Neyman's Smoothness Test: a Trade-off between Moment-based and Distribution-based Leakage Detections

Si Gao, Elisabeth Oswald, and Yan Yan

***Abstract*—Leakage detection tests have become an indispensable tool for testing implementations featuring side channel countermeasures such as masking. Whilst moment-based techniques such as the Welch's t-test are universally powerful if there is leakage in a central moment, they naturally fail if this is not the case. Distribution-based techniques such as the χ^2-test then come to the rescue, but they have shown not to be robust with regards to noise. In this paper, we propose a novel leakage detection technique based on Neyman's smoothness test. We find that our new test is robust with respect to noise (similar to the merit of Welch's t-test), and can pick up on leakage that is not located in central moments (similar to the merit of the χ^2-test). We also find that there is a sweet-spot where Neyman's test outperforms both the t-test and the χ^2-test. Realistic measurements confirm that such a sweet-spot is relevant in practice for detecting implementation flaws.**

I. Introduction

SINCE Kocher's seminal work [1] in the late 1990s, the crypto community realised that the threat of using auxiliary information (execution time [2], power consumption [1], electromagnetic emission [3], etc.) for key-recovery is severe. The past two decades have seen a flourishing of new developments in attack techniques (for instance, Template attacks (TA) [4], Correlation Power Analysis (CPA) [5], Mutual Information Analysis (MIA) [6], etc.), as well as delicate applications of those in various protected/unprotected cryptographic implementations. With countless existing attacks, cryptographic engineers and evaluation labs face a challenge: how can they effectively test a given implementation w.r.t. arbitrary attacks?

One tool that is particularly useful in spotting arbitrary data dependencies in implementations is to perform a non-specific leakage test, as initially formalised in [7]. A non-specific test ascertains if there is a significant statistical difference between a group of power traces with the same fixed plaintext and a group of power traces with randomised plaintexts ("fixed-versus-random"). The original proposal uses Welch's *t*-test; however, one can always choose other statistical tests to replace the t-test or even test among groups with different fixed plaintexts ("fixed-versus-fixed" [8]). The "magic" is, unlike specific tests that set a known target state (e.g. the Sbox outputs) in advance, such test can potentially detect any kind of leakage emitting from the encryption.

Unfortunately, the Welch's *t*-test is incredibly limited: the overall detection is bounded to just the first statistical moment. In other cases (e.g. a d-share masking scheme), it is required that the measured traces get pre-processed, whether by taking the power d of each sample (univariate leakage) or combining d samples using their "mean-free product" (multivariate leakage). Besides the efficiency loss [9], [10], the test relies on the d-order moment only: any other difference will be ignored.

To overcome the shortcomings of Welch's t-test, and the high computational cost of full distribution based techniques such as CMI [11], Moradi et al. introduced the χ^2-test as a complementary tool [9]. Unlike Welch's t-test, the χ^2-test compares the entire distribution, thus does not require the pre-processing step. Without the error accumulation in the pre-processing step, the χ^2-test can outperform Welch's t-test in certain higher-order masking schemes. More specifically, the authors claimed it complements Welch's *t*-tests in two circumstances, namely: when the implementation provides very limited noise, and when the leakage is spread among various statistical moments [9]. However, being "orthogonal" suggests there could be a *terra nullius* sitting in-between where both tests become ineffective: whether because the noise level being too high (for χ^2) or too many shares to combine in the pro-processing step (for Welch's *t*-test).

Our contribution. In this paper, we propose a novel leakage detection test based on Neyman's smoothness test [12], [13]. The highlight of this construction is that it achieves a more balanced trade-off between the *t*-test and the χ^2-test. More specifically, in a low-order masking with high noise setup, it only falls slightly behind *t*-test (with the correct statistical order), while outperforming the χ^2-test. On the other hand, in a higher-order masking with low noise setup, Neyman's test stays close to the χ^2-test (even better in certain circumstances), while significantly surpassing Welch's *t*-test. Furthermore, we find a particular sweet-spot where both Welch's *t*-test and the χ^2 become less effective, where Neyman's test outperforms both previous constructions. Our realistic experiments confirm that such sweet-spot is relevant for practice: considering Neyman's test combines information from various moments (like the χ^2-test), it can serves as a more noise-resilient alternative to the χ^2-test for implementation flaw detection.

Organisation. The following section presents some basic preliminaries: Boolean masking schemes, leakage detections in general, moment-based detections (e.g. Welch's *t*-test) and

S. Gao, E. Oswald and Y. Yan are with the Department of AI and Cybersecurity, University of Klagenfurt, Austria.
(E-mail: Si.Gao@aau.at, Elisabeth.Oswald@aau.at, Yan.Yan@aau.at).

This work was supported by the ERC under grant agreement 725042 (SEAL).

distribution-based detections (e.g. χ^2-test). Neyman's smoothness test, along with how it can be applied in a leakage detection setting is introduced in Section III. Section IV further provides a comprehensive set of simulations to determine the detection power of the proposed technique in comparison with Welch's t-test and the χ^2-test. Realistic experiments in Section V confirm the observations from simulations, as well as the test's potential for combining information from various statistical moments in practice. We provide a discussion in Section VI, explaining some salient technical details about the "orders" of Neyman's test as well as elaborating on the strengths and weaknesses of the three mentioned tests. Finally, we conclude this paper in Section VII.

II. Preliminaries

A. Boolean masking

In this paper, we use Boolean masking for all experiments: for a d-share masking scheme, a secret x is presented as a d-tuple $(x_1, ..., x_d)$, whereby $x = x_1 \oplus \cdots \oplus x_d$. Depending on the specific implementation, the data shares are either processed in parallel or in sequence. The former type of implementation is typically found in hardware, whereas the latter is often found in software platforms.

When shares are processed in sequence, one can expect that the power trace captures the leakage of each share $L_i(x_i)$ at one individual time sample. On the other hand, when shares are processed in parallel, it is often assumed that the measured leakage (at the point in time when the respective shares are processed) can be understood as a linear sum of all leakages $L_i(x_i)$:

$$L(x_1, ..., x_d) = L_1(x_1) + ... + L_d(x_d)$$

A fundamental pre-condition of the above expression is that shares should leak independently of each other. With this assumption, and a suitably designed masking scheme, it is possible to formally prove and verify the security of such a scheme up to some order. In other words, no attack is theoretically possible unless an adversary creates joint leakage by combining the leakage from the independent shares somehow.

However, it is possible that an implementation does not completely adhere to the security assumptions of the proof and thereby introduces a "flaw". Whilst formal verification can identify problems in masking schemes that are captured by the security model, it is not able to identify flaws that are due to physical interactions that are (*a priori*) unknown to the designer. Hence these can only be detected using statistical techniques applied to real device measurements.

B. Leakage detection

A straightforward strategy to verify the security of a cryptographic implementation is to try all possible attacks and see if any one of them could succeed. Alternatively, one may also assess the dependency between the leakage and the intermediates using statistical tools such as NICV [14] and ρ test [8][1]. These approaches can be classified as specific leakage detections as they all require the verifier to select a set of target intermediates to be tested.

While specific detections benefit from the fact that any detected leakage can be easily converted to successful attacks, this strategy suffers from its limited coverage: only the selected intermediates can be verified and there is no guarantee over those omitted. Attempts trying to cover all exhibited intermediates within the implementations usually turns out to be impractical, not only for the enormous computation workload required but also due to the fact that unexpected intermediates could be induced in various ways. For instance, for software implementations, the compilation could introduce additional intermediates along with unexpected interactions that severely undermines the security: for Boolean masking, if each share is individually stored, the security order in practice could be halved [16]. Due to the complex logic interaction within the ALU, security reduction in other implementations could even go beyond the factor 2 [17]. Similarly, hardware platforms could also be affected by the synthesis/P&R process [18]. Physical effects, such as coupling [19], [20], could also cause unexpected security loss.

To overcome the above restrictions, more attention has been directed to the research field of so called non-specific leakage detections which often abbreviated simply as leakage detection. The central idea behind is that it is possible to identify data dependencies without performing a concrete attack. This can be achieved by comparing the statistical distributions resulting from two sets of traces, which are derived as follows. The first set consists of traces where the input data to the cryptographic algorithm get fixed to some (arbitrary) value. The second set consists of traces where the input data to the cryptographic algorithm are chosen randomly. If both distributions are "the same" then there cannot be any information about the fixed data in the first set (as the fixed set shows the same characteristics as the random set). But if the two sets are statistically different, then there must be some information in the set with fixed inputs. Such test can also be performed in a fixed-v.s.-fixed manner [8], which can be even more trace-efficient (i.e. requires less measurements).

