an unprotected implementation (i.e., when information lies in the first-order moments of the univariate distribution corresponding to the leakage samples).

In order to extend the test so that it can detect higher-order dependencies (e.g., in order to assess the leakage of a masked implementation), a pre-processing step tailored to the target security order is needed. For example, for a second-order univariate analysis, the traces should be mean-free squared (at each sample point of the measured traces independently), for a third-order univariate analysis they should be standardized and cubed, for a second-order multivariate analysis the samples should be mean-free multiplied, etc. We refer to [SM15, MS16, RGV17] for more detailed information and efficient implementation techniques to carry out such higher-order detections (which will be used in our comparisons).

We note that the same test can be conducted using two different fixed inputs (which we will refer to as a fixed vs. fixed test, in contrast with the fixed vs. random test originally proposed). The authors in [DS16] discuss its advantages and conclude that well-chosen fixed inputs can lead to successful leakage detections with lower data complexity.

2.3 Pearson's χ^2-Test

Pearson's χ^2-test of independence is used to evaluate the dependence between unpaired observations on two variables. Its null hypothesis states that the occurrences of these

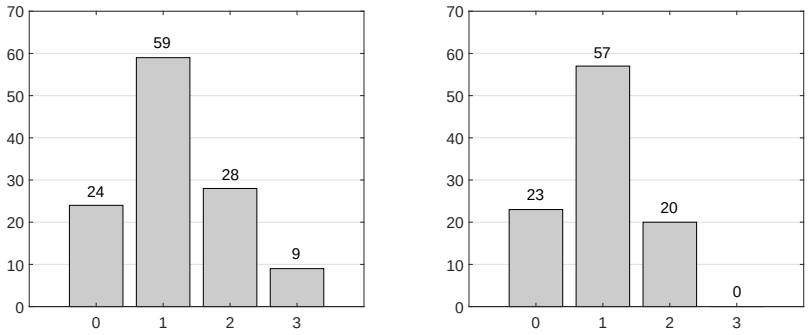


Figure 1: The histograms of the two example trace sets.

observations are independent. In contrast to Welch's t-test, this is not achieved by comparing estimated means (nor any specific statistical moment), but instead the observations are stored in a contingency table and the frequencies of each cell of the table are used to derive the test statistic which follows a χ^2 distribution.

Let us denote the number of rows (resp. columns) of the contingency table as r (resp. c), the frequency of the cell in the i-th row and j-th column as $F_{i,j}$, and the total number of samples as N (i.e., sum of all cells $\sum_{i=0}^{r-1}\sum_{j=0}^{c-1} F_{i,j}$). The χ^2-test statistic x and the degrees of freedom v are computed as

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

where $E_{i,j}$ denotes the expected frequency for a given cell (i, j) which can be derived as

$$E_{i,j} = \frac{\left(\sum_{k=0}^{c-1} F_{i,k}\right)\cdot\left(\sum_{k=0}^{r-1} F_{k,j}\right)}{N}. \tag{3}$$

For the χ^2-test, the confidence level to accept the null hypothesis is estimated through the χ^2 probability density function f as

$$p = \int_x^\infty \mathsf{f}(x, v)\, dx, \qquad \mathsf{f}(x, v) = \begin{cases} \frac{x^{\frac{v}{2}-1}e^{-\frac{x}{2}}}{2^{\frac{v}{2}}\Gamma\left(\frac{v}{2}\right)} & x > 0 \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where again $\Gamma(.)$ denotes the gamma function. As for Welch's t-test, small p values give evidence to reject the null hypothesis and conclude that for these scenarios the occurrences of the observations are not independent.

Example. For a better understanding of the underlying concept of χ^2-test we give the following example. Assume two sets, one with 120 and the other one with 100 samples. Let us consider the histograms shown in Figure 1 as their corresponding frequency of observations. Therefore, the following contingency table is made:

$F_{i,j}$	$j=0$	$j=1$	$j=2$	$j=3$	**total**
$i=0$	24	59	28	9	**120**
$i=1$	23	57	20	0	**100**
total	**47**	**116**	**48**	**9**	**220**

The degrees of freedom v can be easily calculated with the number of rows and columns

$$v = (2-1) \cdot (4-1) = 3.$$

We then exemplarily calculate the expected frequency

$$E_{0,0} = \frac{(24+59+28+9) \cdot (24+23)}{220} = \frac{120 \cdot 47}{220} \approx 25.64$$

Calculating this for all cells results in the following table:

$E_{i,j}$	$j=0$	$j=1$	$j=2$	$j=3$
$i=0$	25.64	63.27	26.18	4.91
$i=1$	21.36	52.73	21.82	4.09

Using both tables, the portions of the χ^2 value corresponding to each cell can be computed. Again, exemplarily for cell $i=0$ and $j=0$:

$$\frac{(24-25.64)^2}{25.64} \approx 0.10$$

Summing up these portions for all cells results in the χ^2 value as

$$0.10 + 0.29 + 0.13 + 3.41 + 0.13 + 0.35 + 0.15 + 4.09 = 8.64$$

Based on Equation (4) we can calculate the probability $p \approx 0.0345$ to accept the null hypothesis, i.e., the occurrences of the observations in the aforementioned sets are dependent.

3 Methodology

In this section, we explore the applicability of Pearson's χ^2-test in the scenarios of leakage detection and key recovery. After explaining the concept for univariate leakages, we discuss different strategies to extend the approach to the multivariate case.

3.1 Leakage detection with Pearson's χ^2-Test

As described in Section 2.3, Pearson's χ^2-test can be used to evaluate the dependence of two variables. To utilize this test in the context of leakage detection, we propose the following methodology.

3.1.1 Test procedure

In a typical leakage detection setting, the evaluator runs the DUT for different input classes $c \subset \mathcal{I}$, with $\mathcal{I}$ the set of possible inputs, observes physical leakages $\ell \in \mathcal{L}$, and stores them in several sets $\mathcal{Q}_j$, with $0 \leq j \leq r-1$ and r the number of input classes considered. In the simplest (exhaustive) case, the number of classes corresponds to the number of inputs $|\mathcal{I}|$, but any class can in principle be considered. Fixed vs. random (or fixed vs. fixed) classes like in the TVLA methodology and Hamming weight classes like in Brier et al.'s Correlation Power Analysis (CPA) [BCO04] are typical examples. For our description, we assume that the evaluator measures multiple traces for each class and evaluates one point of the traces as depicted in Figure 2. To assess the presence of side-channel information, we propose to use Pearson's χ^2-test and check the independence between the input classes and the observed leakages. If the test concludes with enough evidence to reject the null hypothesis, we can also conclude that the leakages are informative.[3]

[3] Up to the risks of false positive and false negatives mentioned in introduction when the number of classes is lower than $|\mathcal{I}|$.

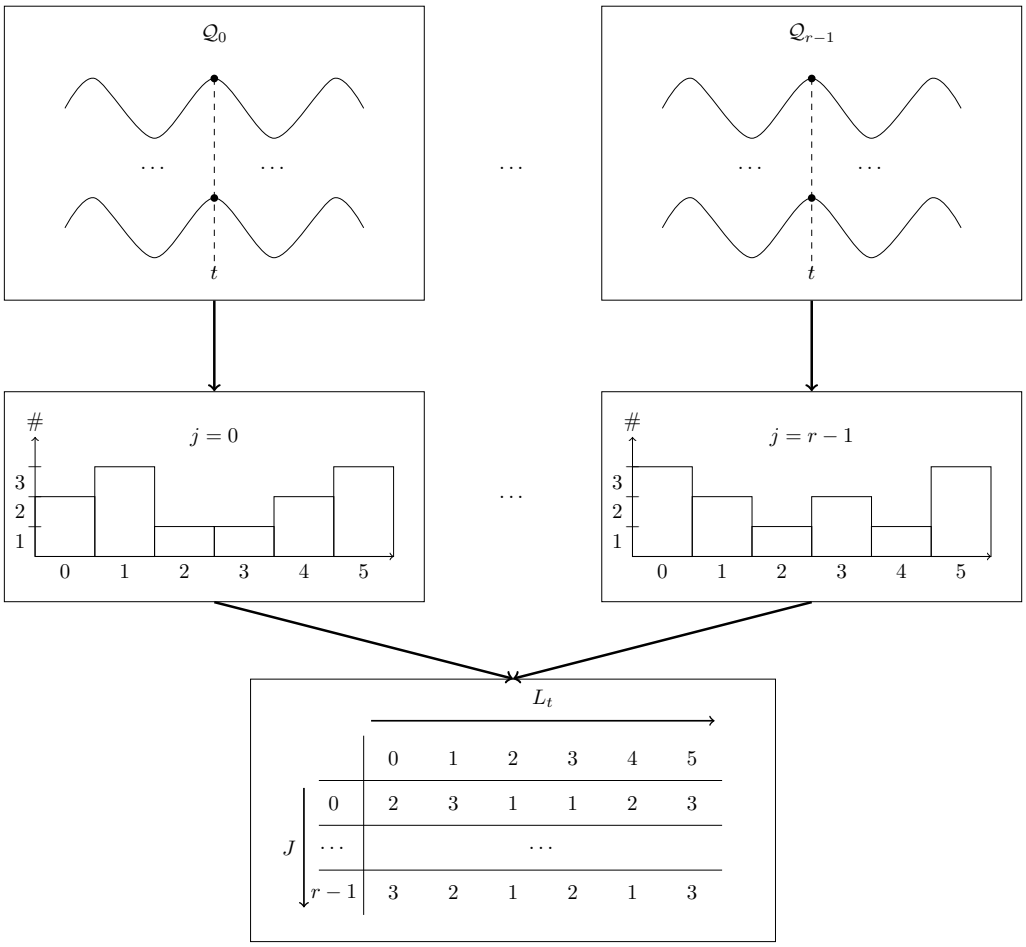


Figure 2: Our proposed leakage detection methodology based on Pearson's χ^2-test.

