From Improved Leakage Detection to the Detection of Points of Interests in Leakage Traces

François Durvaux(✉) and François-Xavier Standaert

ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Louvain-la-neuve, Belgium
francois.durvaux@gmail.com, fstandae@uclouvain.be

Abstract. Leakage detection usually refers to the task of identifying data-dependent information in side-channel measurements, independent of whether this information can be exploited. Detecting Points-Of-Interest (POIs) in leakage traces is a complementary task that is a necessary first step in most side-channel attacks, where the adversary wants to turn this information into (e.g.) a key recovery. In this paper, we discuss the differences between these tasks, by investigating a popular solution to leakage detection based on a t-test, and an alternative method exploiting Pearson's correlation coefficient. We first show that the simpler t-test has better sampling complexity, and that its gain over the correlation-based test can be predicted by looking at the Signal-to-Noise Ratio (SNR) of the leakage partitions used in these tests. This implies that the sampling complexity of both tests relates more to their implicit leakage assumptions than to the actual statistics exploited. We also put forward that this gain comes at the cost of some intuition loss regarding the localization of the exploitable leakage samples in the traces, and their informativeness. Next, and more importantly, we highlight that our reasoning based on the SNR allows defining an improved t-test with significantly faster detection speed (with approximately 5 times less measurements in our experiments), which is therefore highly relevant for evaluation laboratories. We finally conclude that whereas t-tests are the method of choice for leakage detection only, correlation-based tests exploiting larger partitions are preferable for detecting POIs. We confirm this intuition by improving automated tools for the detection of POIs in the leakage measurements of a masked implementation, in a black box manner and without key knowledge, thanks to a correlation-based leakage detection test.

1 Introduction

Leakage detection tests have recently emerged as a convenient solution to perform preliminary (black box) evaluations of resistance against side-channel analysis. Cryptography Research (CRI)'s non specific (fixed vs. random) t-test is a popular example of this trend [4,10]. It works by comparing the leakages of a cryptographic (e.g. block cipher) implementation with fixed plaintexts (and key)

M. Fischlin and J.-S. Coron (Eds.): EUROCRYPT 2016, Part I, LNCS 9665, pp. 240–262, 2016.
DOI: 10.1007/978-3-662-49890-3_10

to the leakages of the same implementation with random plaintexts (and fixed key)[1], thanks to Welch's t-test [38]. Besides their conceptual simplicity, the main advantage of such tests, that were carefully discussed in [18], is their low *sampling complexity*. That is, by comparing only two (fixed vs. random) classes of leakages, one reduces the detection problem to a simpler estimation task. In this paper, we want to push the understanding of leakage detection one step further, by underlining more precisely its pros and cons, and clarifying its difference with the problem of detecting Points-Of-Interest (POIs) in leakage traces. As clear from [9], those two problems are indeed related, and one can also exploit t-tests for the detection of POIs in leakage traces. So as for any side-channel analysis, the main factor influencing the intuitions that one can extract from leakage detection is the implicit assumptions that we make about the partitioning of the leakages (aka leakage model). Our contributions in this respect are threefold.

First, we notice that CRI's fixed vs. random t-test is one extreme in this direction (since it relies on a partitioning in two classes), which is reminiscent of Kocher's single-bit Differential Power Analysis (DPA) [14]. For comparison purposes, we therefore start by specifying an alternative leakage detection test based on the popular Correlation Power Analysis (CPA) distinguisher [3]. The resulting ρ-test directly derives from the hypothesis tests for CPA provided in [16], and relies on a partitioning into 2^s classes, where s is the bitsize of the fixed portion of plaintext in the test. We then compare the t-test and ρ-test approaches, both in terms of sampling complexity and based on their *exploitability*.[2] That is, does a positive answer to leakage detection imply exploitable leakage, and does a negative answer to leakage detection imply no exploitable leakage? Our experimental analysis based on real and simulated data leads to the following observations:

- First, the *sampling complexity* of the t-test is (on average) lower than the one of the ρ-test, as previously hinted [10,18]. Interestingly, we show that the sampling complexity ratio between the two tests can be simply approximated as a function of a Signal-to-Noise Ratio (SNR) for the leakage partition used in these tests. This underlines that the difference between the tests is mainly due to their different leakage assumptions, i.e. is somewhat independent of statistics used (backing up the conclusions of [17] for "standard DPA attacks").
- Second, the *exploitability* of the tests is quite different. On the one hand, leakages that are informative (and therefore can be detected with the ρ-test) but cannot be detected with the t-test are easy to produce (resp. can be observed in practice). Take for example a fixed class of which the mean leakage is the same as (resp. close to) the mean leakage of the random class. On the

[1] The Test Vector Leakage Assessment methodology in [4,10] includes other options, e.g. non specific semi-fixed vs. random tests and specific tests – we focus on this non specific fixed vs. random test that is the most popular in the literature [2,18].

[2] One could also compare the computational complexity of the tests. Since they are based on simple statistics, we will assume that both the t-test and ρ-test can be implemented efficiently. Besides, a minor advantage of the ρ-test is that it can be implemented in a known-plaintexts scenario (vs. chosen-plaintext for the t-test).

other hand, the fixed vs. random t-test leads to the detection of many time samples spread around the complete leakage traces. Hence, not all of these samples can be exploited in a standard DPA (because of the diffusion within the cipher).

Concretely, these observations refine the analysis in [18], where it was argued that leakage detection is a useful preliminary to white box (worst-case) security evaluations such as advertized in [34]. This is indeed the case. Yet, certain leakage detection tests are more connected with the actual security level of a leaking implementation. In this respect, the fixed vs. random t-test is a more efficient way to perform leakage detection only. And the minor drawback regarding its unability to detect certain leakages (e.g. our example with identical means) is easily mitigated in practice, by running the test on large enough traces, or for a couple of keys (as suggested in [4,10]). By contrast, the main price to pay for this efficiency is a loss of intuition regarding (*i*) the localisation of the leakage samples that are exploitable by standard DPA, and (*ii*) the complexity of a side-channel attack taking advantage of the leakage samples for which the detection test is positive. As a result, the ρ-test can be viewed as a perfect complement, since it provides these intuitions (at the cost of higher sampling complexity).

Second, we show that our reasoning based on the SNR not only allows a better statistical understanding of leakage detection, but can also lead to more efficient t-tests. Namely, it directly suggests that if the evaluator's goal is to minimize the number of samples needed to detect data-dependent information in side-channel measurements, considering a partitioning based on two fixed plaintexts (rather than one fixed and one random plaintext) leads to significantly faster detection speeds. This is both due to an improved signal (since when integrated over large execution times, samples with large differences between the two fixed classes will inevitably occur) and a reduced noise (since the random class in CRI's t-test implies a larger algorithmic noise that is cancelled in our proposal). We also confirm these intuitions experimentally, with two representative AES implementations: an 8-bit software one and a 128-bit hardware one. In both cases, we exhibit detections with roughly 5 times less measurements than when using the previous fixed vs. random non specific t-test. We believe these results are highly relevant to evaluation laboratories since (*i*) they lead to reductions of the measurement cost of a leakage detection by a large factor (whereas improvements of a couple of percents are usually considered as significant in the side-channel literature), and (*ii*) they imply that a device for which no leakages have been detected with one million measurements using a fixed vs. random t-test could in fact have detectable leakages with 200,000 (or even less) measurements.

These observations lead to the last contribution of the paper. That is, when extending leakage detection towards the detection of POIs, the ρ-test naturally gains additional interest, since it provides more intuitions regarding the exploitable samples in side-channel traces. More precisely, it allows a better selection of POIs based on the criteria that these POIs depend on an enumerable part of the key. It also maximizes the SNR metric that can be easily connected to the worst-case complexity of standard DPA attacks [5]. Therefore, and more

concretely, our results directly imply that the automated tools for the detection of POIs recently proposed in [7] are also applicable in a fully black box setting, without any key knowledge, by simply adapting the objective function used in their optimization (i.e. replacing it by the ρ-test in this paper). We finally confirm this claim with an experimental evaluation, in the context of first-order secure masked implementations. Doing so, we put forward that the detection of a threshold for which an improvement of the objective function is considered as significative in the optimizations of [7] is made easier when using the ρ-test. We also improve the latter methods by adapting the objective function to the multivariate case and taking advantage of cross-validation to evaluating it.