A leakage detection test is a classical statistical hypothesis test, therefore prone to two types of errors. A test can wrongly identify a leak (false positive), and a test can fail to identify a leak (false negative). For a test to be useful, both types of error need to be kept small. Both types of errors are a function of the "magnitude" of the kind of leaks that we wish to find (effect size) and the number of observations that we can acquire. Clearly this implies a trade-off: smaller effects must require more observations to keep the number of false positives and false negatives low. For a set of given parameters (i.e. the effect size and the number of traces), the power of a test is defined as the proportion of leaks that can be detected when the null hypothesis (i.e. a leak exists) is true. Higher powered tests are better because they detect more existing leaks.

[1]There might be confusion on whether the "generic" property [15] helps make these tool "non-specific". Without further restriction, the answer is no: we provide a brief discussion on this in the Appendix.

For most tests, an analytical expression (which relates its error rates, effect size and the number of observations) is not available; therefore the power of test has to be determined via simulations. This process is called *a priori power analysis*.

In the context of masking schemes, the goal of masking is to ensure the measured leakage is independent of secret intermediate values. This is usually achieved by generating multiple data shares with random values. Since each share alone is effectively a random value, their independent leakage (i.e. not combined with other shares) is also a random value. Hence the linear sum of the leakages should also appear random which can be confirmed by a leakage detection test. On the other hand, however, for a flawed scheme or implementation, the leakage distribution of the fixed data set may exhibit some form of (subtle) dependency to the fixed data which could be revealed by a suitable leakage detection test.

Leakage detection can also be used to confirm the masking order of a scheme. If the scheme uses d shares, then after some pre-processing (in a parallel scheme computing $\left(\sum_{i=1}^{i=d} L_i(x_i)\right)^d$ and in a sequential scheme computing $\prod_{i=1}^{i=d} L_i(x_i)$) will produce joint leakage of all d shares. A leakage detection test can therefore be carried to confirm no lower order combination can result in detectable leakage.

Despite the existing security proofs, as pointed out previously, both software and hardware implementations in practice bring unexpected leakage that is not captured by the model assumption. Although various leakage simulators (with HW or profiling models like [21]) have been proposed recently, capturing all physical/micro-architectural effects remains a challenge [22]. To date, leakage detection tests remain an indispensable tool to check the actual security of masked implementations. This perhaps explains the interest in constructing effective test regimes over the past years. In principle (m)any statistical test could be utilised as it is a priori unclear how the distribution of the fixed and the random data set will differ, but two competing approaches have emerged in the side channel literature. Firstly there are the moment-based tests (e.g. Welch's t-test [7]). These tests excel at identifying leaks in a central moment of a distribution. Secondly there are distribution-based tests. Here two options have been investigated: based on computing an information theoretic metric (MI) [11] and based on a discretised representation of the data (χ^2) [9]. The former type (MI based) has the (significant) drawback of being computationally expensive; therefore it is less used in practice (other than its original proposal in [11]).

C. Moment-based v.s. Distribution-based approach

Welch's t and χ^2 represent two different approaches for testing the distributions: Welch's t-test focuses on a specific statistical moment ("moment-based"). The χ^2-test, on the other hand, potentially detects any existing difference within the entire distribution. The comparison in [9] shows:

- In Hamming weight simulations, with medium noise, χ^2 falls behind Welch's t-test for $d = 1, 2$, but outperforms Welch's t-test for $d = 3, 4$.
- In Hamming weight (HW) simulations, $d = 3$, χ^2 is better in general, but the advantage fades with larger noise
- In a realistic threshold implementation, χ^2 is slightly better than the 3rd order Welch's t-test (almost negligible). As the implementation is likely noisy, the advantage is most likely from combining the weak (but existing) second order leakage and the strong 3rd order leakage.
- In a realistic threshold implementation, non-profiled attacks with the Hamming distance (HD) model, the χ^2-test is significantly better than 1-3 order CPA[2].

In the following sections, we extend the above observations with more non-HW leakage models and propose a new test that achieves a more balanced trade-off between the χ^2-test and the Welch's t-test, accompanied with theoretical justifications, simulations and realistic verifications.

III. Neyman's Smoothness Test

The motivation for using the χ^2-test is to compensate the incapability of Welch's t-test that it cannot detect leakages that did not present in the central moments. As [9] demonstrates via a range of experiments, there should be a "sour"-spot where both Welch's t-test and the χ^2-test become inefficient. In particular, with medium noise level and $d = 3$ or 4, both tests show poor performance because the t-test is hampered by the limited magnitude of dth-order statistic moment, and the χ^2-test is hampered by the noise level. As leakage detection is particularly useful in the initial exploration of the leakage of a device, it is important for any tool to work reliably well across a range of scenarios (i.e. leakage functions and noise levels).

Finding robust and versatile tests to distinguish arbitrary distributions is a well appreciated issue in statistics. The any-order tests (e.g. χ^2, Kolmogorov-Smirnov, Cramer-von-Mises, Anderson-Darling etc.)—also called "omnibus" tests—are known to have limited power against certain alternatives [23], [24]. The obstacle here is that there is little constraint on the alternative hypothesis. Take the classical fixed-versus-random leakage detection setting for instance, the null hypothesis is simply that the two distributions are the same. The alternative hypothesis, on the other hand, has no constraint other than the distributions being somehow "different". This imposes strong generality requirement over the test which inevitably reduces the statistical power, particularly in noisy contexts. Neyman's test [12], on the other hand, provides a more flexible trade-off between generality and the statistical power: by "limiting" the alternative hypothesis as a combination of independent basis functions (aka sacrificing some generality), one can achieve a better trade-off between generality and statistical power.

A. Neyman's Smoothness Test

If a Cumulative Distribution Function (CDF) F is continuous and strictly increasing over its support, the inverse $Q = F^{-1}$ exists (aka the quantile function) and is also continuous and strictly increasing. By definition, for any

[2]Note that this is not exactly a fair comparison though: the χ^2 distinguisher uses only the 9 categories (HD=0-8), not the numerical values, whereas the CPA distinguisher uses the 9 HD values.

$0 \leq a \leq 1$, we have $F(Q(a)) = a$. Thus, if we consider the distribution of $F(X)$,

$$Pr(F(X) \leq a) = Pr(X \leq Q(a)) = F(Q(a)) = a$$

$F(X)$ is uniformly distributed on the unit interval. Note that the only pre-conditions here are $X \sim F$ and F is continuous and strictly increasing: the specific distribution of F is irrelevant. Therefore when testing whether X has distribution F_0, one can instead test the uniformity of $F_0(X)$ [25]. This reduces the general goodness-of-fit problem to testing whether a distribution is uniform or not. Note that depending on the specific measurement setup, realistic side-channel measurements can also be discrete: for clarity, we proceed our discussion here using the continuous case. Discussion about the discrete case can be found in Appendix C.

To test the uniformity of $F_0(X)$, Neyman proposed to use a series of orthogonal polynomials ψ_i for $i = 0, 1, 2, \ldots$ (called the "function basis"). Each polynomial should be normalised ($\int_0^1 \psi_i^2(x)\ dx = 1$) and orthogonal to all others ($\int_0^1 \psi_i(x)\psi_j(x)\ dx = 0,\ i \neq j$). The original proposal [12] used the normalized Legendre polynomials where

$$\begin{aligned}
\psi_0(x) &= 1 \\
\psi_1(x) &= \sqrt{3}(2x - 1) \\
\psi_2(x) &= \sqrt{5}(6x^2 - 6x + 1) \\
\psi_3(x) &= \sqrt{7}(20x^3 - 30x^2 + 12x - 1) \\
\psi_4(x) &= 3(70x^4 - 140x^3 + 90x^2 - 20x + 1)
\end{aligned}$$

Neyman's original proposal limits the basis to degree $k = 4$. However, one can compute the normalised Legendre polynomial of any order k as [13]:

$$\psi_k(x) = \frac{\sqrt{(2k+1)}}{k!} \frac{d^k}{dx^k}(x^2 - x)^k$$

Alternatively, one can also use the trigonometric series given by

$$\psi_k(x) = \sqrt{2}cos(\pi k x)$$

Under the null hypothesis, $X \sim F_0$ ensures $F_0(X)$ follows a uniform distribution. In other words, $\psi_i(F_0(X))$ becomes random sampling on the support of ψ_i. Denote the sample size of X as n. When n is large, by the law of large numbers, the sample mean $\overline{\psi_i(F_0(X))}$ goes to 0. Meanwhile, the Central Limit Theorem (CLT) ensures that $\sqrt{n} \cdot \overline{\psi_i(F_0(X))} \sim N(0, 1)$, where $N(0, 1)$ is the standard Gaussian distribution. Since each polynomial is orthogonal to one another, Neyman's smoothness test verifies whether the following statistic follows the χ^2 distribution with degree of freedom k [25]:

$$\sum_{i=1}^{k} n\overline{\psi_i(F_0(X))}^2 \sim \chi^2(k)$$

B. Two samples' adaptive Neyman's Test

Historically, Neyman proposed this test as a goodness-of-fit test where the expected distribution is known. This is evidently not a good match to our application use case where we are interested in testing the equality of two sampled distributions to decide if there is a leak or not.