To perform the test, the evaluator first has to build the contingency table as described in Section 2.3. Since each cell of the table should hold the frequency of occurrence of each possible pair (j, ℓ), the measurements are grouped based on the input and histograms are created for the leakages. Each of these histograms represents one row of the contingency table as depicted at the bottom of Figure 2. It is important to note that the bins of each histogram should be the same in order to allow a fair comparison. In this simple (univariate) case, the number of bins can be chosen as the number of discrete leakage values output by the oscilloscope (e.g., 256 for an 8-bit sampling). Furthermore, columns which only contain zeros need to be removed. These columns lead to an increase of the degrees of freedom while not affecting the test statistic. Therefore, they can only impact the efficiency of the detection negatively.

Next, the evaluator just needs to compute the p-value according to the formula from Section 2.3, and compare it to a previously-chosen threshold α to decide if there is enough evidence to reject the null hypothesis. This threshold α indicates the level of significance of the test. If $p \leq \alpha$, the null hypothesis is rejected which in our case means that informative leakages are detected. The choice of this α depends on the goal of the evaluator: a low threshold provides higher confidence that the leakage is informative, but requires more measurements. In the original TVLA publications, the authors propose to use a threshold of $\alpha = 10^{-5}$ which we also use for our simulations and experiments.

3.1.2 Discussion and remarks

About the threshold. It should be noted that for Welch's t-test, the p-value is usually not explicitly computed. Instead, the test statistic |t| is compared to a threshold of 4.5 based on the relation $p = 2F(-4.5, \mathrm{v} > 1000) < 10^{-5}$. Such a relation is not easily found for the test statistic of the χ^2-test (since the degrees of freedom depend only on the number of rows and columns of the contingency table, they can change drastically between different test scenarios). Therefore, we next base our comparisons on p-values.

About multiple comparisons. We also note that since in most evaluations the traces consists of multiple sample points, the test procedure needs to be repeated for every point. As a result, the evaluator will have a large number of p-values which need to be combined. The most common solution to this problem (which has been used in many evaluation scenarios [CBR+16, MW15, SM15]) is the min-p approach (i.e., comparing the minimum p-value to the threshold). Recently a more sophisticated strategy was proposed in [ZDD+17]. However, since the problem of combining the p-values is universal to all statistical leakage detection tests, we exclude this aspect from our analysis and in the following rely on the common min-p approach.

About the selection of classes. Despite the χ^2-test naturally extents to multiple classes, the following experiments will show that in most practically-relevant cases, the reduction of the number of classes to two (e.g., fixed vs. random or fixed vs. fixed as in the TVLA methodology) leads to the most efficient detections, for the reasons intuitively pictured in introduction. Yet, it is worth observing that this extension to multiple classes may come in handy when the evaluation has to be performed in a known plaintext (rather than chosen plaintext) scenario, which makes the estimation of multiple classes mandatory. We insist that we do not claim capturing such a scenario is impossible with Welch's t-test (which would require combining the results of multiple tests) or other tools (e.g., information theoretic metrics such as a Signal-to-Noise Ratio (SNR) or the mutual information [DFS15]).

Comparison with Welch's t-test. Intuitively, the most significant difference between the two tests is that while the t-test can only compare statistical moments (for two sets of traces), the χ^2-test considers the full distributions. The latter is instrumental in avoiding the drawbacks of a moment-based security evaluation mentioned in introduction. Hence, this is the main potential advantage that we aim to analyze experimentally in the following sections. Besides, the fact that exploiting multiple moments can lead to stronger attacks, has also been previously demonstrated (e.g., in [SVO+10, SMSG16]). It motivates the next extension of leakage detection based on the χ^2-test towards a distinguisher.

χ^2 distinguisher. Following the general principle of a "partition-based DPA" [SGV09], the χ^2-test can be extended to a distinguisher by splitting the traces into several classes based on a key guess and assigning each key guess a confidence level by indicating whether this partitioning leads to a confident rejection of the null hypothesis. The latter leads to a simple DPA exploiting the full distribuition of the leakages, which is in principle very similar to Gierlichs et al.'s Mutual Information Analysis (MIA) [GBTP08]. As mentioned in introduction, a slight advantage is that it provides a confidence level for each key candidate which may help interpreting the attack results. For the rest, and as for MIA, it requires exploiting a lower number of classes than the number of key candidates (i.e., it cannot work in a strictly generic manner [WOS14]).

3.2 Extension to multivariate detections and attacks

The above-described methodology works perfectly in a scenario where the information occurs at one point in time. Such univariate higher-order leakages are commonly generated

by masked hardware circuits that process all shares concurrently, as typically observed in state-of-the-art TIs [SMG16, MW15, BGN+15, CBR+16, CRB+16]. By contrast, if each share is manipulated in a different clock cycle (e.g., in serial software implementations such as [RP10] and follow up works) information can only be recovered by exploiting a tuple of leakage samples covering all the shares.

In this respect, we note as a preliminary that finding a leaking tuple in large traces of masked implementations is a non-trivial task. Beside the naive exhaustive search which has a complexity exponential in the number of shares, there are several publications related to finding points of interest (POI) [RGV12, DS16, CDP16]. Our concern here is orthogonal. Namely, we investigate the complexity of detecting information in a tuple of samples, and (as in the univariate case – see the remark on multiple comparisons in the previous section) we ignore the problem of comparing many tuples thanks to an exhaustive or advanced analysis.

For Welch's t-test the only solution allowing to deal with these tuples is to first pre-process them in order to obtain a single sensitive sample on which a univariate evaluation can again be performed. The following "normalized product combining function" is a natural option for this purpose

$$\ell' = \prod_{i=0}^{d-1} (\ell_{t_i} - \mu_{t_i}) ,$$

where ℓ_{t_i} denotes a leakage sample at time t_i where the share i is manipulated, μ_{t_i} the sample mean at this point, and ℓ' is the pre-processed (univariate) sample. It has been shown that this function is optimal for a Hamming weight leakages [PRB09].[4]

By contrast, since the χ^2-test uses histograms rather than one specific statistical moment, there are two strategies to process multivariate leakages spread over several samples:

1. *Pre-processing.* Just as for the t-test, one solution is to combine the tuples into one pre-processed sample and conduct a univariate test afterwards. However, since the χ^2-test considers the whole distribution, it is not necessary to use a non-linear combining function. Instead, it is sufficient to simply sum the samples (without raising the result to any power). This linear combining is typically used for dimensionality reduction with Principal Component Analysis (PCA) [APSQ06] or Linear Discriminant Analysis [SA08]. Of course, non-linear combinations (e.g., with the normalized product) or Kernel Discriminant Analysis [CDP16] are also an option (e.g., in high-noise contexts where it is known to be optimal).

2. *Multivariate estimation.* Due to the distribution-based nature of the χ^2-test, the other strategy is to directly build histograms for the multivariate distribution corresponding to the tuples to evaluate.

Remark. As mentioned before, for univariate analyses the number of bins is limited to the accuracy of the sampling facility (256 bins for an 8-bit oscilloscope - which may not even all be filled in practice for low noise leakages). This picture changes significantly when evaluating multivariate leakages. By linear pre-processing the maximum number of bins extends to $256 \cdot d$, and by multivariate estimation to 256^d. Here again, it may happen that not all the bins are filled (e.g., for low noise levels), which is the typical case where the χ^2-test works best. But in general, it is necessary to limit the number of bins in order

[4] An alternative is to first use a sum combining and then raise the resulting samples to a certain power. While this is a priori less efficient, it may becomes more useful in the context of trading time and data complexity for the detection of POIs [DS16]. However, as noted above this aspect is excluded from our analysis.

to avoid the memory complexity to explode. In the following, we limited it to 256 for simplicity (formally, the only strict requirement to detect at order d is to have at least $d+1$ bins). We do not claim optimality for this choice, which is known to be a hard one (e.g., see [BGP+11] for a discussion about a similar issue in the context of MIA).

In the next section, we use simulations to investigate the performance of each of these multivariate approaches and give recommendations which of these should be used given a specific testing scenario.

4 Simulated Experiments

In this section, we use simulations to evaluate the performance of our new leakage detection methodology based on Pearson's χ^2-test. It is analysed in both univariate and multivariate leakage scenarios and compared to the t-test.