2 Background

2.1 Measurement Setups

Most of our experiments are based on measurements of an AES Furious implementation (http://point-at-infinity.org/avraes/) run by an 8-bit Atmel ATMega644P microcontroller, at a 20 MHz clock frequency. We monitored the voltage variations across a 22 Ω resistor introduced in the supply circuit of our target chip. Acquisitions were performed using a Lecroy HRO66ZI oscilloscope running at 200 MHz and providing 8-bit samples. In each of our evaluations, the 128-bit AES master key remains the same for all the measurements and is denoted as $\kappa = s_0||s_1||\ldots||s_{15}$, where the s_i's represent the 16 key bytes. When evaluating the fixed vs. random t-test, we built sets of 2000 traces divided in two subsets of 1000 traces each, one corresponding to a fixed plaintext and key, the other corresponding to random plaintexts and a fixed key, next denoted as $\mathcal{L}_f$ and $\mathcal{L}_r$ respectively. When evaluating the correlation-based test, we built a single set of 2000 traces $\mathcal{L}$, corresponding to random plaintexts and a fixed key. In the following, we denote the encryption traces obtained from a plaintext p including the target byte x under a key κ including the subkey s as: $\mathsf{AES}_{\kappa_s}(p_x) \rightsquigarrow l_y$ (with $y = x \oplus s$). Whenever accessing the points of these traces, we use the notation $l_y(\tau)$ (with $\tau \in [1; 20\,000]$, typically). These different subscripts and indexes will be omitted when not necessary. In Sect. 5, we additionally consider a hardware implementation of the AES of which the design is described in [13]. The same amount of measurement as for the previous Atmel case were taken, based on a prototype chip embedding an AES core with a 128-bit architecture requiring 11 cycles per encryption, implemented in a 65-nanometer low power technology, running at 60 MHz and sampled at 2 GHz. Eventually, Sect. 6 considered masked implementation of the AES in our Atmel microcontroller, based on precomputed table lookups [25,31]. For every pair of input/output masks (m, q), it-precomutes an S-box S^* such that $\mathsf{S}^*(x \oplus s \oplus m) = \mathsf{S}(x \oplus s) \oplus q$. This pre-computation is part of the adversary's measurements, which leads to quite large traces with $30,000$ samples. In this last case, we used an evaluation set with 256×50 traces in total, i.e. 50 per fixed value of the target key byte.

2.2 CPA Distinguisher

Our correlation-based leakage detection test will be based on the Correlation Power Analysis (CPA) distinguisher [3], extended to a profiled setting. In this case, and for each time sample τ, the evaluator starts by estimating a model for his target intermediate variable Y from N_p profiling traces: $\hat{\mathsf{model}}_\tau(Y) \leftarrow \mathcal{L}_p$. This model corresponds to the mean leakages associated with the different values of Y. He then estimates the correlation between measured leakages and modeled leakages. In our AES example, it would lead to $\hat{\rho}(L_Y(\tau), \hat{\mathsf{model}}_\tau(Y))$. In practice, this estimation is performed by sampling (i.e. measuring) a set of N_t test traces from the leakage distribution, that we denote as $\mathcal{L}_t$ (with $\mathcal{L}_p \perp\!\!\!\perp \mathcal{L}_t$).

2.3 Fixed vs. Random Leakage Detection Test

CRI's fixed vs. random t-test essentially works by comparing the leakages corresponding to the fixed and random sets of traces defined in Sect. 2.1. For this purpose, and for each sample, one simply has to estimate and compare two mean values. The first one, denoted as $\hat{\mu}_f(\tau)$, corresponds to the samples in the fixed set of traces $\mathcal{L}_f$. The second one, denoted as $\hat{\mu}_r(\tau)$, corresponds to the samples in the random set of traces $\mathcal{L}_f$. Intuitively, being able to distinguish these two mean values indicates the presence of data-dependencies in the leakages. For this purpose, and in order to determine whether some difference observed in practice is meaningful, Welch's t-test is applied (which is a variant of Student's t-test that considers different variances and sample size for the sets $\mathcal{L}_f$ and $\mathcal{L}_r$). The statistic to be tested is defined as:

$$\Delta(\tau) = \frac{\hat{\mu}_f(\tau) - \hat{\mu}_r(\tau)}{\sqrt{\frac{\hat{\sigma}_f^2(\tau)}{N_f} + \frac{\hat{\sigma}_r^2(\tau)}{N_r}}},$$

where $\hat{\sigma}_f^2(\tau)$ (resp. $\hat{\sigma}_r^2(\tau)$) is the estimated variance over the N_f (resp. N_r) samples of $\mathcal{L}_f$ (resp. $\mathcal{L}_r$). Its p-value, i.e. the probability of the null hypothesis which assumes $\Delta(\tau) = 0$, can be computed as follows:

$$p = 2 \times (1 - \mathsf{CDF}_\mathsf{t}(|\Delta(\tau)|, \nu)),$$

where CDF_t is the cumulative function of a Student's t distribution, and ν is its number of freedom degrees, which is derived from the previous means and variances as: $\nu = (\hat{\sigma}_f^2/N_f + \hat{\sigma}_r^2/N_r)/[(\hat{\sigma}_f^2/N_f)/(N_f - 1) + (\hat{\sigma}_r^2/N_r)/(N_r - 1)]$. Intuitively, the value of ν is proportional to the number of samples N_f and N_r. When increasing, Student's t distribution gets closer to a normal distribution $\mathcal{N}(0, 1)$.

3 A Correlation-Based Leakage Detection Test

We start by describing an alternative leakage detection test based on the CPA distinguisher, inspired from the hypothesis test described in [16], and further

taking advantage of the cross-validation techniques recently introduced in [6]. For k-fold cross–validation, the set of acquired traces $\mathcal{L}$ is first split into k (non overlapping) sets $\mathcal{L}^{(i)}$ of approximately the same size. We then define the profiling sets $\mathcal{L}_p^{(j)} = \bigcup_{i \neq j} \mathcal{L}^{(i)}$ and the test sets $\mathcal{L}_t^{(j)} = \mathcal{L} \setminus \mathcal{L}_p^{(j)}$. Based on these notations, our ρ-test is defined as follows, for a target plaintext byte variable X. First, and for each cross-validation set j with $1 \leq j \leq k$, a model is estimated: $\hat{\mathsf{model}}_\tau^{(j)}(X) \leftarrow \mathcal{L}_p^{(j)}$. For s-bit plaintext bytes, this model corresponds to the sample means of the leakage sample τ corresponding to each value of the plaintext byte, i.e. $\hat{\mu}_x^{(j)}(\tau)$.[3] Next, the correlation between this model and the leakage samples in the test sets $\mathcal{L}_t^{(j)}$ is computed as follows:

$$\hat{r}^{(j)}(\tau) = \hat{\rho}(L_X^{(j)}(\tau), \hat{\mathsf{model}}_\tau^{(j)}(X)).$$

The k cross-validation results $\hat{r}^{(j)}(\tau)$ can then be averaged in order to get a single (unbiased) result $\hat{r}(\tau)$ obtained from the full measurement set $\mathcal{L}$. Following, and as in [16], Fisher's z-transformation is applied to obtain:

$$\hat{r}_z(\tau) = \frac{1}{2} \times \mathsf{ln}\left(\frac{1 + \hat{r}(\tau)}{1 - \hat{r}(\tau)}\right).$$

By normalizing this value with the standard deviation $\frac{1}{\sqrt{N-3}}$, where N is the size of the evaluation set $\mathcal{L}$, we obtain a sample that can be (approximately) interpreted according to a normal distribution $\mathcal{N}(0, 1)$. This allows us to compute the following p-value for a null hypothesis assuming no correlation:

$$p = 2 \times (1 - \mathsf{CDF}_{\mathcal{N}(0,1)}(|\hat{r}_z(\tau)|)),$$

where $\mathsf{CDF}_{\mathcal{N}(0,1)}$ is the cumulative function of a standard normal distribution. Besides exploiting cross-validation (which allows us to obtain unbiased estimates for Pearson's correlation coefficient), the main difference between this test and the hypothesis test in [16] is that our model is built based on a plaintext byte rather than a key-dependent intermediate value. This allows us to implement it in a black box manner and without key knowledge, just as the previous t-test.