A few papers have extended Neyman's smoothness test to a two-sample equality test [26] [13]. The former favours situations where the two sets have very unequal sizes, which does not align with our use case. Utilising the idea from Fan [27], the latter constructed a two-sample test by using the maximum norm. For n samples of $X \sim F$ and m samples of $Y \sim G$, denote F_n the empirical CDF of F. Let $V = F_n(Y)$ and $\psi_i = \overline{\psi_i(V)}$, we have:

$$\sqrt{\frac{nm}{n+m}}(\psi_1, ..., \psi_k)^\top \sim N(0, \mathbf{I}_k)$$

and therefore the tested statistic follows:

$$\Psi_k = \sqrt{\frac{nm}{n+m}} \max(\psi_1, ..., \psi_k) \sim |N(0, \mathbf{I}_k)|_\infty$$

For any significance level α, the null Hypothesis is rejected if and only if

$$\Psi_k \geq \Phi^{-1}\left(\frac{1 + \sqrt[k]{(1-\alpha)}}{2}\right)$$

where Φ stands for the cumulative standard Gaussian distribution. The maximal degree k can increase with the available samples (i.e. n, m): when n and m are comparable, the authors claim k can be the order of $n^{1/4}$ for trigonometric series ($n^{1/9}$ for Legendre polynomials).

C. Neyman's test for leakage detection

With the fixed-versus-random strategy, any two-sample equality test can be used for leakage detection. Therefore, when using [13] the only decision left is regarding the maximal order k. Throughout this paper, we provide two options for the maximal order of Neyman's test: $k = 4$ as Neyman originally recommended (denoted as Neyman4) and $k = 12$ as the largest option that is numerically stable (denoted as Neyman12). In this paper, we refer the basis function to be Legendre polynomials unless specified otherwise, as that have shown better effectiveness in detecting the leakages among our settings than its trigonometric counterpart.

Algorithm 1 presents the work flow of a leakage detection based on the univariate Neyman's smoothness test. If the measurement data turns out to be discrete, line 6 represents a pre-processing step to "smooth" the data (see Appendix C)[3].

Computation Efficiency. The primary computation load comes from calculating the relative distribution V: as we can see in line 4, this step takes two sortings, and costs $O(mlogm) + O(nlogn)$. Both Line 5 and Line 6 (if exists) can be completed in linear time. In practice, as ψ_i is a fixed function and k is a small constant, the overall cost of line 9 is also $O(m)$. Thus, the overall test can be completed in quasilinear time.

[3]Note that this is the counterpart of the binning process in χ^2-test, when the data is continuous, therefore should not be taken as a disadvantage of this test. Although digital readings of raw measurements are often discretised, any pre-processing (e.g. point combination, signal processing etc.) can make them continuous, therefore it is unfair to claim side-channel usage should always be discrete or continuous.

Algorithm 1: Leakage detection with Neyman's smoothness test

Input: Fix measurement group $(X_1,...,X_n)$ $X \sim F$
Input: Random measurement group $(Y_1,...,Y_m)$ $Y \sim G$
Output: p-value p for $F = G$

Select a maximal order k;
Calculate $V = F_n(Y)$;
begin
  Sort X and Y separately;
  For each Y_j, $V_j = \#\{X_r | X_r \leq Y_j\}/n$;
  (optional) When necessary, smooth V;
end
for $i \leq k$ **do**
  $\overline{\psi_i} = \sum_{j=1}^{m} \psi_i(V_j)/m$;
end
$\Psi = \sqrt{\frac{nm}{n+m}} \max\left(\overline{\psi_1}, ..., \overline{\psi_k}\right)$;
$p = 1 - (2\Phi(\Psi) - 1)^k$;
return p

IV. Simulated Experiments

A. Simulation setup

Throughout this section, we use simulation experiments to demonstrate the performance of Neyman's test. Our testing setup is a d-shared Boolean masking, where each share is always 8-bit. The masking scheme is processed in parallel, so the attacker can observe only one univariate leakage time-sample. The random noise is generated by the Matlab default normal distributed random function. To avoid unnecessary confusion, we list the noise variance instead of the signal-to-noise ratio (SNR). Unless stated explicitly, all following experiments use 0 as the fixed constant.

In order to cover a variety of potentially interesting circumstances, we consider the following options for the leakage function L:

- Standard Hamming weight (HW):
$$L_i(x_i) = \sum_{j=1}^{j=8} x_{i,j}$$
where $x_{i,j}$ is the j-th bit of the i-th share of x
- Least significant bit (LSB) emphasised:
$$L_d(x_d) = 10 \cdot x_{d,1} + \sum_{j=2}^{j=8} x_{d,j}$$
All other L_i-s follow the standard HW model. This is motivated by the fact that sometimes the LSB may dominate the leakage [11].
- Randomised weight HW:
$$L_i(x_i) = \alpha_0 + \sum_{j=1}^{j=8} \alpha_j \cdot x_{i,j}$$
where $\alpha_i \leftarrow \mathcal{U}(0,1)$. This is motivated by the fact that realistic devices often have different weights for different bits [28], [29].
- HW model with adjacent bit-interaction (within one share):
$$L_i(x_i) = \alpha_0 + \sum_{j=1}^{j=7} \alpha_j \cdot x_{i,j} + \sum_{j=1}^{j=8} (\alpha_{j,j+1} \cdot x_{i,j} \cdot x_{i,j+1})$$
This model has been observed in [21], [30]. Note that this is not a security flaw because there is no cross-share interaction (unlike the case in [17]). Therefore, for the target 3-share scheme, we perform only the 3rd order t-test in Figure 1, 2 and 4; whilst present also the 1st/2nd order t-test in Figure 5.
- Affine model:
$$L_i(x_i) = HW(A(x_i))$$
where A is the affine transformation in AES's Sbox. If the underlying masking scheme is a code-based one or the hardware circuit is computing a linear transformation in parallel, then such behaviour would occur in practice.
- Sbox model:
$$L_i(x_i) = HW(S(x_i))$$
where S is the AES's Sbox. This model represents complicated leakage from combinatorial logic in hardware, which is usually highly non-linear.

To ensure the noise levels are comparable, we explicitly multiply a factor to each power model so that their "signal variances" equal to the standard HW model.

B. A priori power analysis

Considering in practice various devices produce various leakage models, we now subject the novel Neyman's test alongside Welch's t-test and the χ^2-test to a battery of repeated experiments (100 repetitions) featuring different leakage models and noise, in the typical use case of a 3-share masking scheme (in both the parallel and the sequential case). We assume a fixed alpha of 0.00001 which is in line with previous work, and ask how many traces are necessary to reach a power of 0.8[4].

1) Results for parallel leakage: We first assume that all shares are processed in parallel and their total sum is leaked. The results are displayed in Figure 1. From the upper-left, we have:

- **Standard HW**: χ^2 works better when the noise is low, whereas Welch's t test becomes better for higher noise. Neyman's test (especially Neyman12) can serve as a fair trade-off option.
- **LSB-emphasised**: The unbalanced weight has an impact on all the tests, although the χ^2-test seems to be the least affected.
- **Randomised weighted HW**: For this particular choice of weights, χ^2-test falls behind Welch's t, even when the noise level is extremely low. Both Neyman4 and Neyman12 works as good as Welch's t in this setup, although the advantage is quite limited.

[4]In a typical figure, this means that 8 out of 10 leaks will be detection.