4.1 Univariate Simulations

To model the leakage of a masked hardware design in which the shares are processed in parallel (i.e., the common target for univariate higher-order evaluations), we rely on the common assumptions of a Hamming weight leakage function and additive Gaussian noise. Furthermore, we assume a Boolean-masked variable X that is split into d shares X_i with $\bigoplus_{i=0}^{d-1} X_i = X$. The leakage of these shares is summed and noise is added to the result as

$$L = \sum_{i=0}^{d-1} w(X_i) + \mathcal{N}_{0,\sigma}, \tag{5}$$

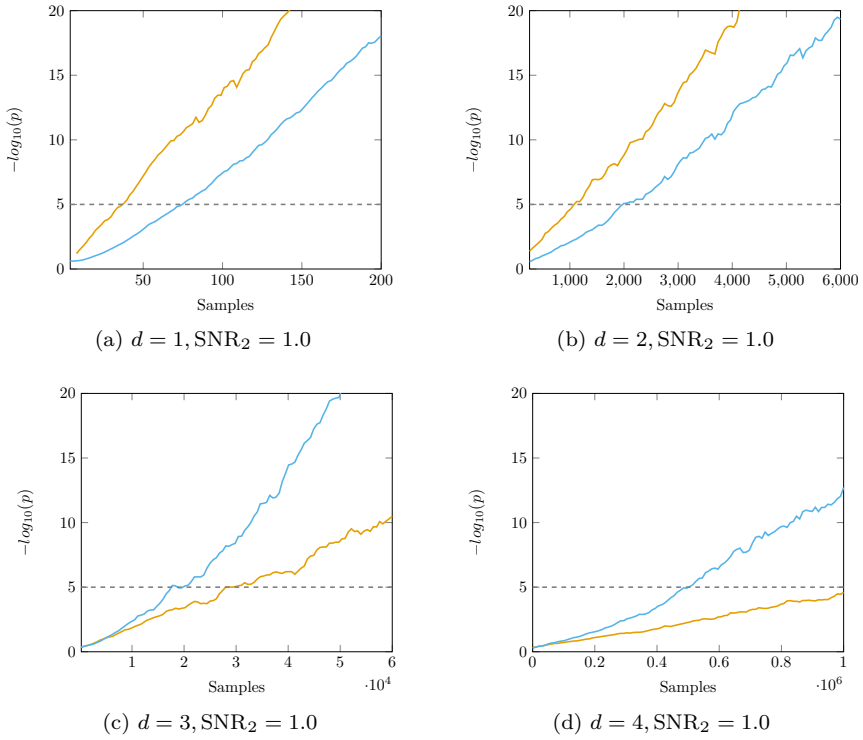
where $w(.)$ denotes the Hamming weight and $\mathcal{N}_{0,\sigma}$ the Gaussian noise with a mean of zero and standard deviation σ. Since the χ^2-test uses histograms, we round the result (after the addition of the noise) to the next integral value to emulate the effect, where the leakages are sampled by an oscilloscope[5]. For our evaluations, we consider three SNRs to cover different evaluation scenarios and examine the sensitivity of each test to increasing noise:

1. $\mathrm{SNR}_1 = 0.1$, high noise with $\sigma_1 = 4.4$,
2. $\mathrm{SNR}_2 = 1.0$, medium noise with $\sigma_2 = 1.4$,
3. $\mathrm{SNR}_3 = 10.0$, low noise with $\sigma_3 = 0.4$.

The samples are generated according to the fixed vs. random strategy in which we consider two equally sized sets of samples $\mathcal{Q}_{\text{random}}$ and $\mathcal{Q}_{\text{fixed}}$, where the samples in $\mathcal{Q}_{\text{random}}$ (resp. $\mathcal{Q}_{\text{fixed}}$) are simulated using random (resp. fixed) input values. Each experiment is repeated 150 times and average is taken for better comparison. As a metric, we compute the p-values for increasing number of samples and examine which test reaches the threshold of $p \leq 10^{-5}$ with the fewest number of samples. In the following, we use a subset of all cases to highlight the differences between the two tests. The figures for the remaining SNRs and orders are provided in Appendix B.

First, we evaluated the performance of the tests assuming an ideal leakage function as in Equation (5). The results for an unprotected implementation and masked implementations up to $d = 4$ are depicted in Figure 3 for $\mathrm{SNR}_2 = 1.0$. It is noticeable that the t-test significantly outperforms the χ^2-test for $d = 1, 2$. However, this differences becomes smaller

[5] We repeated the experiments with a larger quantization size and reached the same conclusions, as both tests were affected similarly.



(a) $d = 1, \mathrm{SNR}_2 = 1.0$ (b) $d = 2, \mathrm{SNR}_2 = 1.0$

(c) $d = 3, \mathrm{SNR}_2 = 1.0$ (d) $d = 4, \mathrm{SNR}_2 = 1.0$

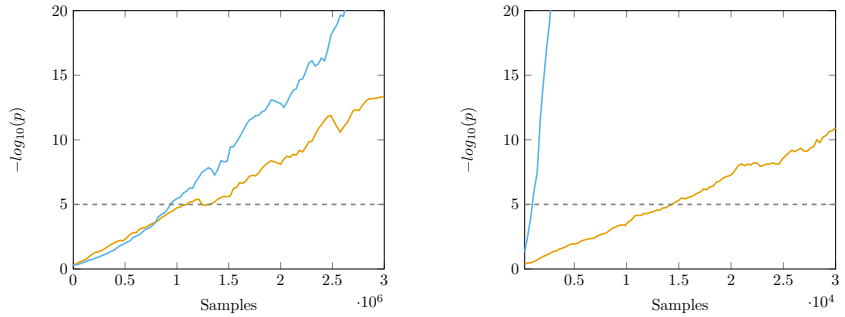
Figure 3: Performance of the (orange) t-test and (blue) χ^2-test for simulated univariate 1st-, 2nd-, 3rd-, and 4th-order leakage with $\mathrm{SNR}_2 = 1.0$.

with increasing the number of shares and is completely reversed for $d = 4$, where the χ^2-test reaches the threshold much earlier than the t-test. We expect that this advantage of the χ^2-test over the t-test continues for even higher orders, making it an ideal evaluation tool of masked hardware designs with many shares.

We also found that this advantage strongly depends on the SNR of the measurements. Figure 4 depicts the case of $d = 3$ for the other settings $\mathrm{SNR}_1 = 0.1$ and $\mathrm{SNR}_3 = 10.0$. A decrease in the SNR, also results in a reduced superiority of the χ^2-test over the t-test as shown in the left part of the figure. It is to be expected that the t-test will reach the threshold faster again for even lower SNRs. For $\mathrm{SNR}_3 = 10.0$, a similar relation to the performances of the tests can be observed. By reducing the standard deviation of the noise, the difference between the tests increases in the favor of the χ^2-test.

The later experiments are quite consistent with an information theoretic analysis of univariate leakages such as performed in [Sta17]. Namely, with low noise levels / large SNRs, the leakage distribution is a Gaussian mixture for which the estimation of a single statistical moment (as exploited by Welch's t-test) becomes increasingly suboptimal as the number of shares increases. It confirms that the χ^2-test can reveal useful intuition about the tradeoff between the noise level and the number of shares of a masked implementation in a black box manner.

Remark. As already mentioned, in most of our experiments, the χ^2-test with two input classes outperformed the χ^2-test with nine input classes (i.e., one for each Hamming weight of X) for practically-relevant p-values ($p \geq 10^{-20}$). In this section, it was only for very low noise scenarios and very small p-values ($p \leq 10^{-100}$) that the test with more input classes became better. Therefore, we do not include the results in the figures.



(a) $d = 3, \mathrm{SNR}_1 = 0.1$ (b) $d = 3, \mathrm{SNR}_3 = 10.0$

Figure 4: Performance of the (orange) t-test and (blue) χ^2-test for simulated univariate 3rd-order leakage with $\mathrm{SNR}_1 = 0.1$ and $\mathrm{SNR}_3 = 10.0$.

4.2 Multivariate Simulations

In software implementations and serialized hardware designs (e.g., in [MM13]), the shares are not processed in parallel. Instead, each share leaks at a different point in time. Therefore, we simulate the samples for these multivariate leakages as

$$L_{t_i} = HW(X_i) + \mathcal{N}_{0,\sigma},\ 0 \leq i < d \tag{6}$$

separately for each share. As noted in Section 3, we rely on the normalized product as a combining function for the t-test and evaluate three different strategies for the χ^2-test:

1. *Normalized Product.* We evaluate this non-linear pre-processing approach for both tests. The samples for each share are combined as

$$L' = \prod_{i=0}^{d-1} (L_{t_i} - \mu_{t_i})$$

and the tests are conducted on the pre-processed samples L'.

2. *Sum combining.* We evaluate this linear pre-processing approach only for the χ^2-test, since it would not be effective for the t-test which only compares the means. The samples for each share are trivially summed as

$$L' = \sum_{i=0}^{d-1} L_{t_i}.$$

As noted before, this comes with the advantage that noise terms are not multiplied.

3. *Multivariate Histograms.* We build histograms directly for the leakage tuple

$$L' = \left(L_{t_0}, L_{t_1}, \ldots, L_{t_{d-1}}\right)$$

covering all shares.

The results for the tests up to $d = 4$ for $\mathrm{SNR}_2 = 1.0$ are depicted in Figure 5 (a) - (c). It is noticeable that the t-test outperforms the χ^2-test for all cases and that the χ^2-test suffers more from increasing the number of shares. The main reason for this phenomenon is that contrary to the previous section, increasing the number of shares does not only increase the security order but also the number of dimensions which may either increase

the noise after re-combination or increase the complexity of the multivariate estimation (which both hurt the χ^2-test more than the t-test). As a result, increasing the noise standard deviation has the same impact as in the univariate case, but further amplified. And therefore, the only context where the χ^2-test can improve over the t-test is when the noise is very low. For example Figure 5 (d) shows the result of an analysis with 4 shares and such a very low noise level, where the χ^2-test is significantly better than the t-test (which is also one of the rare case where using multiple classes helps).