4 Experimental Results

In order to discuss the pros and cons of the two previous leakage detection test, we now consider various experimental results. We start with a simulated setting which allows us to control all the parameters of the leakages to detect, in order to discuss the sampling complexity of both methods. Next, we analyze actual leakage traces obtained from the measurement setup described in Sect. 2.1, which allows us to put forward the intuitions provided by the t-test and ρ-test regarding the time localization of the informative samples in our traces.

[3] If there is no available trace for a given value of x, which happens when the evaluation set is small, the model takes the mean leakage taken over all the traces in $\mathcal{L}_p^{(j)}$.

4.1 Simulated Experiments

We define a standard simulated setting for the leakages of a block cipher, where an intermediate computation $z = \mathsf{S}(y = x \oplus s)$ is performed, with S an 8-bit S-box. It gives rise to a (multivariate) leakage variable of the form:

$$L_X = [\mathsf{HW}(X) + R_1,\ \mathsf{HW}(Y) + R_2,\ \mathsf{HW}(Z) + R_3],$$

where HW is the Hamming weight function, R_1, R_2 and R_3 are Gaussian distributed random noises with mean 0 and variance σ_n^2, and the index X recalls that in our detection setup, the evaluator only varies the plaintext. For t-tests, the set $\mathcal{L}_f$ contains leakages corresponding to fixed values of x, y or z, denoted as x_f, y_f, z_f, while the set $\mathcal{L}_r$ corresponds uniformly random x's, y's or z's. For ρ-tests, the leakages all correspond to uniformly random x's, y's or z's.

Concretely, we analyzed the t-test based on the third sample of L_X (which corresponds to the target intermediate value z), and for different fixed values of this intermediate value. This choice is naturally motivated by the counter-example given in introduction. That is, since the average leakage of the random class equals 4 in our simulation setting, a fixed class such that $\mathsf{HW}(z_f) = 4$ should not lead to any detection. And extending this example, the bigger the difference between $\mathsf{HW}(z_f)$ and 4, the easier the detection should be.

In parallel, we investigated the ρ-test for the same sample in two cases. First the realistic case, where the model estimation using k-fold cross-validation described in Sect. 3 is applied (using a standard $k = 10$). Second, a theoretical simplification where we assume that the evaluator knows the perfect (Hamming weight) model, which implies that all the samples in the set $\mathcal{L}$ are directly used to compute a single estimate for the correlation $\hat{r}(\tau) = \hat{\rho}(L_X(\tau), \mathsf{model}_\tau(X))$.

The results of our experiments are given in Fig. 1, where the upper part corresponds to a noise variance $\sigma_n^2 = 50$ and the lower part to a noise variance $\sigma_n^2 = 100$. In both cases, we set the detection threshold to 5, which is the value suggested in [2]. They allow the following relevant observations.

(1) *On the impact of the noise.* As doubling the noise variance generally doubles the measurement complexity of a side-channel attack, it has the same impact on the sample complexity of a leakage detection test. For example, detecting a difference between a fixed class such that $\mathsf{HW}(z_f) = 2$ and a random class with the t-test requires ≈ 1300 traces in the upper part of the figure and ≈ 2600 traces in its lower part. Similar observations hold for all the tests.
(2) *On the impact of the fixed value for the t-test.* As expected, for both σ_n^2, a fixed class such that $\mathsf{HW}(z_f) = 4$ cannot be distinguished at all from the random class (since they have the same mean). By contrast, a fixed class such that $\mathsf{HW}(z_f) = 0$ is extremely fast to distinguish from the random class.
(3) *The ρ-test can have (much) larger sampling complexity.* This naturally depends on the fixed value for the t-test. But assuming that several samples from a trace are used in a the leakage detection (which is usually the case, as will be shown in our following measured experiments), there should be some of them that lead to faster leakage detection with the t-test than with the ρ-test.

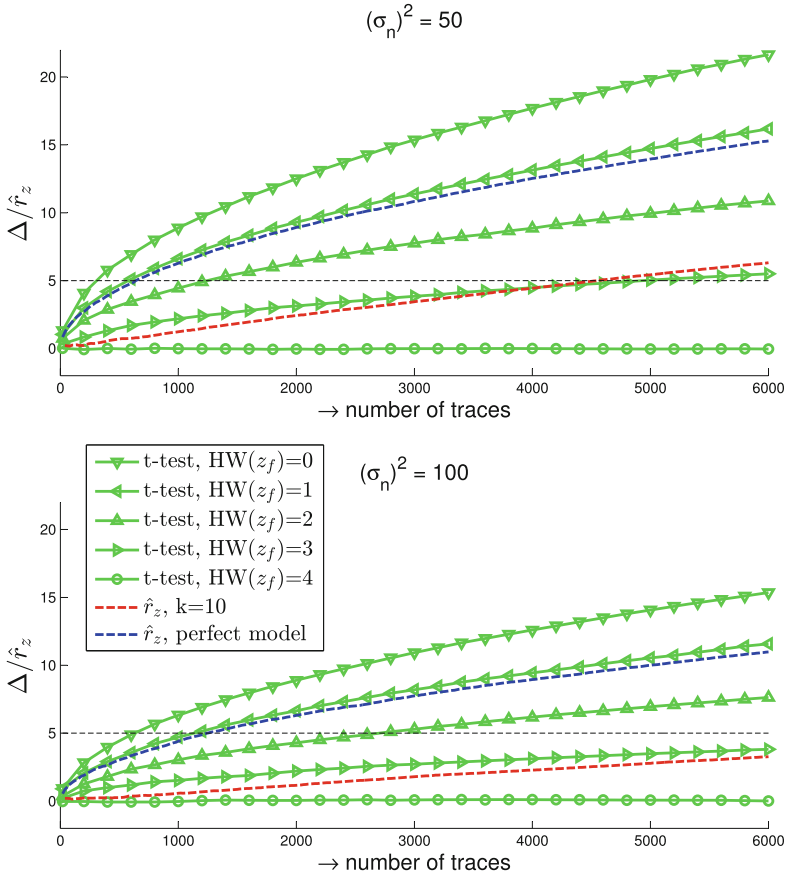

Fig. 1. Leakage detection on simulated traces, Hamming weight leakage function.

(4) *It's all in the SNR.* Most importantly, and just as in standard DPA, the sampling complexity of a detection test essentially depends on the SNR of its leakage partitioning. For the ρ-test, we can directly exploit Mangard's definition from CT-RSA 2004 for this purpose [15]. That is, the signal corresponds to the variance of the random variable $\mathsf{HW}(Z)$ with Z uniform, which equals 2 for 8-bit values, and the noise variance equals to σ_n^2. As for the t-test, we need to define an binary random variable B that is worth $\mathsf{HW}(z_f)$ with probability 1/2 and $\mathsf{HW} = 4$ with probability 1/2. For each value of the fixed z_f, the signal then corresponds to the variance of B, and the noise variance equals to σ_n^2 for the fixed class, and $\sigma_n^2 + 2$ for the random class (since in this case, the noise comes both from the variable Z and from the noise R). For example, this means a signal 0 for the fixed class $\mathsf{HW}(z_f) = 4$, a signal 0.25 for the fixed class $\mathsf{HW}(z_f) = 3$, a signal 1 for the fixed class $\mathsf{HW}(z_f) = 2$, a signal 2.25 for the fixed class $\mathsf{HW}(z_f) = 1$, and a signal 4

for the fixed class $\mathsf{HW}(z_f) = 0$. Ignoring the small noise differences between the tests, it means that the sampling complexity for detecting leakages with the t-test and a fixed class $\mathsf{HW}(z_f) = 1$ should be close to (and slightly smaller than) the sampling complexity for detecting leakages with the ρ-test. And this is exactly what we observe on the figure, for the ρ-test *with a perfect model*. The same reasoning can be used to explain the sampling complexities of the t-test for different fixed values. For example, the case $\mathsf{HW}(z_f) = 3$ requires four times more traces than the case $\mathsf{HW}(z_f) = 2$ on the figure.