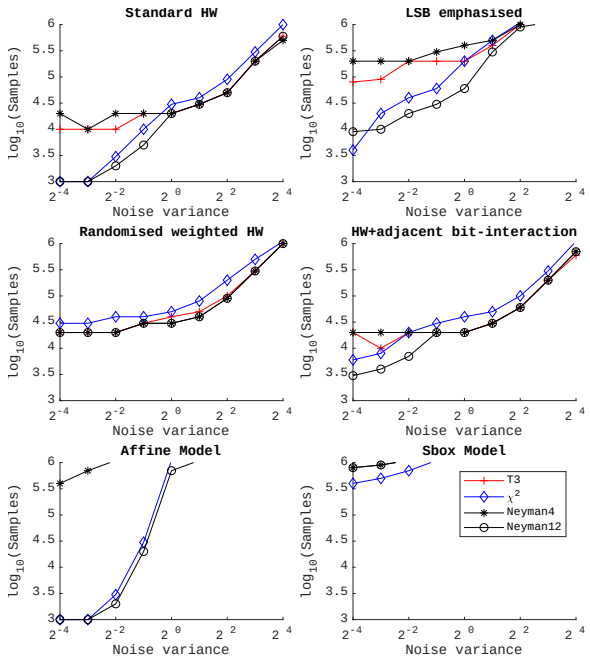

Fig. 1: 3-share Parallel leakage: required sample size for 80% power: χ^2 v.s. 3rd order t v.s. Neyman

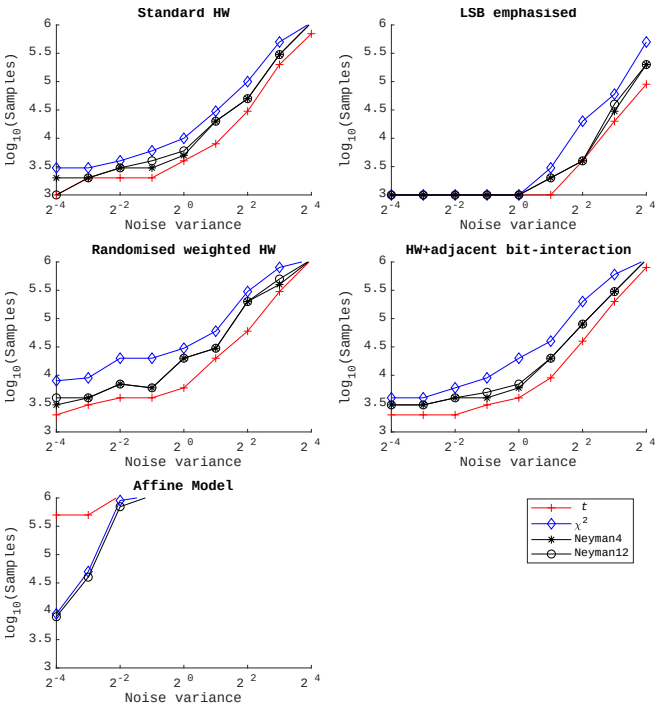

Fig. 2: 3-share Sequential leakage: required sample size for 80% power: χ^2 v.s. 3rd order t v.s. Neyman

- **HW + adjacent bit-interaction**. This model makes detection harder for χ^2 and Neyman12 under a low noise level, but does not have a significant impact when the noise level is high.
- **Affine Model**: Note that a cryptographically linear transformation is only linear on $GF(2^n)$, not on $\mathbb{R}$. Therefore, this affine model actually leads to a quite high-order leakage function L_i. Consequently, the 3rd order Welch's t-test failed completely, while only χ^2 and Neyman12 work with a low level of noise.
- **Sbox Model**: Welch's t (3rd order) failed in this setting. Neyman's test and χ^2, on the other hand, can detect a small difference with little noise. The fact that χ^2 shows an advantage against Neyman12 suggests there might be some useful information in the "high-order components" discarded by Neyman12. Throughout our simulation, only in this relatively extreme case we found such an advantage for χ^2 over Neyman12.

To sum up, for most scenarios, χ^2-test has an advantage in extremely low noise scenarios. Such advantage shrinks quickly with increasing noise, supposing the other tests can actually detect some meaningful difference. Welch's t-test is relatively robust against noise, yet the detecting power is restricted to that moment only. Neyman's test on the other hand, can be adjusted with Neyman's "order": higher order like Neyman12 can capture most (but not all) of the coverage of the χ^2-test, while Neyman4 remains more robust against noise, just like Welch's t-test. Having said that, the leakage model plays a critical role here: in certain implementations, it can possibly push the advantage region of χ^2-test (Welch's t test) to some point with extremely low (high) noise level, which leads to no practical impact at all.

2) Results for sequential leakage: For this experiment we assume that all shares are processed in sequence. In principle this results in multivariate leakage, thus we perform the standard preprocessing step that consists of subtracting the mean from all leakage points and then multiplying them [31]. If the leakage function features some bit-linear components, then this preprocessing has been shown to retain those features, and consequently we expect the t-test to be particularly powerful.

The results of this experiment are displayed in Figure 2: the t-test performs very well, this time even in the affine leakage model setup. In most cases, Neyman's test lies somewhere between the χ^2-test and the t-test. The Sbox model is eliminated here since all tests fail to achieve 80% power.

C. In depth comparison for additive HW leakages

We replicated the comparison in [9] including the Neyman tests for ease of comparison with previous work. This comparison is only based on observations using the HW power model. Our experiments confirm the observation — Welch's t-test performs well with higher noise and less data shares (red rectangle), whereas the χ^2 has advantages when there is limited noise and many data shares (blue rectangle). The χ^2 test and Welch's t-test are complementary to each other, specifically at these two corners. The yellow circle, on the other hand, illustrates the other side of the coin: Welch's t-test is hampered by the larger d and the χ^2 is hampered by a less-favourable noise level (namely their "sour-spot"), whilst Neyman12 found itself a "sweet-spot".

Zoom in for $d = 3$: Although Figure 3 presents a vivid overview of our constructions as well as the previous techniques, identifying small differences from various shades

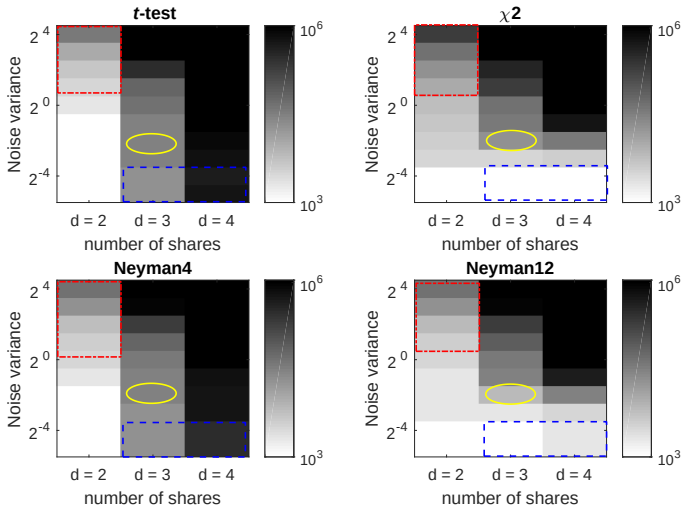

Fig. 3: Required sample size for 99% detection: darker cells require more samples to successfully detect the leakage

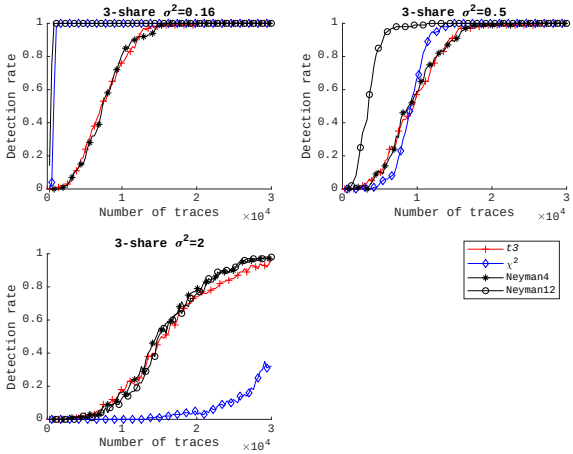

Fig. 4: χ^2 v.s. 3rd order t v.s. Neyman

of grey is arguably challenging. Therefore, we provided a zoomed view for the centre of Figure 3, where $d = 3$ and σ^2 is around 1.

Figure 4 clearly demonstrates that as the noise increases, the testing results are moving from a "χ^2 favourable" direction towards a "t favourable" direction. As expected, there is a sweet-spot within the spectrum: in the middle graph, Neyman12 clearly shows an advantage against both Welch's t-test and the χ^2-test. We would like to stress that the practical impact is much more complicated than Figure 3: for instances, some of the cases where the χ^2-test outperforms all other tests have a extremely low noise which could be unrealistic in practice. Similarly, the "sweet-spots" of Neyman's test showed in Figure 3 may vary depending on the characteristics of the traces to be tested.

D. Detecting "flaws"

Arguably, one of the most essential use cases for leakage detection is identifying implementation "flaws". For typical Boolean masking schemes, this is often caused by some unexpected register/bus transition occurring within the processor [16], as well as unexpected leakage contributed by combinatorial logic and glitches [17].

In a parallel univariate setting, the *de facto* standard is testing all lower than d Welch's t-tests. Moradi et al. suggested this could be another advantage for the χ^2-test, as it can possibly combine various lower order leakages and construct a more efficient statistic [9]. Therefore, we would now illustrate what happens in the case of such a "flawed" leakage function. Specifically, consider a 3-share flawed leakage function L' is:

$$L'(x_1, x_2, x_3) = L_1(x_1) + L_2(x_2) + L_3(x_3)$$
$$+\alpha\left(L_{1,2}(x_1 \cdot x_2) + L_{2,3}(x_2 \cdot x_3) + L_{1,3}(x_1 \cdot x_3)\right)$$
$$+\beta L_{1,2,3}(x_1 \cdot x_2 \cdot x_3)$$

where $\cdot$ stands for the bitwise-and and x_i stands for the i-th share of x. Any cross-share combination produces unexpected leakage that is not captured by the independent assumption (aka a "flaw"). In practice, such interaction terms usually come from physical sources as observed in [19], [20] or micro-architectural effects as observed in [17][5]. Considering the leakage from those interactions often has rather limit magnitude, we set $\alpha = 0.1$ and $\beta = 0.01$ to emulate the practice.