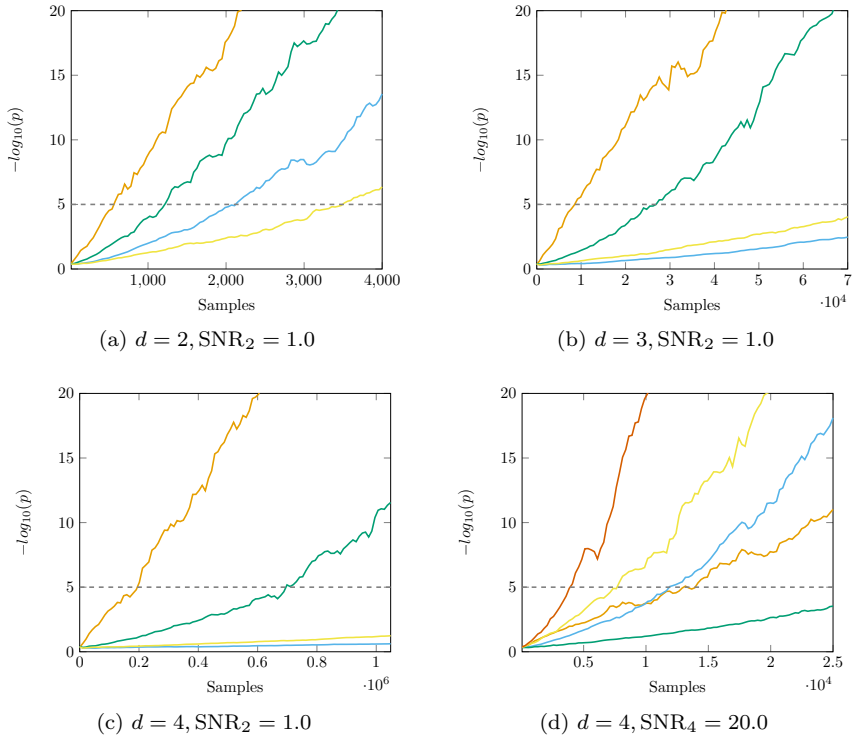
A similar intuition can be extracted from the combining functions. Namely, the normalized product is the best option for non-negligible noises, and the sum combining becomes better when the noise becomes very low. This change in effectiveness of the normalized product and sum combining functions is in line with the results of [SVO+10], where it is shown that for small noise standard deviations the normalized product performs worse than the sum, while it becomes better than the sum for larger σ. Interestingly, we also see that the χ^2-test with multivariate histograms is not the best option in our case (contrary to the use of the joint distribution in the profiled analysis of [SVO+10]).

So overall, these results outline similar but less definitive intuitions regarding the type of leakages analyzed as in the previous section. Namely, for a given number of shares, observing a better detection with the χ^2-test guarantees that the noise is too low. By contrast the opposite situation is harder to interpret, since in theory it might be due to both a large enough noise level or a hard to estimate distribution. In the latter case, launching a worst-case (information theoretic) metric is therefore advisable in order to gain a full understanding of the leakages.

5 Experiments

In this section, we compare our simulated results from the previous section with real measurements. For the experiments we measured a threshold implementation of PRESENT whose intermediate state is split into three Boolean shares as shown in Figure 6. The nibbles of the shared state (x_1, x_2, x_3) – after being XORed with the corresponding round key nibble – are serially shifted through the state register into the S-box which is divided into two functions G and F with registers in between. The output of the F function equals the masked S-box output $y_1 \oplus y_2 \oplus y_3 = y := S(x)$, and the PLayer is performed in parallel in one clock cycle. We indeed have realized the uniform shared TI of the S-box based on the details given in [PMK+11]. The cipher is implemented on the Xilinx Spartan-6 FPGA of a SAKURA-G board [sak] and its power consumption curves (through the integrated amplifier of the SAKURA board) were measured by means of a digital oscilloscope at a sampling rate of 1 GS/s. It is noteworthy that the PRESENT TI core was being operated at a frequency of around 160 MHz, and the masks for initial sharing have been provided by an AES core in a counter mode. We further made sure that the masks follow a uniform distribution.

To perform the different analyses in an efficient way, we first precomputed the histograms for each point in time of the different populations. This reduces the amount of data needed to process for different tests, so they can be performed in an efficient way. It is similar to the approach Reparaz et al. followed in [RGV17] to compute t-statistics. Appendix A shows the used C++ implementations of the t-test and the χ^2-test based on histograms. Benchmarks of these functions confirmed that the χ^2-test can be executed with a similar computational effort as a single order of the t-test. Both functions need approximately 2.8 μs per point on a single core of an Intel i7-6600U CPU @2.6 GHz. When omitting the calculation of the degree of freedom and the p value for the t-test the function only speeds up by 0.4 μs.



(a) $d = 2, \mathrm{SNR}_2 = 1.0$ (b) $d = 3, \mathrm{SNR}_2 = 1.0$

(c) $d = 4, \mathrm{SNR}_2 = 1.0$ (d) $d = 4, \mathrm{SNR}_4 = 20.0$

Figure 5: Performance of the (orange) t-test with normalized product, (green) χ^2-test with normalized product, (yellow) χ^2-test with sum combining, and (blue) χ^2-test with multivariate histograms for simulated multivariate 2nd-, 3rd-, and 4th-order leakages with $\mathrm{SNR}_2 = 1.0$.
4th-order leakages with $\mathrm{SNR}_4 = 20.0$ including the performance of (red) χ^2-test with sum combining for nine input classes.

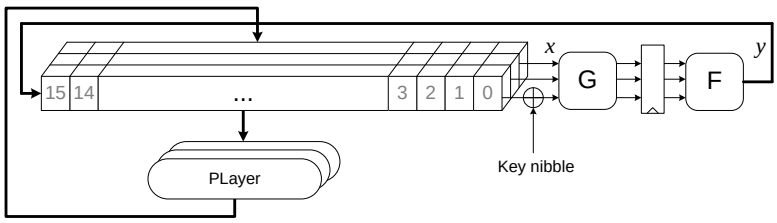


Figure 6: 3-share PRESENT TI architecture used for the experiments.

5.1 Leakage Detection

We performed two experiments to evaluate the performance of the χ^2-test in comparison to the t-test for leakage detection. First, we conducted the analysis on a set of fixed versus random traces. Secondly, we performed the tests for different combinations of two fixed plaintexts (out of eight) and when all eight fixed plaintexts are considered in the χ^2-test. For the measurements, we also followed the scenario recommended in [SM15] to efficiently randomize the order of giving either different fixed or random plaintexts.

5.1.1 Fixed versus Random

To compare the χ^2-test and the t-test in the fixed-vs-random scenario, we measured 100,000,000 traces with the plaintext being randomly selected between random values and a fixed plaintext. For the results shown in Figure 7 only 5,000,000 traces were used since the higher order leakage are already detected. The underlying TI design was developed to be first-order secure which is reassured by the p-value of the first order t-test staying above the threshold of $p = 10^{-5}$ (chosen to compare it to the commonly used threshold for t-statistics of 4.5).

Taking the t-test into account, the main leakage lies in the third order. This is also confirmed by the χ^2-test which has a very similar shape to the third order t-test. However, with $p \approx 10^{-68}$ it gives a much higher confidence than the t-test with $p \approx 10^{-40}$. Concerning the number of traces needed to exceed the threshold, the tests are also very similar detecting the leakage after 100,000 and respectively 20,0000 traces. This behavior is consistent with our simulations for univariate leakage of three shares (c.f. Figure 3), and shows the advantage of χ^2-test over t-test.

It is noteworthy to highlight that this practical experiment shows how χ^2-test captures all leakages lying in multiple statistical moments, although it is dominated by the most informative moment (here by the 3rd-order leakage).

5.1.2 Fixed versus Fixed

The other leakage detection approach we tested is fixed versus fixed. For this we recorded 20,000,000 traces with eight different fixed plaintexts, i.e., with around 2,500,000 traces for each fixed plaintext. For different combinations of two fixed plaintexts we calculated the χ^2-test as well as 1st- to 3rd-order t-tests. We further calculated the χ^2-test with eight categories utilizing all fixed plaintexts.

Figure 8 shows the results for five of such combinations. Different selections of two fixed plaintexts lead to considerably various results. One of the χ^2-tests shows a similar behavior as the fixed-vs-random test in the areas between 0 ns and 500 ns but additionally highlights a leakage in the middle of the trace. In general, different combinations (of fixed plaintexts) highlight different areas of the trace.

The χ^2-test with all eight fixed plaintexts shows the leakage at the beginning of the traces with a similar probability as the fixed-vs-random test but highlights also additional areas with lower confidence. However, it needs considerably more traces compared the fixed-vs-random test, i.e., 1,200,0000 traces versus 200,000 traces.

While the first and second order t-tests do not detect or only detect leakages with a low confidence, the third order t-test detects nearly the same areas as the χ^2-test. The same combinations of plaintexts highlight the same areas, but the χ^2-test in general gives a higher confidence with, e.g., $p \approx 10^{-64}$ compared to $p \approx 10^{-32}$ considering the first part of the trace.