A consequence of this observation is that, as for standard DPA attacks, the choice of statistic (here the t-test or ρ-test) has limited impact on the sampling complexity of the detection. For example, one could totally design a ρ-test based on a partition in two (fixed and random) classes, that would then lead to very similar results as the t-test (up to statistical artifacts, as discussed in [17]).

(5) *Estimating a model can only make it worse.* Besides the potentially lower signal, another drawback of the 256-class ρ-test from the sampling complexity point-of-view is that it requires the estimation of a model made of 256 mean values. This further increases its overheads compared to the t-test, as illustrated in Fig. 1 (see the $\hat{r}_z$ curve with $k = 10$-fold cross-validation). In this respect, we first note that considering larger k's only leads to very marginal improvements of the detection (at the cost of significant computational overheads). Besides, we insist that this estimation is unavoidable. For example, ignoring the cross-validation and testing a model with the same set as its profiling set would lead to overfitting and poor detection performances. In other words, it is the size of the partition used in the ρ-test that fixes its SNR (as previously discussed) and estimation cost, and both determine the final sampling complexity of the test.

Note that the above conclusions are independent of the leakage function considered (we repeated experiments with identity rather than Hamming weight leakages, and reached the same conclusions). Therefore, these simulated results confirm our introduction claim that for leakage detection only, a non specific t-test is the method of choice, and that its gains over a ρ-test can be easily predicted from a leakage function/partition and its resulting SNR metric.

4.2 Measured Experiments

We now extend the previous simulated analysis to the practically-relevant case of actual AES measurements, obtained from the setup described in Sect. 2.1. We will divide our investigations in two parts. First, a global analysis will consider the leakage traces of the full AES executions, in order to discuss the sampling complexity and intuitions regarding the POIs for our two detection tests. Next, a local analysis will be used in order to discuss possible false negatives in the t-test, and intuitions regarding the informativeness of the detected samples.

Global Analysis. The results of a fixed vs. random t-test and a ρ-test for leakage traces corresponding to an entire AES Furious execution are provided in Fig. 2, from which two main observations can be extracted.

(1) *The t-test has lower sampling complexity on average.* This is essentially the concrete counterpart of observation (3) in the previous section. That is, we already know that for some fixed values of the plaintext, the t-test should have a lower sampling complexity. Figure 2 confirms that when looking at complete AES traces, those "easy-to-detect" fixed values are indeed observed (which is natural since the AES Furious implementation accounts for a bit more than 3000 clock cycles, and the intermediate values within such a block cipher execution should be uniformly distributed after a couple of rounds). Concretely, this means that the sampling complexity for detecting leakages with a similar confidence increases from ≈ 200 traces for the t-test to ≈ 2000 traces for the ρ-test, i.e. a factor ≈ 10 which is consistent with the previous simulations. Note that even in the context of a hardware implementation with a reduced cycle count (e.g. 11 cycles per AES execution), finding fixed values that are easy-to-detect for the t-test is feasible by trying a couple of fixed plaintexts and keys.

(2) *The ρ-test (resp. t-test) does (resp. not) provide intuitions regarding exploitable leakage samples.* This is easily seen from the figure as well. Whereas the t-test detects information leakage everywhere in the trace, the ρ-test is much more localized, and points towards the samples that depend on the single plaintext byte of which the leakage is considered as signal (here corresponding to the first round and first S-box). Since the key is fixed in leakage detection, it implies that peaks are observed whenever this (useless)

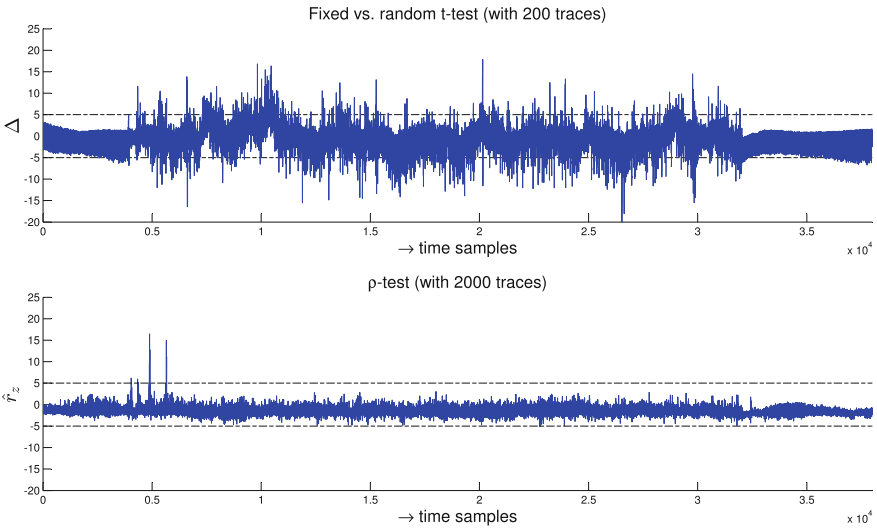

Fig. 2. Leakage detection on real traces, entire AES execution.

plaintext byte and the (useful) intermediate values that bijectively depend on it are manipulated, e.g. the key addition and S-box outputs in Fig. 2. In other words, the ρ-test is mostly relevant for the detection of POIs that are exploitable in a standard DPA attack (i.e. excluding the false positives corresponding to plaintext manipulations).

Local Analysis. The results of a fixed vs. random t-test and a ρ-test for leakage traces corresponding to the beginning of the first AES round execution are provided in Fig. 3, from which two main observations can be extracted.[4]

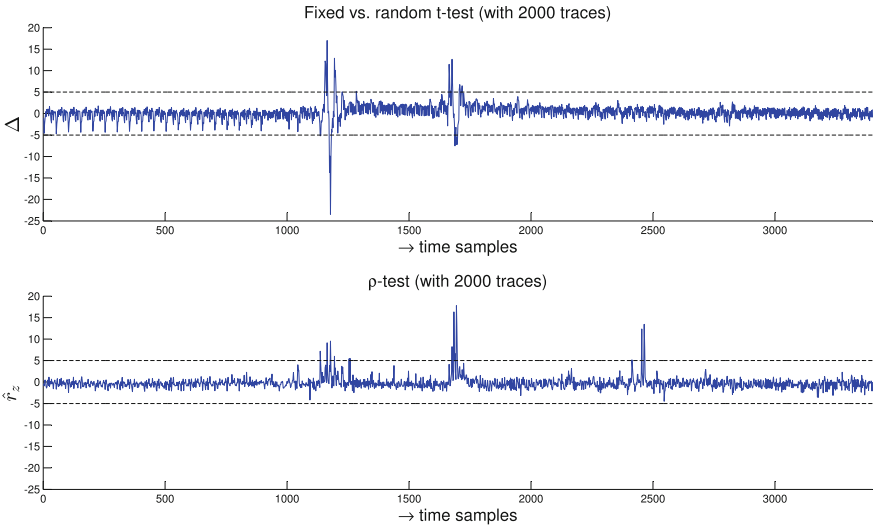

Fig. 3. Leakage detection on real traces, first-round AES key addition and S-box.

(1) *Hard-to-detect leakage samples for the t-test can be observed.* More precisely, the lower part of Fig. 3 exhibits three peaks which exactly correspond to the manipulation of a plaintext byte (first peak), the key addition (second peak) and the S-box execution (third peak), just as the three samples of our simulated setting in Sect. 4.1. Knowing that our Atmel implementation of the AES has leakages that can be efficiently exploited with a Hamming weight model (as in our simulations) [33], we selected the fixed plaintext byte of the t-test such that $\mathsf{HW}(z_f) = 4$. As illustrated in the upper part of the figure, the leakages of this fixed intermediate value are indeed difficult to tell apart from the one of its random counterpart. More precisely, the ρ-test clearly exhibits a peak for this intermediate value after 2000 traces, which does not exist in the t-test experiment using a similar sampling complexity. Whereas we cannot exclude that such a peak would appear for a larger number of

[4] Exceptionally for this experiment, we considered a single varying byte for the t-test, in order to better exhibit intuitions regarding the detected samples for a single S-box.