Figure 5 illustrates the impact of having L' as the leakage model. Although L' has an interaction term that combines all 3 shares, as $\beta = 0.01$, first order Welch's t-test never succeed. Second order Welch's t-test, on the other hand, becomes the best option with larger noise ($\sigma^2 = 16$). We observe the similar trend as Figure 4: with lower noise, Neyman12 and χ^2 are better, while Neyman4 stays close to the 3rd order Welch's t-test. However, when the second order t-test becomes a better choice, Neyman4 adaptively "de-couples" with the 3rd order t-test and achieves similar performance as the second order t-test. Neyman12 has similar behaviour: as noise level becomes higher, its performance "smartly de-couples" from the χ^2-test and moves towards the best possible t-test. This phenomenon suggests at least in this particular setting, Neyman's test achieves a better balance, which might be a critical property as leakage detection techniques could face various leakage functions and noise levels in practice.

V. Experiments with Real Device Data

Previous work has extensively focused on the use case of hardware masking. Results from [9] confirm the superior power of the t test whenever there is leakage in a central moment, and the usefulness of the χ^2-test otherwise. These experiments, because derived from hardware masking, thus are the practical counterpart to the case of parallel HW/linear leakage from our simulations.

On the other hand, masking per se is not limited to hardware implementations, and one could argue that the more immediate use case is in fact in software implementations. We therefore, in order to complement the existing work, concentrate on the application of leakage detection tests on software

[5]Combinatorial leakage is a well-recognised component in hardware implementation, see [32] for instance. However, it only becomes a "flaw" if the implementation fails to avoid them to be "crossing-shares".

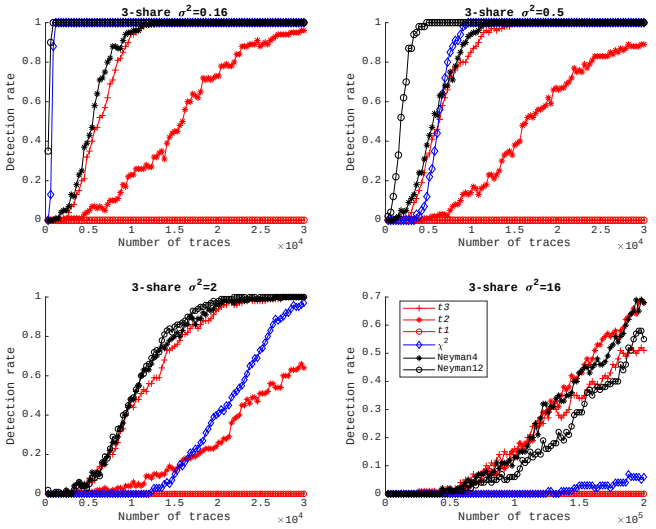

Fig. 5: HW leakage with a second order "flaw": χ^2 v.s. 3rd order t v.s. Neyman

implementations. We study an implementation of a 3-share masking scheme (written in C, which implies that some flaws might occur due to the transition leakage [16]), as well as an implementation of a 4-share masking scheme using "share-slicing" [17] to improve its performance.

All traces in this section were captured with a Picoscope 2206B, running at 250MSa/s. The target core is either an NXP LPC 1114 (ARM Cortex M0), or an NXP LPC 1313 (ARM Cortex M3), both running at 12 MHz.

A. 4-share software masking with share-slicing

Implementing masking incurs a significant performance loss. Thus many techniques have been proposed as mitigation, one of which is to "pack shares" cleverly into processor words. This trick leads to more efficient implementations as it enables executing the algorithm with several shares running in "parallel". However, parallel masking implementations in practice are often found to be not adhering to the necessary independence requirements on the shares. In the case of hardware masking, the coupling effect can produce joint leakage between parallel data shares [19]; for software implementation, not only coupling still persists [20], but also other micro-architectural effects contribute [17]. Arguably, this is exactly what leakage detection is aiming for: finding any unintentional leakage that the designers might not be aware of.

We now analyse the 4-share multiplication of Barthe et al. [33]: the underlying leakage comes from the specific bit-allocation strategy (called "share-slicing" in [17]), regardless of the actual masked multiplication. In a 4-share share-slicing scheme, all 4 shares will be stored in the same register, and this produces small yet exploitable low-order leakages [17].

Using this particular implementation as our target gives us some flexibility. As we can see in Table I, for any secret bit a, the implementation creates 4 random bit shares (a_1, a_2, a_3, a_4) and stores them in a register. For a 32-bit processor, except for $a_1 \oplus a_2 \oplus a_3 \oplus a_4 = a$, the rest 28 bits are up to the specific implementation. These 28 bits allow us to "manipulate" the

TABLE I: Bit-allocation for bit-slicing

Bit no	1	2	3	4	...	29	30	31	32
Stored bit	a_1	a_2	a_3	a_4	...	h_1	h_2	h_3	h_4

noise level as we did in the simulation section, from all 0-s (little noise) to all random bits (large noise). We would remind readers that such modification may also impact the leakage function itself in addition to the noise levels due to the change of the execution course. Meanwhile, whatever stored in these 28 bits will not affect the correctness of the masked encryption: only (a_1, a_2, a_3, a_4) are computing the true masked encryption; other bits are merely some parallel padding data.

For our experiments, we consider the following range of scenarios:

- *Zero-pad* variant. Setting all 28 bits to 0 ($b_1 = b_2 = ... = h_3 = h_4 = 0$), so that the overall noise level is expected to be the lowest.
- *Same-bit-pad* variant. Using all 28 bits as valid data shares, but the unshared secret remains exactly the same ($a_1 \oplus a_2 \oplus a_3 \oplus a_4 = b_1 \oplus b_2 \oplus b_3 \oplus b_4 = ... = h_1 \oplus h_2 \oplus h_3 \oplus h_4$). In practice, this could mean all 8 concurrent blocks are encrypting, yet the attacker can do a chosen-plaintext access, setting all 8 blocks to be the same. The noise level here is higher than the *Zero-pad* variant, yet depending on the specific leakage function, the same unshared value could also amplify the leakage "signal" which leads to easier detections.
- *Random-pad* variant. All 28 bits as completely random bits ($b_1 \oplus b_2 \oplus b_3 \oplus b_4 = r_b, ..., h_1 \oplus h_2 \oplus h_3 \oplus h_4 = r_h$), so that the overall noise level is expected to be the highest.

We use the same implementation code as [34]: for the *zero-pad* variant, we simply erase the other 28 bits before the target multiplication. Since the SNR in this implementation is relatively low, we repeat the same measurement 100 (1000) times and keep the averaged trace to achieve a higher resolution (same as [17], [35]), so that the testing procedure can finish in a reasonable amount of time. Note that this means the following results represent a scenario where the attacker has a much more accurate acquisition setup than ours: there is no guarantee whether an attacker in practice can actually achieve this (or not).

Zero-pad variant. Since all other bits are set to zero, we are testing a relatively low noise scenario[6]: the "sweet-spot" that favours Neyman's test still exists (the left one in Figure 6). Neyman4 confirms the leakage within 20k traces whilst χ^2-test requires double the amount (= 40k traces) to reach the same confidence. Perhaps the even more attractive property of Neyman's test is its ability to cope with both the "high noise, low order" (favours Welch's t-test) scenarios and the "high order, low noise" (favours the χ^2-test) scenarios. Neyman12 is almost overlapping with Neyman4 here: the actual p-value is a bit higher, suggesting the 4-12 order components in Neyman's test (aka $(\psi_5, ..., \psi_{12})$) seems to be redundant in this particular

[6]Unlike the simulation experiments, we cannot completely control the environmental noise here. Thus, the "low" noise only guarantees the algorithmic noise is low: the overall noise is medium if we compared it to Figure 4.