Comparing the best combinations for each test in Figure 9, the χ^2-test with two fixed plaintexts and the 3rd-order t-test detect the leakages using nearly the same amount of traces. As mentioned in Section 4.1, χ^2-test with more classes also need more traces to give a significant confidence. This also corresponds to our results with the eight fixed plaintext χ^2-test which needs 1,100,000 traces to exceed the threshold. However, it achieves a higher p-value than the 3-rd order t-test using 5,000,000 traces.

5.2 Attack

To examine the χ^2-test as a distinguisher, we performed an attack on the same implementation using the same traces collected for the fixed-vs-random tests of Section 5.1.1. In other words, we used the half of the collected traces (i.e., 50,000,000 traces) associated

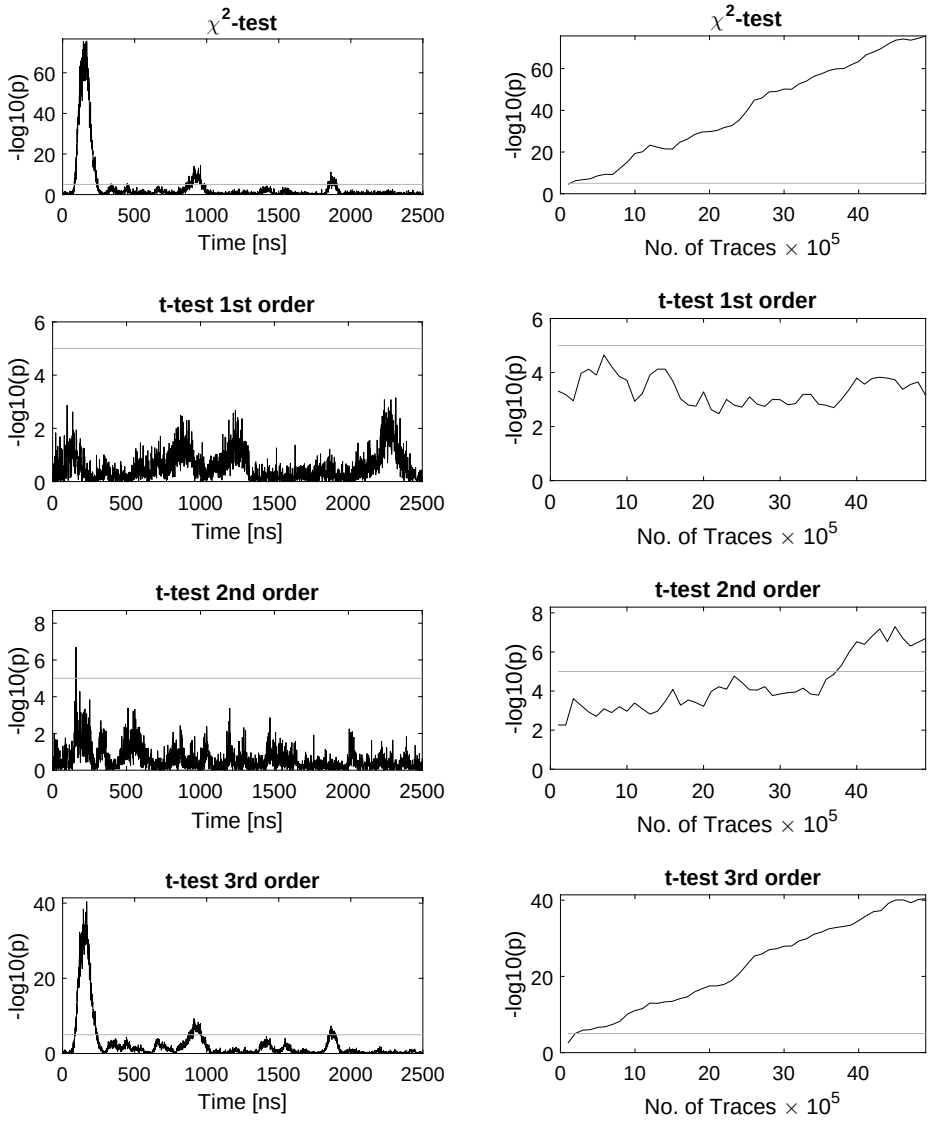


Figure 7: Results of the fixed-vs-random test using χ^2-test, and 1st to 3rd order t-test, using 5,000,000 traces.

with random plaintexts. Considering the underlying architecture of the implementation (see Figure 6), the state is shifted 4-bit-wise through the registers. Hence, we have chosen the Hamming distance (HD) between two consecutive 4-bit S-box outputs $\mathrm{HD}(S(x_i \oplus k_i) \oplus S(x_{i+1} \oplus k_{i+1}))$ as the power model. For comparison purposes we also considered 1st- to 3rd-order CPA attacks using the same power model.

We performed the attacks on the distance between the 10th and 11th S-box output and plotted the results over the number of traces and over time for the entire 50,000,000 traces in Figure 10. For the attack based on χ^2-test, the correct key is clearly distinguishable after approximately 28,000,000 traces with steadily rising χ^2 value. It also gives a high confidence level $p \approx 10^{-10}$ for the correct key candidate. The exploited leakage appears for about 48 ns, i.e., 6 clock cycles. This corresponds to the shift register architecture which shifts all values in the state registers, so the predicted HD reoccurs during all clock cycles

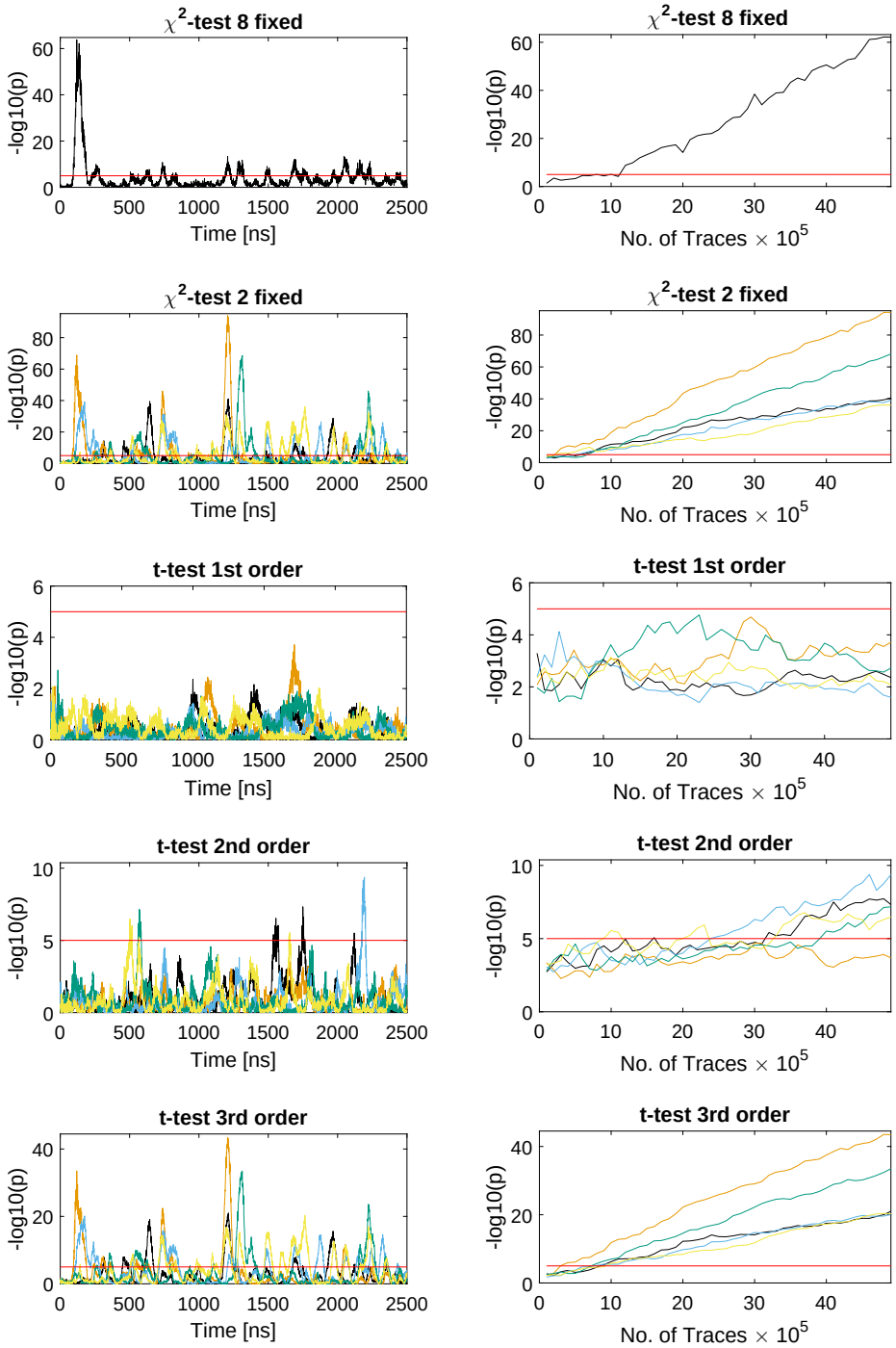


Figure 8: Results of the fixed-vs-fixed test using χ^2, and 1st- to 3rd-order t-test. The colors represent different combinations of two fixed plaintexts.

after the calculation of the targeted S-box. The CPAs however are shown to be unable to identify the correct key.