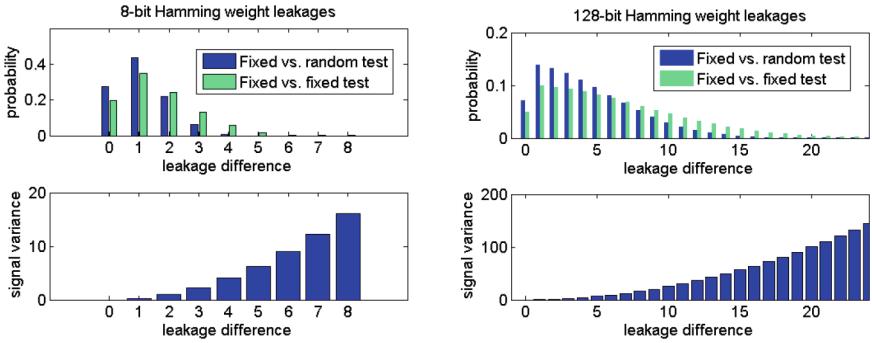

Fig. 4. Fixed vs. random and fixed vs. fixed leakage detection signal.

traces (since the chip does not exactly follow the Hamming weight leakage model), this confirms that not all leakage samples are easier to detect with the t-test than with the ρ-test.

(2) *The ρ-test does provide intuitions regarding the informativeness of the leakage samples.* Eventually, a straightforward advantage of the ρ-test is that the value of its correlation coefficient estimates brings some intuition regarding the complexity of a side-channel attack exploiting this sample, which is only provided up to a limited extent by the t-test. Indeed, a side-channel attack exploiting an s-bit intermediate value is most efficient if it relies on an s-bit model, as considered by the ρ-test (otherwise $s-1$ bits out of s will produce "algorithmic noise"). In this context, we can take advantage of the connection between Pearson's correlation coefficient and the information theoretic metrics in [34] (see [17]), themselves related to the worst-case complexity of standard DPA attacks [5].

5 Improved Leakage Detection Test

One central conclusion of the previous section is that the sampling complexity of leakage detection tests highly depends on the SNR of the leakage partition on which they are based. Interestingly, this observation directly suggests a natural improvement of CRI's non specific (fixed vs. random) t-test. Namely, rather than performing the test based on a fixed and a random class, a more efficient solution is to perform a similar test based on two fixed classes (i.e. two fixed plaintexts). On the one hand, this directly reduces the detection noise from $2\sigma_n^2 + \sigma_{\text{alg}}^2$ to $2\sigma_n^2$, since it cancels the algorithmic noise due to the variations of the random class. Taking the example of Hamming weight leakages, this algorithmic noise corresponds to $\sigma_{\text{alg}}^2 = 2$ for 8-bit values, but it increases for larger parallel implementations (e.g. it is worth $\sigma_{\text{alg}}^2 = 32$ for 128-bit implementations). On the other hand, and when applied to large traces, such a partitioning also increases the signal with high probability, for the same argument as used to avoid false

positives in CRI's t-test (i.e. by applying the detection to large enough traces, large differences between the two fixed classes will inevitably occur). Taking the example of Hamming weight leakages again, we can easily compute the probability (over random inputs) that a certain leakage difference is obtained for both types of partitions (i.e. fixed vs. random and fixed vs. fixed), and the resulting signal variance, as illustrated in Fig. 4. We conclude from this figure that (*i*) the fixed vs. fixed partitioning allows reaching larger differences (so larger signals) and (*ii*) the fixed vs. fixed partitioning allows doubling the average signal (i.e. the dot product of the probabilities and variances in the figure). So both from the noise variance and the (best-case and average case) signal points-of-views, it should improve the sampling complexity of the detection test.[5] In other words, a leakage detection based on a fixed vs. fixed leakage partition should theoretically have better sampling complexity than with a fixed vs. random one.

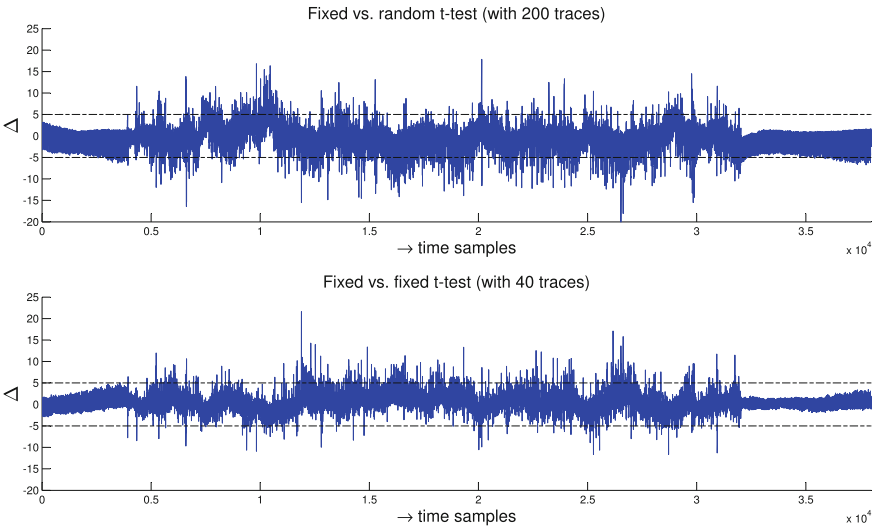

Fig. 5. Improved leakage detection on real traces (Atmel implementation).

Quite naturally, the exact gains of this new detection test depend on the actual leakages. So as in the previous section, we confirmed our expectations with two case studies. First, we compared the fixed vs. random and fixed vs. fixed t-tests based on our software AES implementation. The results of this experiment are in Fig. 5 where we observe that data-dependent leakages are detected with similar confidence with approximately 5 times less traces thanks to our new partitioning. Next, we investigated the context of the hardware implementation of the AES described in Sect. 2.1. As illustrated in Fig. 6, similar gains are obtained.

[5] A similar conclusion can be obtained for other leakage functions, though the binomial distribution of the Hamming weight leakages naturally make computations easier.

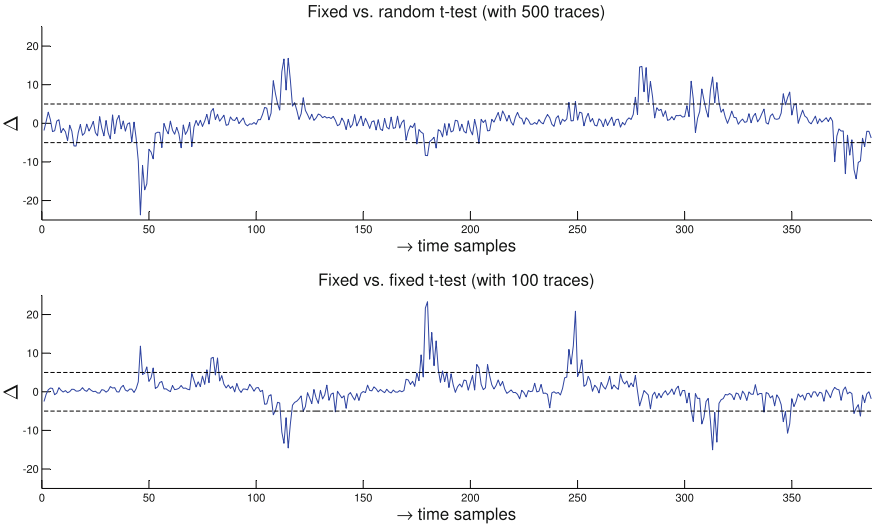

Fig. 6. Improved leakage detection on real traces (ASIC implementation).