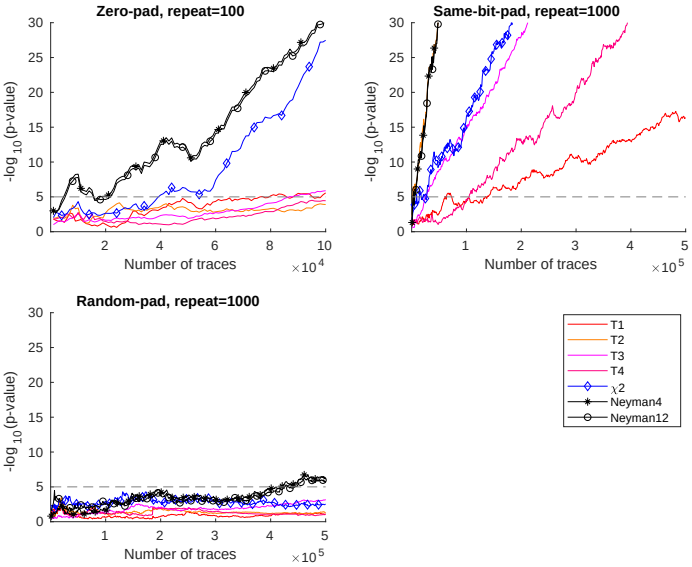

Fig. 6: 4-share share-slicing: χ^2 vs. t-test vs. Neyman

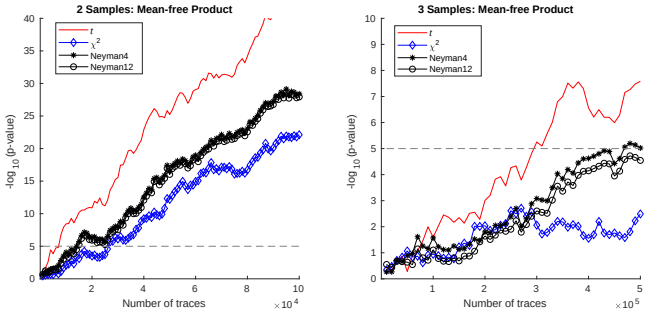

Fig. 7: Sample combination: χ^2 vs. t-test vs. Neyman

case. Nevertheless, the overlapping plot can also be viewed as an evidence proving that the negative impact of having useless high order components is almost negligible.

Same-bit-pad variant. Unlike the *zero-pad* variant, if all other 7 blocks are active, the overall noise level is higher. However, the SNR does not necessarily decrease, as the signal level also becomes higher. The middle graph in Figure 6 shows that the leakage detection result is quite different: the second-order t-test performs quite well in this case, successfully finds the leakage within 20k traces. Both Neyman4 and Neyman12 have similar performance, slightly better than the second order Welch's t-test. The χ^2-test is still effective, although it takes much more traces.

Random-pad variant. Further increasing the noise shows an intriguing phenomenon that did not appear in our simulation: when all other 7 blocks are randomly picked, both Neyman4 and Neyman12 clearly outperform all other tests. In the lower half of Figure 6, both Neyman's tests successfully find the leakage with 450k traces, whereas no other test succeed within 500k traces. One plausible explanation would be, as this implementation is "flawed" (having exploitable differences at the lower-than 4 order statistical moments), Neyman's test starts to combine information from various orders and hence being able to develop an effective statistic in a faster manner. The χ^2-test should eventually detect the leakage as well, although the "capturing everything" property also suggests the test can be less resilient to random noise.

B. 3-share software masking in C

We use an open source higher-order masking implementation [36] with 3 shares. The underlying masking scheme is from Rivain and Prouff [37], using the "common shares" technique [38] and the "random reduction" method[7] by [41] to improve the performance. Our experimental platform is exactly the same as before: the only difference is here we use an ARM Cortex M3 core (NXP LPC1313), which has a larger RAM to support higher order masking computations.

Because the entire implementation is written in C, we expect that a 3-share scheme can be attacked with second order attacks, due to the fact that transition leakage can lower the security order [16]. Similar to Section IV, we search for a pair of samples whose mean-free product demonstrates some leakage. We did find one of such combinations within the Sbox region: since all three shares are proceeding with a table lookup in sequence, there is a leakage sample corresponds to $x_1 \oplus x_2$. Combined with the leakage sample of x_3, a second order test can be built.

In the left half of Figure 7, we can see that if the "mean-free" product combining function is selected, Welch's t-test is clearly the best approach, with Neyman's test follows. This does not come as a surprise as the mean-free product combining function (when the individual points are dominated by HW leakage) results in a function that is strongly correlated to the HW of the sum of the respective intermediate values.

The right half of Figure 7 gives the performance of each test with "mean-free" product combining all three shares: Welch's t-test clearly outperforms the other two, with Neyman's test being second, while χ^2 falls further behind. This is indeed consistent with the previous leakage testing [9] and information theoretic analysis results [42]. As a consequence, if the implementation operate data shares in sequence and we know in advance the leakage is subject to the standard HW model, we recommend to use Welch's t-test. Otherwise, it is recommended to try all possible detection techniques: however, if users insist to use one representative, based on what we learned in Section V-A, Neyman's test seems to be a more balanced choice.

VI. Discussion and Perspectives

Leakage detection can be used as a tool in different contexts. For instance, as a part of ISO 17825, it is used as a conformance testing regime and thus serves the purpose of certifying security. The Federal Information Processing Standard (FIPS 140-3) also includes leakage detection for the same purpose, through adopting various ISO/IEC standards [43]. As a recent analysis has demonstrated, this is an extremely challenging use case—not only the power of tests might be concerning, but also the detection outputs require careful multiplicity

[7]Theoretically speaking, this is a problematic scheme as pointed out by Barthe, Dupressoir and Gregoire: the claimed security proof is not achieved by this approach [39]. We have patched this implementation beforehand according to [40].

corrections [44]. A more realistic use case of leakage detection is that to either demonstrate (or even certify) vulnerability or to identify concrete weaknesses. We now reflect on what can be achieved with the new method, and those we compare it to, with regards to the latter use cases, and then reflect on open research.

A. Verifying Security "Orders"

Moment-based tests may have one advantage: their results have a simple link to the security proof of the underlying masking scheme. Thus for a d-order secure scheme, if any lower than d-th order t-test fails, this fact certifies that the implementation has a "flaw" (either caused by implementation pitfalls or invalid security assumption). Note that the opposite is not necessarily true: if all lower than d-th order t-test find no leakage, it does not guarantee the security assumption is respected because any test is based on a set number of traces. Therefore a test can never prove the absence of a leak, it can only fail to gather enough evidence for the presence of a leak.

Distribution-based tests do not offer a link to the security order of a masking scheme. In fact, as those tests can potentially capture any type of distributional difference, they can possibly detect leakage based on $\geq d$-th order moments. Although Neyman's test has its own definition of "order", it is not different than other distribution based tests: it offers no link to a security order. Thus it is best suited to demonstrate potential vulnerability at a certain trace level. For any more investigative endeavour, a developer best deploys specific tests instead (like is the case for the χ^2-test).

A legitimate question is whether knowing which security order an implementation fails helps to fix the problem. Consider for instance a highly parallel masking scheme: knowing that there is a security flaw at an order below the security proof's guarantee does not help to narrow down what circuit components actually causes the problem. This is perhaps slightly different in a sequential software implementation, where (assuming we know which trace points produce the joint leakage) the presence of a lower order leakage potentially reveals the offending instructions/intermediate values. However, one can argue in this case, leakage reasoning is driven by the fact that trace points can be associated with specific intermediate values, not by the order of the flaw itself. Thus knowing/testing a specific security order alone does not ensure security in practice (see also [10]); nor does it help in explaining the leakage or working out a security patch. Summarising the role of a generic leakage test really is to demonstrate potential vulnerability at a certain trace level (i.e. the statement that results from such a test is that "implementation A shows a potential vulnerability with X traces). Any stronger statement is not possible in a black box scenario: to certify vulnerability additional measure with regards to false positives via sensible multiplicity corrections are also required [44].

B. How to deploy leakage detection in a black box setting

The most intuitive use case for non-specific leakage detection is for an initial exploratory analysis of a new implementation/device. Recall Figure 3 suggests that Welch's t test and the χ^2-test are "complementary" in two orthogonal directions (at least when considering parallel HW leakages). As a trade-off option, Neyman's test "pushes" the white region in Figure 3 by balancing the two directions.

However, when analysing a new implementation that is still "black box", we do not know if it will exhibit strong enough HW leakage: that is to say, we cannot conclude the leakage behaviour will follow the situation in Figure 3 or [9]. When dealing with a new implementation/device it is also completely unclear what the effect size will be, a developer have to "guess" how many traces might/should reveal a flaw. This is a problem that is independent of whichever test we are using.

The most sensible strategy is then to run at least two independent tests simultaneously in an incremental manner until a leakage is identified in traces (controlling the rate of false positives suitably following e.g. [44]). We recommend to always run t-tests on all orders up to the masking order. If the device leakage has some bit-linear leakage in some central moments, then the t-test will reliably identify it with the least number of traces. Otherwise, Neyman's test will be able to identify possible vulnerability, especially if realistic noise level becomes too high for the χ^2-test.

C. Open research

Although there is a plethora of research on leakage detections in the side channel literature, there are still some open questions of fundamental interest left.