Note that for this experiment, we focused on a short window of the traces covering

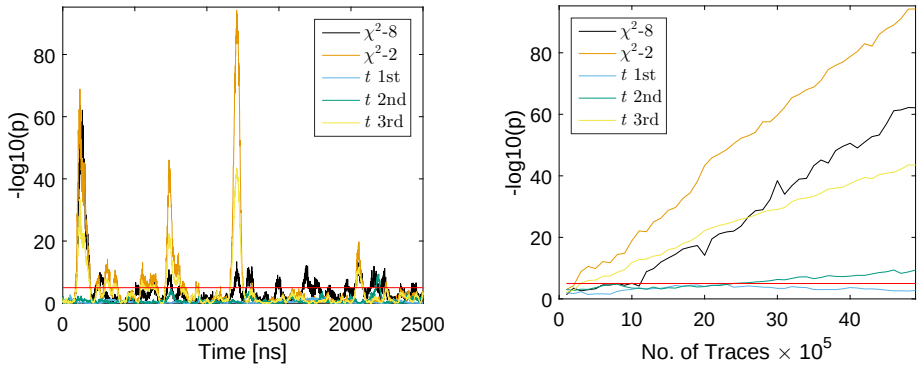


Figure 9: Comparison between 8- and 2-fixed plaintext χ^2-test and 1st- to 3rd-order t-tests for the best combinations of two fixed plaintexts.

300 ns corresponding to the first encryption round. We further – for simplicity – supposed that the 10th key nibble is known and searched in a space of 2^4 to recover the 11th key nibble. This scenario is common in serialized architecture while only for the first step of a divide-and-conquer attack the entire 2^8 key space should be searched.

6 Conclusion and Future Work

We have shown how to use Pearson's χ^2-test, a popular statistical hypothesis test, in the context of side-channel analysis. Its application in leakage detection (as a complement to Welch's t-test) and in attacks (as a distinguisher) has been demonstrated. Supported by simulation and practical experiments, we highlighted the advantages and disadvantages of the χ^2-test compared to the publicly-known and commonly-applied t-test. We mainly observe that the χ^2-test is sometimes able to outperform the t-test either if the noise level is not sufficient or the leakage is such that its information is split over multiple statistical moments. Therefore, the χ^2-test is able to detect flaws in insecure designs, which are undetectable with evaluations based on t-test. We insist however that the χ^2-test alone is not sufficient as an evaluation tool, as there are many cases in which it does not detect existing leakage for a fixed number of measurement, while the t-test does. So both tests reflect different implementation requirements (i.e., the security order for the t-test, the noise level for the χ^2-test), and our results suggest that the χ^2-test in combination with the t-test is a powerful evaluation tool.

When used as a distinguisher, it is a complement to MIA. It has all the advantages of MIA (including not being limited to a certain statistical moment and relaxing the necessity of a linearly-matching hypothetical power model) while being advantageous with respect to lower complexity and being able to give a confidence level to the key candidates. In short, the proposed methodology is shown to be useful as a novel ingredient to the side-channel analysis toolbox.

One interesting aspect of future work is the problem of combining multiple p-values. The χ^2-test might provide a better solution to this problem than the min-p approach, since it can combine the histograms of multiple points in time into one test by an easy concatenation. It is to be examined if this results in a more efficient test than the established strategies. Another further aspect is the application of the χ^2-test in the search for POIs. In the aforementioned solution [DS16], more sample points are summed than necessary to trade time and data complexity. It would be interesting to see how the χ^2-test would perform in such a scenario and if its beneficial properties can improve the search.

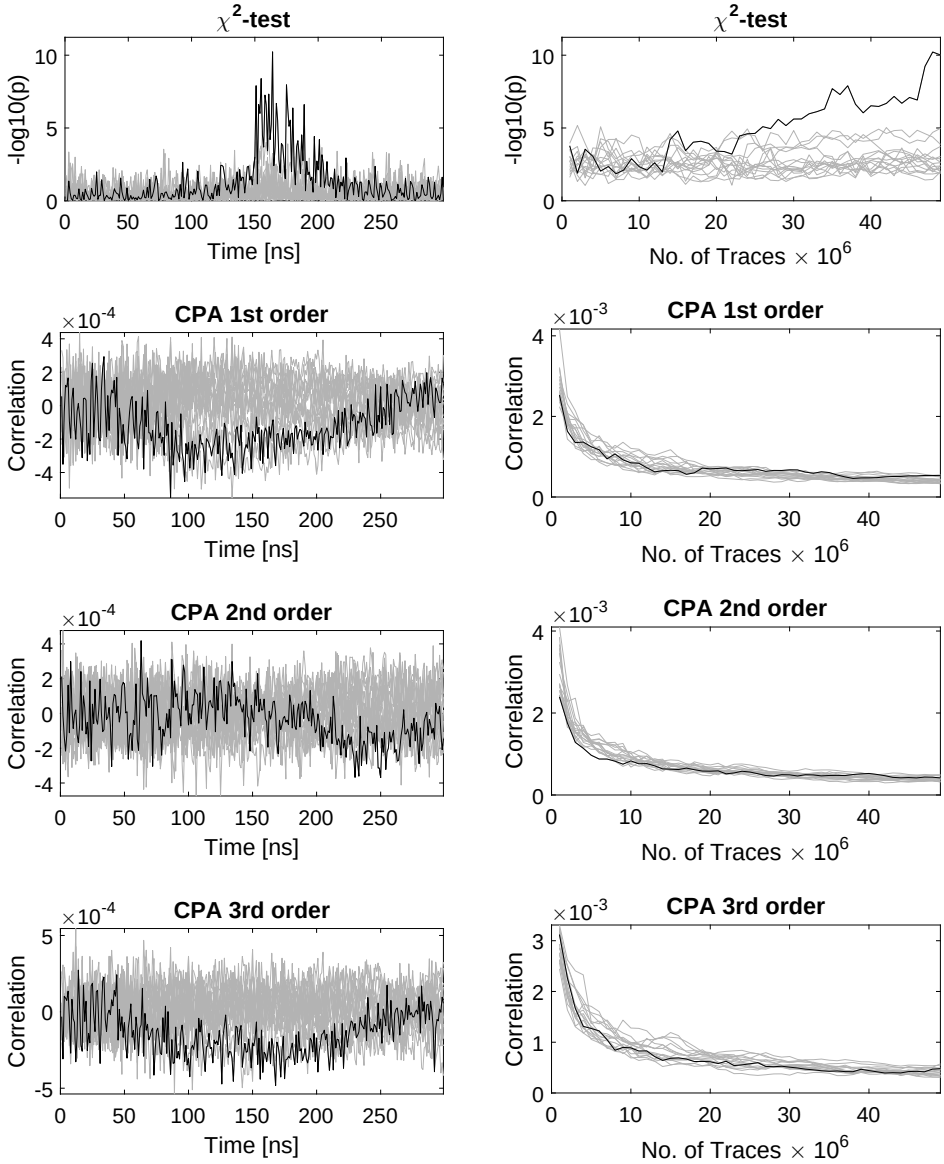


Figure 10: Results of the attack using χ^2-test as a distinguisher and the corresponding 1st- to 3rd-order CPA.

Acknowledgements

This work has been funded in part by the European Commission through the H2020 project 731591 (acronym REASSURE) and the ERC project 724725 (acronym SWORD), the German Research Foundation (DFG) through the project NaSCA (Nano-Scale Side-Channel Analysis) and the European Commission and the Walloon Region through the FEDER project USERMedia (convention number 501907-379156). François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research. We would also like to thank the anonymous reviewers for their very valuable and helpful feedback.

References

[APSQ06] Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template Attacks in Principal Subspaces. In *CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2006.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[BGH+16] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Olivier Rioul, François-Xavier Standaert, and Yannick Teglia. Taylor Expansion of Maximum Likelihood Attacks for Masked and Shuffled Implementations. In *ASIACRYPT 2016*, volume 10031 of *Lecture Notes in Computer Science*, pages 573–601, 2016.

[BGN+14] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Higher-Order Threshold Implementations. In *ASIACRYPT 2014*, volume 8874 of *Lecture Notes in Computer Science*, pages 326–343. Springer, 2014.

[BGN+15] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Trade-Offs for Threshold Implementations Illustrated on AES. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 34(7):1188–1200, 2015.

[BGP+11] Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual Information Analysis: a Comprehensive Study. *J. Cryptology*, 24(2):269–291, 2011.

[booa] Boost Framework 1.63 Documentation Chi Squared Distribution. http://www.boost.org/doc/libs/1_63_0/libs/math/doc/html/math_toolkit/dist_ref/dists/chi_squared_dist.html.

[boob] Boost Framework 1.63 Documentation Students t Distribution. http://www.boost.org/doc/libs/1_63_0/libs/math/doc/html/math_toolkit/dist_ref/dists/students_t_dist.html.

[CBR+16] Thomas De Cnudde, Begül Bilgin, Oscar Reparaz, Ventzislav Nikov, and Svetla Nikova. Higher-Order Threshold Implementation of the AES S-Box. In *CARDIS 2015*, volume 9514 of *Lecture Notes in Computer Science*, pages 259–272. Springer, 2016.