Note however that despite we gain an approximate factor 5 in both cases, the reasons of this gain are different. Indeed, the software implementation case is dominated by an increase of signal (due to its large cycle count) and has limited algorithmic noise. By contrast, the hardware implementation has larger algorithmic noise (corresponding to 128-bit random values) but less improvements of the signal (because its traces are only 11-cycle long). Even larger gains could be obtained by combining both the signal and noise effects (e.g. by considering multiple keys for the hardware implementation). Based on these theoretical arguments and experimental confirmation, we expect our fixed vs. fixed partitioning to lead to faster leakage detections in most practical scenarios.

Remark. The fixed vs. fixed test can only be successful if the vectors used in the test exhibit significant differences for the target algorithm's intermediate computations (which is the counterpart of having fixed leakages different from average leakages in CRI's fixed vs. random test). This is easily obtained with block cipher implementations for which these intermediates are pseudorandom.

6 From Leakage Detection to POI Detection

The previous sections lead to the natural conclusion that non specific tests are a method of choice for leakage detection. In particular, their application to full leakage traces (or multiple keys) allows overcoming the issue of false positives mentioned in Sect. 4.1. By contrast, the correlation-based test is better suited to the detection of POIs because it provides useful intuitions regarding the exploitable samples in side-channel traces and their informativeness. As a result, it is a good candidate for the more specific task of detecting POIs for mounting an attack. In this section, we conclude the paper by putting forward that a

ρ-test is in fact perfectly suited for integration in (an improvement of) a recent POI detection tool proposed by Durvaux et al. at COSADE 2015 [7]. For this purpose, we first briefly recall how this tool works, then describe our improvements based on our proposed ρ-test, and provide experimental confirmation of our claims.

Note that in general, the problem of detecting POIs is relatively easy in the context of unprotected implementations. Indeed, exhaustive analysis is usually feasible in this case, and it is even possible to look for optimal transforms that project the samples towards small (hence easier-to-evaluate) subspaces such that most of their informativeness is preserved, e.g. using Principal Component Analysis (PCA) [1], which maximizes the side-channel signal, or Linear Discriminant Analysis (LDA) [32], which maximizes the side-channel SNR. In fact, in this context, any criteria can be easily optimized using local search [7,22], and most criteria are essentially equivalent anyway (i.e. correlation, SNR, mutual information and success rate [5,17]). Therefore, our focus will be on the more challenging case of masked implementation, which requires a specialized local search.

6.1 The COSADE 2015 POI Detection Tool

The COSADE 2015 POI detection aims at finding a projection vector $\boldsymbol{\alpha}$ that converts the N_s samples $l_y(\tau)$ of a leakage trace into a single projected sample λ_y:

$$\lambda_y = \sum_{\tau=1}^{N_s} \alpha(\tau) \cdot l_y(\tau).$$

In the case of unprotected implementations, and as previously mentioned, it is possible to find projections $\boldsymbol{\alpha}$ that optimize the informativeness of the projected sample (where the $\alpha(\tau)$ coefficients are real numbers, typically). By contrast, in the context of masking, the task is arguably more difficult since (*i*) single samples may not contain information (e.g. in the context of software implementations where the different shares are manipulated at different time samples), and (*ii*) the information about the target intermediate variables lies in higher-order moments of the leakage distribution. Therefore, Durvaux et al. focused on the simplified problem of finding a projection such that $\alpha(\tau) = 1$ if the time sample $l_y(\tau)$ contains some information about a share, and $\alpha(\tau) = 0$ otherwise.

In this context, a naive solution would be to consider each possible combination of time samples, but this scales badly (i.e. exponentially) with the number of shares to detect, and is rapidly prohibitive in practice, even for two shares (since masked implementations generally imply traces with many samples). In order to avoid this drawback, the algorithm proposed in [7] works by considering d non-overlapping windows of length W_{len} that set the covered weights to 1 (and leaves to others stuck at 0). Algorithm 1 provides a succinct description of this method. Besides the previously mentioned window length, it mainly requires defining an objective function f_{obj} and a detection threshold T_{det}, and works in two steps. First, the find_solution phase places the windows randomly at different

Algorithm 1. Local search algorithm for finding POIs in masked traces.

```
Local_Search(d, W_len, T_det, @f_obj)
    α = find_solution(d, W_len, T_det, @f_obj);
    if(α ≠ null)
        return improve_solution(α, @f_obj);
    end
end
```

locations of the trace, until the returned value of the objective function crosses the threshold. Then, the improve_solution modifies the windows' size in order to best fit the informative time samples. As a result, we obtain the position and the size of each window that maximizes f_{obj}. By changing the number of windows and objective function, this approach can easily be extended to masking schemes of any order and number of shares. Intuitively, the W_{len} parameter leads to a natural tradeoff between the time complexity and sampling complexity of the algorithm. Namely, small window lengths are more time intensive[6], and large ones more rapidly cover POIs, but imply an estimation of the objective function for samples projected according to larger windows, which are potentially more noisy. Eventually, the objective function proposed in the COSADE paper is the Moments-Correlating Profiled DPA (MCP-DPA) introduced in [21], which can be viewed as a classical higher-order DPA based on the CPA distinguisher given in Sect. 2.2, where one correlates the leakages samples raised to a power d with a model corresponding to the dth-order statistical moment of the leakage distribution. We refer to the previous papers for the details on these tools.

6.2 Our Contribution

We first recall that the COSADE 2015 POI detection tool is *black box* in the sense that it does not require any knowledge of the target implementation. By contrast, it does require *key profiling*, since the MCP-DPA distinguisher is a profiled one. In this respect, our first contribution is the simple but useful observation that one can easiliy apply such a black box POI detection without key profiling, by simply profiling the MCP-DPA objective function based on plaintext knowledge, just as the ρ-test in this paper. Indeed, when detecting POIs, it is sufficient to know the leakage model up to a permutation corresponding to key knowledge (a quite similar idea has been exploited in [28] for similar purposes). As previously discussed, this solution will suffer from the (minor) risk of detecting plaintext samples, but as will be detailed next, this can be easily mitigated in practice.

Based on these premises, our second contribution starts from the equally simple observation that the ρ-test of this paper can be used identically with the MCP-DPA distinguisher. So it is theoretically eligible for detecting leakages and POIs of any order. Therefore, by replacing the MCP-DPA objective function in [7] by the ρ-test in this paper (based on CPA or MCP-DPA), we obtain

[6] For example, $W_{len} = 1$ is equivalent to testing all combinations of time samples.

a very simple and rigorous way to set the detection threshold in Algorithm 1. That is, one just has to use the same "five sigma rule" as used in the leakage detections of Figs. 2 and 3. Note by changing the objective function and selection of a detection threshold in this way, we benefit from the additional advantage of (more efficiently) estimating the objective function with cross-validation, which is another improvement over the method described at COSADE 2015.

Third and maybe most importantly, we notice that the COSADE 2015 objective function is based on the estimation of central (second-order) statistical moments. That is, given the (average) leakage of the two windows, it first sums them and then computes a variance. But looking at the discussion in [35], Sect. 4, it is clear that whenever the leakages are relatively noisy – which will always happen in our context of which the goal is to exploit the largest possible windows in order to reduce the time needed to detect POIs – considering mixed statistical moments is a better choice. In other words, by exploiting the multivariate MCP-DPA mentioned at the end of [21], we should be able to detect POIs with larger windows. In our following experiments based on a first-order (2-shares) masking scheme, this just means using the normalized product between the (average) leakage of the two windows as objective function, which has been shown optimal in the context of Hamming weight leakages in [26].