- *Discrete vs. Continuous.* This is a rather general challenge in statistics: discrete techniques like χ^2 lose information through the binning process; whereas continuous techniques may have an inaccurate statistical inference for discrete inputs. This is not specific to Neyman's or the χ^2-test: e.g. the Kolmogorov-Smirnov test has exactly the same issue [45]. Optimal binning strategies, or ways to bound the error that is due to binning, remain an open question of practical relevance for our community.
- *Multivariate testing.* Instead of combining multiple samples through pre-processing techniques (sum or product) in the case of sequential leakage, Neyman's test does have a multivariate extension [13], which is computationally expensive. Finding powerful multivariate extensions (with reasonable computation cost) for leakage detection tests is still a largely uninvestigated question (with the exception of [46]).

VII. Conclusion

We proposed to use the Neyman's smoothness test as a supplementary tool for the leakage detection toolkit. The highlight of this approach is that it achieves a valid trade-off between the χ^2-test and Welch's t-test. More specifically, Neyman's test can possibly detect many higher order leakages (compared with Welch's t-test), while demonstrating reasonable power for lower order leakages with considerable noise (compared with the χ^2). Both cases are clearly illustrated in our simulation experiments, while the latter is also confirmed in realistic measurements. We believe Neyman's smoothness test is a good candidate (alongside the classical Welch's t-test) for leakage

detection in black box scenarios because of its good all-round performance.

APPENDIX A
WELCH'S t-TEST AND PEARSON'S χ^2-TEST

A. Welch's t-test

Welch's t-test is a widely adopted method by the community where an early standardisation was proposed by [7]. Let $\mathcal{Q}_f$ and $\mathcal{Q}_r$ be the fixed and random trace sets. Denote n_f and n_r their sample sizes, μ_f and μ_r the sample means and s_f^2 and s_r^2 the sample variances. The t-test statistic and the degrees of freedom v are computed as

$$t = \frac{\mu_f - \mu_r}{\sqrt{\frac{s_f^2}{n_f} + \frac{s_r^2}{n_r}}} \qquad v = \frac{\left(\frac{s_f^2}{n_f} + \frac{s_r^2}{n_r}\right)^2}{\frac{\left(\frac{s_f^2}{n_f}\right)^2}{n_f - 1} + \frac{\left(\frac{s_r^2}{n_r}\right)^2}{n_r - 1}}$$

The p-value can be derived from the quantile function of Student's t-distribution: smaller p-values give evidence to reject the null hypothesis (H_0: the fixed and random group have the same mean) which leads to conclude that a potential leakage is detected. The original proposal of [7] simplified the computation of the quantile function by using the threshold $t = \pm 4.5$ [7], which is close to $p = 10^{-5}$ for a large number of traces.

Note that Welch's t-test relies on the assumption that the underlining $\mathcal{Q}_f$ and $\mathcal{Q}_r$ are normal. Also it is a test of means which indicates it is expected to be insensitive to leakages above the first order. A common practical measure that extends Welch's t-test to a d-th order leakage detection is to apply the same test on traces taken to their d-th order moments.

B. Pearson's χ^2-test

To relax the order specific constraint of Welch's t test , [9] proposed the use of χ^2 for leakage detection. However, a drawback of this method is that the leakage distributions are now required to be presented in a contingency table which inevitably introduces a "data binning" process. In the contingency table, each column corresponds to a bin of data and each row a distribution to be compared. For fixed-vs-random, only the fixed and random distributions of traces are considered. Denote the number of columns as c, the frequencies of the fixed and random traces in the j-th bin as $F_{f,j}$ and $F_{r,j}$ and the total number of samples as N. The χ^2 statistic can be computed as:

$$x = \sum_{j=0}^{c-1} \frac{(F_{f,j} - E_{f,j})^2}{E_{f,j}} + \sum_{j=0}^{c-1} \frac{(F_{r,j} - E_{r,j})^2}{E_{r,j}}$$

where $E_{f,j}$ ($E_{r,j}$) represents the expected frequency of value j in the fixed (random) distribution

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

the p-value can be derived from the quantile function of χ^2 distribution with degree of freedom $c - 1$. As of Welch's t-test, smaller p-values suggest there is a significant difference between the fixed and the random group and thus a leakage is detected.

APPENDIX B
SPECIFIC V.S. NON-SPECIFIC

Historically, as the concept of leakage detection becomes prevalent after the proposal of TVLA [7], we are often suggesting that a statistical test that can possible capture "arbitrary leakage". Admittedly, this is an overly ideal statement that is perhaps, never fully achieved. The "fixed-v.s.-random" strategy used in [7] (or the "fixed-v.s.-fixed" strategy in [8]) detects any kind of data-dependency on the power trace. Although this in theory, does cover all possible arbitrary leakage, it will also report many unexploitable "leakage" such as the leakage from the plaintexts or ciphertexts. Nonetheless, this is an inevitable trade-off for all the so-called non-specific tests today, including Welch's t-test, the χ^2-test and the CMI test [11]. A common misunderstanding is confusing "non-specific" with the concept of "generic distinguisher/attack" [15]. Formally speaking, we often denote the leakage as $L(x)$, where

1) x: A known enumerate intermediate state (e.g. first round Sbox output)
2) L: The function form of L (e.g. in standard HW model, L is the Hamming weight function)

A distinguisher/attack is called "generic" if for a given x, it applies to any kind of function L. Both the NICV [14] and the ρ test [8] achieves this property: however, they cannot capture any leakage that replies on not x, but some other intermediate state y. "Non-specific" tests are aimed at losing the restriction on x rather than on L.

Alternatively, one can also be combined a specific test with the "fixed-v.s.-random" strategy: however, the superiority of these tools is simply lost during such convention (details can be found in [47]).

APPENDIX C
DISCRETE VS. CONTINUOUS

Generally speaking, continuous/discrete distributions usually have their distinct features that benefit some testing methodologies while hinder others. For instance, the χ^2-test is designed for discrete data: if the tested data are continuous, a binning procedure is usually required beforehand. Such process inevitably introduces some information loss[8], depending on the specific binning scheme [9].

On the other hand, using any continuous test on discrete data is not trivial either. Take one sample good-of-fitness Kolmogorov-Smirnov test for instance: if the observed data is highly discrete, adjustment would be necessary to extract meaningful results [45]. Neyman's smoothness here is not an exception: distribution of $F(X)$ will only be uniform, if F is continuous and strictly increasing. Theoretically, in a leakage detection setting, this is not an issue: although the

[8]Note that this is not equivalent to less efficient detections.

leakage itself is very likely discrete, the additional random noise is usually assumed to be Gaussian. As a consequence, the observed leakage will be continuous and strictly increasing on its support.

However, as most of our measurements are from digital oscilloscopes, the traces we get have already gone through an Analog-Digital-Converter(ADC). That is to say, some measurements are already binned to a certain number of bins. This makes them perfect for the χ^2-test, while problematic for Neyman's test: after all, the test compares the distribution of V to an ideal uniform distribution, not a binned one.

One option is to use the categorised version of Neyman's test [48]: in fact, the χ^2-test itself can be regarded as a special case of categorised Neyman's test. Unfortunately, we have found limited literatures on this topic: none of them covers the two-sample test, only the goodness-of-fit test.

The other option, would be as Algorithm 1 suggested, "smooth" V so that Neyman's test won't detect any "binning errors". Of course, these work-arounds seem ad-hoc and might also affect the statistical inference a little bit. Nonetheless, in our realistic experiments, they did not become a significant issue.

- Adding noise. A trivial way to solve this issue, would be adding a small random noise (preferably smaller than the bin size so that it won't convolute the useful signal): the additional noise would again makes the binned data "continuous" again and avoid the "jumps" on the histogram.
- Interpolation. Or, one can try uniformly interpolate on within one bin, making all elements in this bin uniformly distributed on the width of that bin.

Both have been tested in our experiments: both solves the discrete issue, while not introducing significant bias on the testing results. If the interpolation is selected and there are a lot of elements in that bin, one can also use the Riemann sum as a speed-up approximation. More specifically, if the bin between $[V_a, V_b]$ has $h_y(V_a)$ elements, the sum of interpolation can be approximated by

$$\sum_{t=0}^{t=h_y(a)-1} \psi_i(V_a + \Delta_v * t) \approx \frac{\int_{V_a}^{V_b} \psi_i(x)\ dx}{\Delta_v}, \quad \Delta_v = \frac{V_b - V_a}{h_y(V_a)}$$

Although this formula brings integration calculation, in a discrete setting, there are only limited possible values for V. Thus, the integration part can be pre-computed as a look-up table. In the meantime, there is no need to repetitively compute ψ_i any more.

REFERENCES

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, 1999, pp. 388–397.

[2] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, 1996, pp. 104–113.

[3] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic Analysis: Concrete Results," in *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, 2001, pp. 251–261.