[CDG+13] Jeremy Cooper, Elke Demulder, Gilbert Goodwill, Joshua Jaffe, Gary Kenworthy, and Pankaj Rohatgi. Test Vector Leakage Assessment (TVLA) Methodology in Practice. International Cryptographic Module Conference, 2013.

[CDP16] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Kernel discriminant analysis for information extraction in the presence of masking. In *CARDIS*, volume 10146 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2016.

[CMG+] Jeremy Cooper, Elke De Mulder, Gilbert Goodwill, Josh Jaffe, Gary Kenworthy, and Pankaj Rohatgi. Test vector leakage assessment (TVLA) methodology in practice (extended abstract). ICMC 2013. http://icmc-2013.org/wp/wp-content/uploads/2013/09/goodwillkenworthtestvector.pdf.

[CRB+16] Thomas De Cnudde, Oscar Reparaz, Begül Bilgin, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Masking AES with d+1 Shares in Hardware. In *CHES*, volume 9813 of *Lecture Notes in Computer Science*, pages 194–212. Springer, 2016.

[DFS15] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making Masking Security Proofs Concrete - Or How to Evaluate the Security of Any Leaking Device. In *EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 401–429. Springer, 2015.

[DS16] François Durvaux and François-Xavier Standaert. From Improved Leakage Detection to the Detection of Points of Interests in Leakage Traces. In *EUROCRYPT 2016*, volume 9665 of *Lecture Notes in Computer Science*, pages 240–262. Springer, 2016.

[DZFL14] A. Adam Ding, Liwei Zhang, Yunsi Fei, and Pei Luo. A statistical model for higher order DPA on masked devices. In *CHES*, volume 8731 of *Lecture Notes in Computer Science*, pages 147–169. Springer, 2014.

[GBTP08] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual Information Analysis. In *CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.

[GJJR11a] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi. A testing methodology for side channel resistance validation. In *NIST non-invasive attack testing workshop*, 2011.

[GJJR11b] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for side channel resistance validation. NIST non-invasive attack testing workshop, 2011. http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[LDL14] Yanis Linge, Cécile Dumas, and Sophie Lambert-Lacroix. Using the joint distributions of a cryptographic function in side channel analysis. In *COSADE*, volume 8622 of *Lecture Notes in Computer Science*, pages 199–213. Springer, 2014.

[LPR+14] Victor Lomné, Emmanuel Prouff, Matthieu Rivain, Thomas Roche, and Adrian Thillard. How to estimate the success rate of higher-order side-channel attacks. In *CHES*, volume 8731 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2014.

[MM13] Amir Moradi and Oliver Mischke. On the Simplicity of Converting Leakages from Multivariate to Univariate - (Case Study of a Glitch-Resistant Masking Scheme). In *CHES 2013*, volume 8086 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2013.

[MOBW13] Luke Mather, Elisabeth Oswald, Joe Bandenburg, and Marcin Wójcik. Does My Device Leak Information? An a priori Statistical Power Analysis of Leakage Detection Tests. In *ASIACRYPT 2013*, volume 8269 of *Lecture Notes in Computer Science*, pages 486–505. Springer, 2013.

[MS16] Amir Moradi and François-Xavier Standaert. Moments-Correlating DPA. In *ACM Workshop on Theory of Implementation Security, TIS@CCS*, pages 5–15. ACM, 2016.

[MW15] Amir Moradi and Alexander Wild. Assessment of Hiding the Higher-Order Leakages in Hardware - What Are the Achievements Versus Overheads? In *CHES 2015*, volume 9293 of *Lecture Notes in Computer Science*, pages 453–474. Springer, 2015.

[NRS11] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. *J. Cryptology*, 24(2):292–321, 2011.

[PMK+11] Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-channel resistant crypto for less than 2, 300 GE. *J. Cryptology*, 24(2):322–345, 2011.

[PRB09] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.

[RGV12] Oscar Reparaz, Benedikt Gierlichs, and Ingrid Verbauwhede. Selecting Time Samples for Multivariate DPA Attacks. In *CHES 2012*, volume 7428 of *Lecture Notes in Computer Science*, pages 155–174. Springer, 2012.

[RGV17] Oscar Reparaz, Benedikt Gierlichs, and Ingrid Verbauwhede. Fast Leakage Assessment. In *CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 387–399. Springer, 2017.

[RP10] Matthieu Rivain and Emmanuel Prouff. Provably Secure Higher-Order Masking of AES. In *CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2010.

[SA08] François-Xavier Standaert and Cédric Archambeau. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In *CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, 2008.

[sak] Side-channel AttacK User Reference Architecture. http://satoh.cs.uec.ac.jp/SAKURA/index.html.

[SGV09] François-Xavier Standaert, Benedikt Gierlichs, and Ingrid Verbauwhede. Partition vs. Comparison Side-Channel Distinguishers: An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices. In *ICISC 2008*, volume 5461 of *Lecture Notes in Computer Science*, pages 253–267. Springer, 2009.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In *CHES*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.

[SMG16] Tobias Schneider, Amir Moradi, and Tim Güneysu. ParTI - Towards Combined Hardware Countermeasures Against Side-Channel and Fault-Injection Attacks. In *CRYPTO 2016*, volume 9815 of *Lecture Notes in Computer Science*, pages 302–332. Springer, 2016.

[SMSG16] Tobias Schneider, Amir Moradi, François-Xavier Standaert, and Tim Güneysu. Bridging the gap: Advanced tools for side-channel leakage estimation beyond gaussian templates and histograms. In *SAC*, volume 10532 of *Lecture Notes in Computer Science*, pages 58–78. Springer, 2016.

[SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

[Sta17] François-Xavier Standaert. How (not) to Use Welch's T-test in Side-Channel Security Evaluations. *IACR Cryptology ePrint Archive*, 2017:138, 2017.

[SVO+10] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The World Is Not Enough: Another Look on Second-Order DPA. In *ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.

[TGWC17] Hugues Thiebeauld, Georges Gagnerot, Antoine Wurcker, and Christophe Clavier. SCATTER : A new dimension in side-channel. *IACR Cryptology ePrint Archive*, 2017:706, 2017.

[WH17] Mathias Wagner and Stefan Heyse. Single-trace template attack on the DES round keys of a recent smart card. *IACR Cryptology ePrint Archive*, 2017:57, 2017.

[WHZZ16] Mathias Wagner, Yongbo Hu, Chen Zhang, and Yeyang Zheng. Comparative study of various approximations to the covariance matrix in template attacks. *IACR Cryptology ePrint Archive*, 2016:1155, 2016.

[WOS14] Carolyn Whitnall, Elisabeth Oswald, and François-Xavier Standaert. The Myth of Generic DPA...and the Magic of Learning. In *CT-RSA 2014*, volume 8366 of *Lecture Notes in Computer Science*, pages 183–205. Springer, 2014.

[ZDD+17] Liwei Zhang, A. Adam Ding, François Durvaux, François-Xavier Standaert, and Yunsi Fei. Towards Sound and Optimal Leakage Detection Procedure. In *CARDIS 2017*, Lecture Notes in Computer Science. Springer, 2017. to appear.

A Code

A.1 *t*-test

```
#include<boost/math/distributions/students_t.hpp>

void calc_t(unsigned long *H[2], double *range, double *t_ret, double *
    t_dof_ret, double *t_p_ret)
{
  double mean[2] = { 0.0, 0.0 };
  double var[2] = { 0.0, 0.0 };
  double n[2] = { 0.0, 0.0 };

  //only calculate for bins which are non-zero -> faster for real
    measurements
  vector<int> nonZeroBins;
  for (int idx_bin = 0; idx_bin < number_of_bins; idx_bin++)
  {
    bool isNonZero = false;
    for (size_t idx_category = 0; idx_category < number_of_categories;
    idx_category++)
    {
      if (H[idx_category][idx_bin] != 0)
      isNonZero = true;
    }
    if (isNonZero)
      nonZeroBins.push_back(idx_bin);
  }


  for (size_t idx_category = 0; idx_category < number_of_categories;
    idx_category++)
  {
    for each (auto idx_bin in nonZeroBins)
    {
      mean[idx_category] += H[idx_category][idx_bin] * range[idx_bin];
      n[idx_category] += H[idx_category][idx_bin];
    }

    mean[idx_category] = mean[idx_category] / n[idx_category];

    for each (auto idx_bin in nonZeroBins)
    {
      double temp = (range[idx_bin] - mean[idx_category]);
      var[idx_category] += (temp*temp)*H[idx_category][idx_bin];
    }

    var[idx_category] = var[idx_category] / n[idx_category];
  }

  //calculate t-value
  double mean_diff = mean[0] - mean[1];
  double variance_sum = (var[0] / n[0]) + (var[1] / n[1]);
  *t_ret = mean_diff / sqrt(variance_sum);

  //calcualte degree of freedom
  double denominator = ((var[0] / n[0])*(var[0] / n[0])) / (n[0] - 1) + ((
    var[1] / n[1])*(var[1] / n[1])) / (n[1] - 1);
  *t_dof_ret = (variance_sum*variance_sum) / denominator;

  boost::math::students_t_distribution<> t_dist(*t_dof_ret);
  *t_p_ret = 2 * boost::math::cdf(t_dist, -fabs(*t_ret));
}
```