6.3 Experimental Validation

In order to confirm the previous claims, we tested Algorithm 1 using exactly the previously described modifications, based on a target implementation and measurement setup very similar to the one in [7]. That is, we first analyzed the leakages of the masked implementation described in Sect. 2.1 which leads to large traces with $N_s = 30,000$ samples (for which an exhaustive analysis of all the pairs of samples is out of reach). As in the COSADE 2015 paper, we verified that our implementation does not lead to any first-order leakages (this time with the ρ-based test from Sect. 3). We further set the window length to 25 samples, which corresponds to a bit more than two clock cycles at our clock frequency and sampling rate. With these parameters, the local search was able to return a solution within the same number of objective function calls as [7], namely $\approx 12\,000$ on average. An example of leakage trace together with windows obtained thanks to Algorithm 1 is given in Fig. 7. As clear from the zoomed plots at the bottom of the figure (where we represent the sum of the projection vectors obtained after 100 experiments), the selection of POIs corresponds to leakage samples that combine the precomputation and masked S-box computation. Interestingly, we could expect some false positives due to the detection of plaintext bytes that is possible in our non-profiled scenario. However, the improve_solution of Algorithm 1 (where the window size is adapted to be most informative) combined with the fact that the most informative leakage samples in our traces correspond to memory accesses (i.e. the S-box computations) prevented these to happen. Note that even if the leakage of the plaintext manipulations was more informative, we could easily "mark" the cycles that correspond to plaintext knowledge only, and exclude them from our optimization. Since the number of POIs corresponding

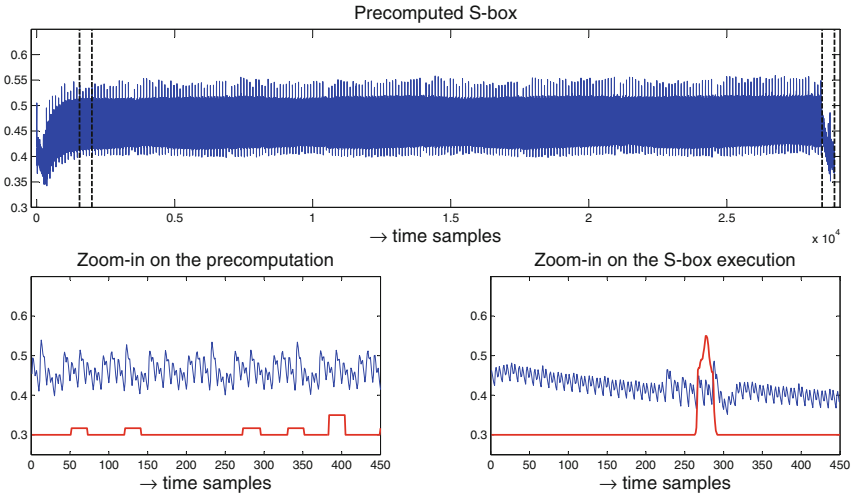

Fig. 7. Non-profiled detection of POIs based on our ρ-test.

to a single plaintext byte is usually limited, this would lead to the detection of a valid pair of POIs after a couple of iterations of Algorithm 1. Besides, we note that Simple Power Analysis, or DPA against the plaintext (before it is XORed with the masks) are other simple ways to gain the minimum intuition about the time localization of the POIs, in order to avoid false positives when running a non-profiled local search.

Next, we analyzed the impact of our modified objective function on the largest window lengths for which we could detect POIs. As illustrated in Fig. 8, we can clearly observe a (significant) gain of an approximate factor > 3 when using a normalized product combining as objective function rather than the previously used square of sum (for which the figure also suggests that the window of length 25 was close to optimal). It means that for exactly the same amount of traces in an evaluation set, we are able to detect POIs with > 3 times larger windows with our improved objective function and detection threshold. Concretely, this corresponds to a reduction of the time complexity by a factor > 3 compared to the COSADE 2015 results (and by a factor ≈ 90 compared to a naive combinatorial search). Interestingly, we also see that increasing the window length is not always detrimental, which corresponds to the fact that larger windows do not only contain more noise, but also more informative samples.

To conclude, we believe the connections made in this section are important to raise awareness that up to the selection of POIs in the leakage traces, side-channel security evaluations can essentially be performed in a black box way, and without any key profiling. In this respect, leakage detection and the detection of POIs are indeed related tasks, with the significant difference that the latter has to take the exploitability of the detected samples into account. And this is exactly the difference between simple t-tests and more measurement-intensive ρ-tests based on larger leakage partitions. Note that the non-profiled detection in

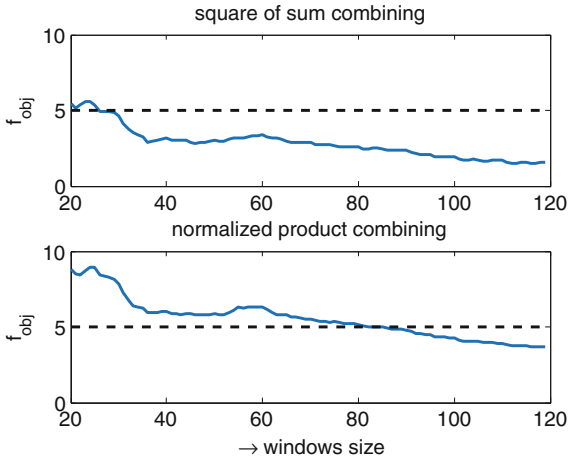

Fig. 8. Estimation of objective functions (with cross-validation) based on central and mixed statistical moments, in front of a detection threshold of five sigma.

this section only applies to the first/last block cipher rounds (i.e. before diffusion is complete), which captures many relevant practical scenarios but could be an issue, e.g. in contexts where these extreme rounds are better protected than the central ones. Besides, and more generally, we recall that as soon as the POIs are detected and the evaluator has to build a model for these samples, key profiling becomes strictly necessary to evaluate a worst-case security level [39].

7 Summary and Open Problems

The discussion in this paper highlights that there are significant differences between current approaches to side-channel security evaluation. On the one hand, CRI's Test Vector Assessment Methodology (TVLA) aims at minimizing the evaluator's efforts. Very concretely, non specific t-tests as proposed in [4,10] are indeed good to detect univariate and first-order leakages. As we observed in Sect. 5, slightly tweaking the selection of the classes (from fixed vs. random to fixed vs. fixed) allows significantly improving the detection speed in this case. We can expect these gains to be even more significant in the context of masked implementations (for which the impact of noise is exponentially amplified). The fixed vs. fixed test also has good potential for evaluating the implementations of asymmetric cryptographic primitives. So despite minor theoretical caveats (i.e. the possibility of false positives and negatives), the application of such 2-class t-tests turns out to be extremely efficient. On the other side of the spectrum, complete (ideally worst-case) security evaluations such as discussed in [34] rather aim at a precise rating of the security level, possibly considering the adversary's computing power [36], which is an arguably more expensive task. In this case, the selection of POIs is a usually a necessary first step. As also discussed in

this paper, and when restriced to univariate and first-order leakages, the main reason for the additional cost of this approach (including the selection of POIs) is the larger number of classes for which the leakage distribution has to be well estimated. In this context as well, our investigations focused on non-profiled POI detection (which can be performed efficiently for the first/last cipher rounds). But similar conclusions hold in the profiled evaluation setting, which allows finding POIs in all the cipher rounds, and is necessary for worst-case analysis.

These different methodologies naturally raise the question of which one to use in which context, and whether they can be connected to some extent, leading to the following open problems. First, how to generalize (simple) detection tests to capture more types of leakages? Moving from univariate first-order leakages to univariate higher-order leakages is already reachable with existing tools. One option, already described in Sect. 6, is to work "by moments". This implied to implement a Moments-Correlating DPA in our multi-class context, but could naturally be specialized to simpler t-tests, F-tests,..., if only 2 classes were considered: see [30] for a recent discussion that is complementary to our results. Another option is to exploit more general statistical tests, e.g. Mutual Information Analysis [8], as already applied in the context of leakage detection by Mather et al. [18]. Moving to multivariate leakage detection appears much more difficult. At least, testing all pairs/triples/... of samples in a trace rapidly turns out to be unfeasible as the size of the traces increase, which usually leads current evaluations to be based on heuristics (e.g. the ones discussed in Sect. 6). Note that the gap between univariate and multivariate attacks is probably among the most important remaining challenge in side-channel security evaluations, where significant risks of false negatives remain. A typical example is the case of static leakages that may only be revealed in the context of (highly) multivariate analyses [20,24]. More generally, limiting an evaluation to univariate leakages typically ignores the significant gains that can be obtained with dimensionality reductions techniques (aka projections), and multi-target attacks [12,19,37].