[4] S. Chari, J. R. Rao, and P. Rohatgi, "Template Attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, 2002, pp. 13–28.

[5] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," in *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, 2004, pp. 16–29.

[6] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual Information Analysis," in *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, 2008, pp. 426–442.

[7] B. J. Gilbert Goodwill, J. Jaffe, P. Rohatgi *et al.*, "A testing methodology for side-channel resistance validation," in *NIST non-invasive attack testing workshop*, vol. 7, 2011, pp. 115–136.

[8] F. Durvaux and F. Standaert, "From Improved Leakage Detection to the Detection of Points of Interests in Leakage Traces," in *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*, ser. Lecture Notes in Computer Science, M. Fischlin and J. Coron, Eds., vol. 9665. Springer, 2016, pp. 240–262.

[9] A. Moradi, B. Richter, T. Schneider, and F. Standaert, "Leakage Detection with the x2-Test," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 1, pp. 209–237, 2018.

[10] F. Standaert, "How (Not) to Use Welch's T-Test in Side-Channel Security Evaluations," in *Smart Card Research and Advanced Applications, 17th International Conference, CARDIS 2018, Montpellier, France, November 12-14, 2018, Revised Selected Papers*, 2018, pp. 65–79.

[11] L. Mather, E. Oswald, J. Bandenburg, and M. Wójcik, "Does My Device Leak Information? An a priori Statistical Power Analysis of Leakage Detection Tests," in *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, 2013, pp. 486–505.

[12] J. Neyman, "Smooth test for goodness of fit," *Scandinavian Actuarial Journal*, vol. 1937, pp. 149–199, 1937.

[13] W.-X. Zhou, C. Zheng, and Z. Zhang, "Two-sample smooth tests for the equality of distributions," *Bernoulli*, vol. 23, no. 2, p. 951989, May 2017.

[14] S. Bhasin, J. Danger, S. Guilley, and Z. Najm, "Side-channel leakage and trace compression using normalized inter-class variance," in *HASP 2014, Hardware and Architectural Support for Security and Privacy, Minneapolis, MN, USA, June 15, 2014*, R. B. Lee and W. Shi, Eds. ACM, 2014, pp. 7:1–7:9.

[15] C. Whitnall, E. Oswald, and F. Standaert, "The Myth of Generic DPA...and the Magic of Learning," in *Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings*, 2014, pp. 183–205.

[16] J. Balasch, B. Gierlichs, V. Grosso, O. Reparaz, and F. Standaert, "On the Cost of Lazy Engineering for Masked Software Implementations," in *Smart Card Research and Advanced Applications - 13th International Conference, CARDIS 2014, Paris, France, November 5-7, 2014. Revised Selected Papers*, 2014, pp. 64–81.

[17] S. Gao, B. Marshall, D. Page, and E. Oswald, "Share-slicing: Friend or Foe?" *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, pp. 152–174, Nov. 2019.

[18] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Higher-order threshold implementations," in *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, ser. Lecture Notes in Computer Science, P. Sarkar and T. Iwata, Eds., vol. 8874. Springer, 2014, pp. 326–343.

[19] T. D. Cnudde, M. Ender, and A. Moradi, "Hardware Masking, Revisited," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 2, pp. 123–148, 2018.

[20] I. Levi, D. Bellizia, and F. Standaert, "Reducing a Masked Implementation's Effective Security Order with Setup Manipulations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 2, pp. 293–317, Feb. 2019.

[21] D. McCann, E. Oswald, and C. Whitnall, "Towards Practical Tools for Side Channel Aware Software Engineering: 'Grey Box' Modelling for Instruction Leakages," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, 2017, pp. 199–216.

[22] I. Buhan, L. Batina, Y. Yarom, and P. Schaumont, "SoK: Design Tools for Side-Channel-Aware Implementions," 2021.

[23] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses, Third Edition*, ser. Springer texts in statistics. Springer, 2008.

[24] A. K. Bera and A. Ghosh, "Neyman's smooth test and its use in econometrics," 2001.

[25] C. R. Shalizi, "Advanced Data Analysis from an Elementary Point of View." [Online]. Available: https://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/

[26] A. K. Bera, A. Ghosh, and Z. Xiao, "A Smooth Test for the Equality of Distributions," *Econometric Theory*, vol. 29, no. 2, pp. 419–446, 2013.

[27] J. Fan, "Test of significance based on wavelet thresholding and neyman's truncation," *Journal of the American Statistical Association*, vol. 91, no. 434, pp. 674–688, 1996.

[28] W. Schindler, K. Lemke, and C. Paar, "A stochastic model for differential side channel cryptanalysis," in *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, ser. Lecture Notes in Computer Science, J. R. Rao and B. Sunar, Eds., vol. 3659. Springer, 2005, pp. 30–46.

[29] A. Heuser, W. Schindler, and M. Stöttinger, "Revealing side-channel issues of complex circuits by enhanced leakage models," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2012, pp. 1179–1184.

[30] M. Akkar, R. Bevan, P. Dischamp, and D. Moyart, "Power Analysis, What Is Now Possible..." in *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, 2000, pp. 489–502.

[31] E. Prouff, M. Rivain, and R. Bevan, "Statistical Analysis of Second Order Differential Power Analysis," *IEEE Trans. Computers*, vol. 58, no. 6, pp. 799–811, 2009.

[32] F. D. Santis, M. Kasper, S. Mangard, G. Sigl, O. Stein, and M. Stöttinger, "On the relationship between correlation power analysis and the stochastic approach: An ASIC designer perspective," in *Progress in Cryptology - INDOCRYPT 2013 - 14th International Conference on Cryptology in India, Mumbai, India, December 7-10, 2013. Proceedings*, ser. Lecture Notes in Computer Science, G. Paul and S. Vaudenay, Eds., vol. 8250. Springer, 2013, pp. 215–226.

[33] G. Barthe, F. Dupressoir, S. Faust, B. Grégoire, F. Standaert, and P. Strub, "Parallel Implementations of Masking Schemes and the Bounded Moment Leakage Model," in *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, 2017, pp. 535–566.

[34] S. Gao, "Share-sliced AES implementation with 16-bit Thumb instructions," https://github.com/sca-research/ShareSlicing_AES.

[35] A. Journault and F. Standaert, "Very High Order Masking: Efficient Implementation and Security Evaluation," in *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, 2017, pp. 623–643.

[36] F. Hemsworth, "Higher-order Masking of AES-128 based on the Rivain and Prouff method, CPRR method and Common Shares with Random Reduction method," https://github.com/knarfrank/Higher-Order-Masked-AES-128.

[37] M. Rivain and E. Prouff, "Provably secure higher-order masking of AES," in *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, 2010, pp. 413–427.

[38] J. Coron, A. Greuet, E. Prouff, and R. Zeitoun, "Faster Evaluation of SBoxes via Common Shares," in *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, 2016, pp. 498–514.

[39] G. Barthe, F. Dupressoir, and B. Grgoire, "A Note on 'Further Improving Efficiency of Higher-Order Masking Scheme by Decreasing Randomness Complexity'," Cryptology ePrint Archive, Report 2017/1053, 2017.

[40] S. Qiu, R. Zhang, Y. Zhou, and W. Cheng, "Corrections to "Further Improving Efficiency of Higher-Order Masking Schemes by Decreasing Randomness Complexity"," Cryptology ePrint Archive, Report 2017/1244, 2017.

[41] R. Zhang, S. Qiu, and Y. Zhou, "Further Improving Efficiency of Higher Order Masking Schemes by Decreasing Randomness Complexity," *IEEE Trans. Information Forensics and Security*, vol. 12, no. 11, pp. 2590–2598, 2017.

[42] F. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, and S. Mangard, "The world is not enough: Another look on second-order DPA," in *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, 2010, pp. 112–129.

[43] N. Information Technology Laboratory, "SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES," https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf.

[44] C. Whitnall and E. Oswald, "A Critical Analysis of ISO 17825 ('Testing Methods for the Mitigation of Non-invasive Attack Classes Against Cryptographic Modules')," in *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, 2019, pp. 256–284.

[45] G. E. Noether, "Note on the kolmogorov statistic in the discrete case," *Metrika*, vol. 7, pp. 115–116, 1963.

[46] O. Bronchain, T. Schneider, and F.-X. Standaert, "Multi-Tuple Leakage Detection and the Dependent Signal Issue," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 2, pp. 318–345, Feb. 2019.

[47] C. Whitnall and E. Oswald, "A Cautionary Note Regarding the Usage of Leakage Detection Tests in Security Evaluation," Cryptology ePrint Archive, Report 2019/703, 2019.

[48] J. C. W. Rayner and D. J. Best, "Smooth Tests of Goodness of Fit: An Overview," *International Statistical Review / Revue Internationale de Statistique*, vol. 58, no. 1, pp. 9–17, 1990.