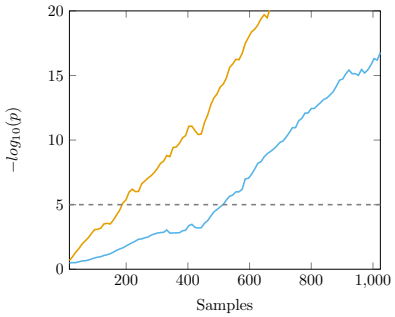
Listing 1: C++ Function for calculating t-statistics (uses Boost framework for cdf [boob])

A.2 χ^2-test

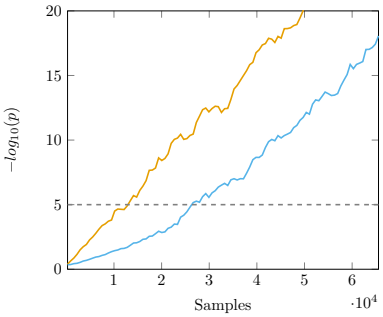
```
#include<boost/math/distributions/chi_squared.hpp>

void calc_chi(unsigned long *H[number_of_categories], double *chi_ret,
    double *chi_dof_ret, double *chi_p_ret)
{
  //find bins which are non-zero in at least one category
  vector<int> nonZeroBins;
  for (int idx_bin = 0; idx_bin < number_of_bins; idx_bin++)
  {
    bool isNonZero = false;
    for (size_t idx_category = 0; idx_category < number_of_categories;
    idx_category++)
    {
      if (H[idx_category][idx_bin] != 0)
      isNonZero = true;
    }
    if (isNonZero)
      nonZeroBins.push_back(idx_bin);
  }

  //degrees of freedom
  *chi_dof_ret = (double)(nonZeroBins.size()-1)*(number_of_categories-1);

  //chi^2 value
  double sums_rows[number_of_categories];
  double sums_columns[number_of_bins];
  double N = 0.0;

  //calculate sums for expected values
  for (size_t i = 0; i < number_of_categories; i++)
    sums_rows[i] = 0.0;
  for (size_t i = 0; i < number_of_bins; i++)
    sums_columns[i] = 0.0;

  for each (auto idx_bin in nonZeroBins)
  {
    for (size_t idx_category = 0; idx_category < number_of_categories;
    idx_category++)
    {
      sums_rows[idx_category] += H[idx_category][idx_bin];
      sums_columns[idx_bin] += H[idx_category][idx_bin];
      N += H[idx_category][idx_bin];
    }
  }

  double chi_temp = 0.0;
  //calculate chi^2 value
  for each (auto idx_bin in nonZeroBins)
  {
    for (size_t idx_category = 0; idx_category < number_of_categories;
    idx_category++)
    {
      double E = (sums_rows[idx_category] * sums_columns[idx_bin]) / N;
      double temp = (H[idx_category][idx_bin] - E);
      chi_temp += (temp*temp) / E;
    }
  }

  *chi_ret = chi_temp;
  boost::math::chi_squared_distribution<> chi_dist(*chi_dof_ret);
  *chi_p_ret = 1 - boost::math::cdf(chi_dist, chi_temp);
}
```

Listing 2: Function for calculating χ^2-statistics (uses Boost framework for cdf [booa])

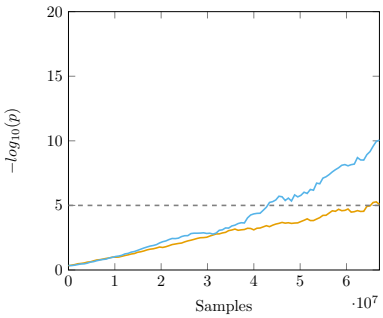
B Additional Figures



(a) $d = 1, \mathrm{SNR}_1 = 0.1$

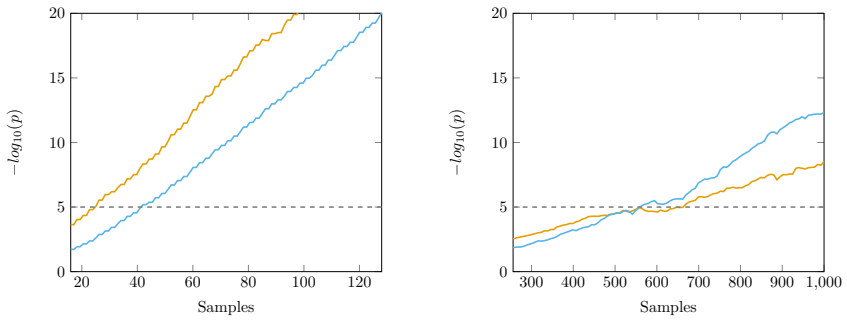


(b) $d = 2, \mathrm{SNR}_1 = 0.1$

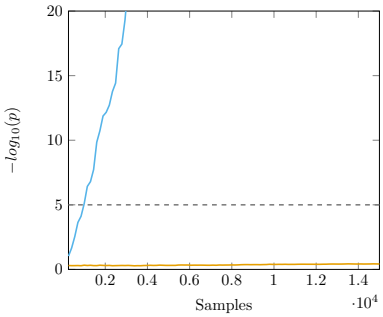


(c) $d = 4, \mathrm{SNR}_1 = 0.1$

Figure 11: Performance of the (orange) t-test and (blue) χ^2-test for simulated univariate 1st-, 2nd-, and 4th-order leakage with $\mathrm{SNR}_1 = 0.1$.

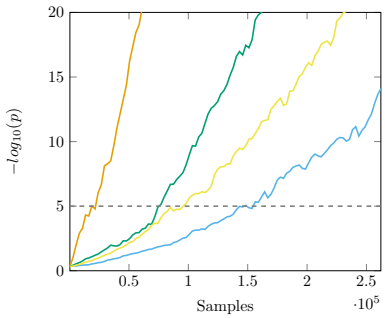


(a) $d = 1, \mathrm{SNR}_3 = 10.0$ (b) $d = 2, \mathrm{SNR}_3 = 10.0$

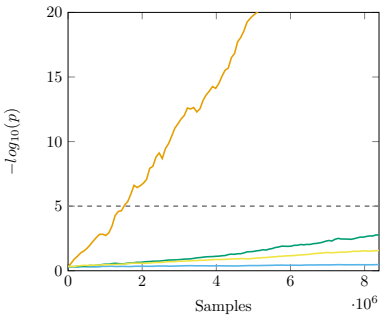


(c) $d = 4, \mathrm{SNR}_3 = 10.0$

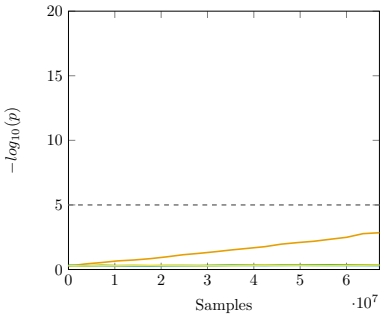
Figure 12: Performance of the (orange) t-test and (blue) χ^2-test for simulated univariate 1st-, 2nd-, and 4th-order leakage with $\mathrm{SNR}_3 = 10.0$.



(a) $d = 2, \mathrm{SNR}_1 = 0.1$

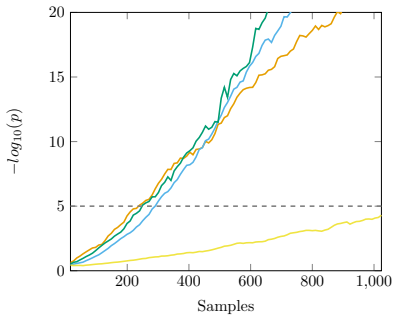


(b) $d = 3, \mathrm{SNR}_1 = 0.1$

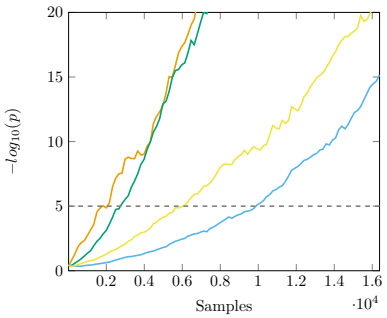


(c) $d = 4, \mathrm{SNR}_1 = 0.1$

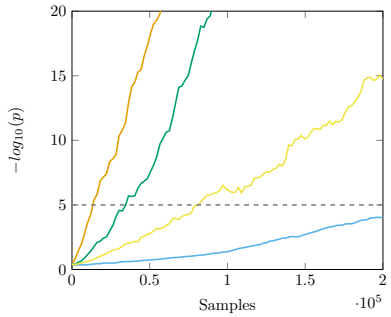
Figure 13: Performance of the (orange) t-test with normalized product, (green) χ^2-test with normalized product, (yellow) χ^2-test with sum combining, and (blue) χ^2-test with multivariate histograms for simulated multivariate 2nd-, 3rd-, and 4th-order leakages with $\mathrm{SNR}_1 = 0.1$.



(a) $d = 2, \mathrm{SNR}_3 = 10.0$



(b) $d = 3, \mathrm{SNR}_3 = 10.0$



(c) $d = 4, \mathrm{SNR}_3 = 10.0$

Figure 14: Performance of the (orange) t-test with normalized product, (green) χ^2-test with normalized product, (yellow) χ^2-test with sum combining, and (blue) χ^2-test with multivariate histograms for simulated multivariate 2nd-, 3rd-, and 4th-order leakages with $\mathrm{SNR}_3 = 10.0$.