Second, can we extrapolate or bound the worst-case security level of an implementation based on simple statistical tests? For example, the recent work in [5] shows that one can (in certain well-defined conditions) bound the security level of an implementation, measured with a success rate and in function of the number of measurements and computing power of the adversary, based on information theoretic metrics (such as the mutual information in general, and the SNR if we only consider univariate attacks). But as discussed in this paper, evaluating an SNR is still significantly more expensive than detecting leakages with non specific tests. So of course, it would be interesting to investigate whether it is possible to bound the security level based on simpler leakage detection tests. In case of negative answer, it anyway remains that such leakage detection tests can always be used as a preliminary to more expensive approaches (detecting POIs, security evaluations), e.g. to reduce the dimensionality of the traces.

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

References

1. Archambeau, C., Peeters, E., Standaert, F.-X., Quisquater, J.-J.: Template attacks in principal subspaces. In: Goubin and Matsui [11], pp. 1–14
2. Balasch, J., Gierlichs, B., Grosso, V., Reparaz, O., Standaert, F.-X.: On the cost of lazy engineering for masked software implementations. In: Joye, M., Moradi, A. (eds.) CARDIS 2014. LNCS, vol. 8968, pp. 64–81. Springer, Heidelberg (2015)
3. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)
4. Cooper, J., De Mulder, E., Goodwill, G., Jaffe, J., Kenworthy, G., Rohatgi, P.: Test vector leakage assessment (tvla) methodology in practice (extended abstract). ICMC 2013. http://icmc-2013.org/wp/wp-content/uploads/2013/09/goodwillkenworthtestvector.pdf
5. Duc, A., Faust, S., Standaert, F.-X.: Making masking security proofs concrete. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 401–429. Springer, Heidelberg (2015)
6. Durvaux, F., Standaert, F.-X., Veyrat-Charvillon, N.: How to certify the leakage of a chip? In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 459–476. Springer, Heidelberg (2014)
7. Durvaux, F., Standaert, F.-X., Veyrat-Charvillon, N., Mairy, J.-B., Deville, Y.: Efficient selection of time samples for higher-order DPA with projection pursuits. In: Mangard, S., Poschmann, A.Y. (eds.) COSADE 2015. LNCS, vol. 9064, pp. 34–50. Springer, Heidelberg (2015)
8. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual information analysis. In: Oswald and Rohatgi [23], pp. 426–442
9. Gierlichs, B., Lemke-Rust, K., Paar, C.: Templates vs. stochastic methods. In: Goubin and Matsui [11], pp. 15–29
10. Goodwill, G., Jun, B., Jaffe, J., Rohatgi, P.: A testing methodology for side channel resistance validation. NIST non-invasive attack testing workshopp (2011). http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf
11. Goubin, L., Matsui, M. (eds.): CHES 2006. LNCS, vol. 4249. Springer, Heidelberg (2006)
12. Grosso, V., Standaert, F.-X.: ASCA, SASCA and DPA with enumeration: which one beats the other and when? In: Iwata, T., et al. (eds.) ASIACRYPT 2015. LNCS, vol. 9453, pp. 291–312. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48800-3_12
13. Kerckhof, S., Durvaux, F., Hocquet, C., Bol, D., Standaert, F.-X.: Towards green cryptography: A comparison of lightweight ciphers from the energy viewpoint. In: Prouff and Schaumont [27], pp. 390–407
14. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
15. Mangard, S.: Hardware countermeasures against DPA – a statistical analysis of their effectiveness. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 222–235. Springer, Heidelberg (2004)

16. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks - Revealing the Secrets of Smart Cards. Springer, New York (2007)
17. Mangard, S., Oswald, E., Standaert, F.-X.: One for all - all for one: unifying standard differential power analysis attacks. IET Inf. Secur. **5**(2), 100–110 (2011)
18. Mather, L., Oswald, E., Bandenburg, J., Wójcik, M.: Does my device leak information? an *a priori* statistical power analysis of leakage detection tests. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 486–505. Springer, Heidelberg (2013)
19. Mather, L., Oswald, E., Whitnall, C.: Multi-target DPA attacks: Pushing DPA beyond the limits of a desktop computer. In: Sarkar and Iwata [29], pp. 243–261
20. Moradi, A.: Side-channel leakage through static power. In: Batina, L., Robshaw, M. (eds.) CHES 2014. LNCS, vol. 8731, pp. 562–579. Springer, Heidelberg (2014)
21. Moradi, A., Standaert, F.-X.: Moments-correlating DPA. IACR Cryptology ePrint Archive, 2014:409 (2014)
22. Oswald, D., Paar, C.: Improving side-channel analysis with optimal linear transforms. In: Mangard, S. (ed.) CARDIS 2012. LNCS, vol. 7771, pp. 219–233. Springer, Heidelberg (2013)
23. Oswald, E., Rohatgi, P. (eds.): CHES 2008. LNCS, vol. 5154. Springer, Heidelberg (2008)
24. Pozo, S.M., Del Standaert, F.-X., Kamel, D., Moradi, A.: Side-channel attacks from static power: when should we care? In: Nebel, W., Atienza, D. (eds.) DATE 2015, Grenoble, France, March 9–13, 2015, pp. 145–150. ACM (2015)
25. Prouff, E., Rivain, M.: A generic method for secure sbox implementation. In: Kim, S., Yung, M., Lee, H.-W. (eds.) WISA 2007. LNCS, vol. 4867, pp. 227–244. Springer, Heidelberg (2008)
26. Prouff, E., Rivain, M., Bevan, R.: Statistical analysis of second order differential power analysis. IEEE Trans. Comput. **58**(6), 799–811 (2009)
27. Prouff, E., Schaumont, P. (eds.): CHES 2012. LNCS, vol. 7428. Springer, Heidelberg (2012)
28. Reparaz, O., Gierlichs, B., Verbauwhede, I.: Selecting time samples for multivariate DPA attacks. In: Prouff and Schaumont [27], pp. 155–174
29. Sarkar, P., Iwata, T. (eds.): ASIACRYPT 2014. LNCS, vol. 8873. Springer, Heidelberg (2014)
30. Schneider, T., Moradi, A.: Leakage assessment methodology. In: Güneysu, T., Handschuh, H. (eds.) CHES 2015. LNCS, vol. 9293, pp. 495–513. Springer, Heidelberg (2015)
31. Schramm, K., Paar, C.: Higher order masking of the AES. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 208–225. Springer, Heidelberg (2006)
32. Standaert, F.-X., Archambeau, C.: Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In: Oswald and Rohatgi [23], pp. 411–425
33. Standaert, F.-X., Gierlichs, B., Verbauwhede, I.: Partition *vs.* comparison side-channel distinguishers: an empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected CMOS devices. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 253–267. Springer, Heidelberg (2009)
34. Standaert, F.-X., Malkin, T.G., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 443–461. Springer, Heidelberg (2009)

35. Standaert, F.-X., Veyrat-Charvillon, N., Oswald, E., Gierlichs, B., Medwed, M., Kasper, M., Mangard, S.: The world is not enough: another look on second-order DPA. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 112–129. Springer, Heidelberg (2010)
36. Veyrat-Charvillon, N., Gérard, B., Standaert, F.-X.: Security evaluations beyond computing power. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 126–141. Springer, Heidelberg (2013)
37. Veyrat-Charvillon, N., Gérard, B., Standaert, F.-X.: Soft analytical side-channel attacks. In: Sarkar and Iwata [29], pp. 282–296
38. Welch, B.L.: The generalization of Student's problem when several different population variances are involved. Biometrika, pp. 28–35 (1947)
39. Whitnall, C., Oswald, E., Standaert, F.-X.: The myth of generic DPA...and the magic of learning. In: Benaloh, J. (ed.) CT-RSA 2014. LNCS, vol. 8366, pp. 183–205. Springer, Heidelberg (2014)