Second-order *Scatter* Attack

Hugues Thiebeauld[1], Aurelien Vasselle[1] and Antoine Wurcker[1]

eshard, France, surname.name@eshard.com

Abstract. Second-order analyses have shown a great interest to defeat first level of masking protections. Their practical realization remains tedious in a lot of cases. This is partly due to the difficulties of achieving a fine alignment of two areas that are combined together afterward. Classical protections makes therefore use of random jitter or shuffling to make the alignment difficult or even impossible.
This paper extends *Scatter* attack to high-order analyses. Processing the joint-distribution of two selection of points, it becomes possible to retrieve the secret key even when traces are not fully aligned.
The results presented in this paper are validated through practical experimentation and compared with existing window-based techniques, such as the FFT. *Scatter* shows the best results when misalignment is significant.
This illustrates that *Scatter* offers an alternative to existing high-order attacks and can target all kinds of cryptography implementations, regardless they are executed in hardware or software. With the ability to exploit several leakage points, it may be valuable also when applying a second-order attack on aligned traces.

Keywords: Side-channel · Misalignment · *Scatter* · Second-order · Mutual Information · Sobel · χ^2 · Image Processing · Boolean Masking

1 Introduction

Introduced in [Koc96], side-channel attacks were found to be efficient to retrieve secret from signals emitted by any hardware device. All cryptography implementations in embedded systems are potentially threatened by side-channel attacks. One way to avoid a direct exploitation of leakage information is to avoid the device handling a value correlated to the sensitive one. To achieve this, Boolean masking can be applied. Instead of processing the value val, the device manipulates $val \oplus mask$, with $mask$ being a random value [CJRR99].

High-order side-channel attacks came after the Differential Power Analysis [KJJ99] publication. In [Mes00], the author showed that Boolean masking could be defeated by computing the difference of means between two points of a trace, one representing the $mask$, and the other the masked value $val \oplus mask$. Since then, it has been common to develop a high-order variant each time a new side-channel technique is introduced. A deep study explored the high-order Correlation Power Analysis (CPA) [PRB09] and later on, shortly after the Mutual Information Analysis (MIA) [GBTP08] was introduced, complementary studies expressed how MIA could be extended to high-order [GBPV10, PR10, BGP+11].

As statistical attacks, these techniques have the strong prerequisite that a fine alignment must be achieved prior to the analysis, otherwise any observation misalignment would replace leakage information with noise. Since the alignment is not a simple task, it may represent a show-stopper for some attacks. Some researchers explored the opportunity to integrate points in time, as detailed in Section 2.2. For first-order leakages, [CCD00] suggested to average points before processing them with the statistical function. In the

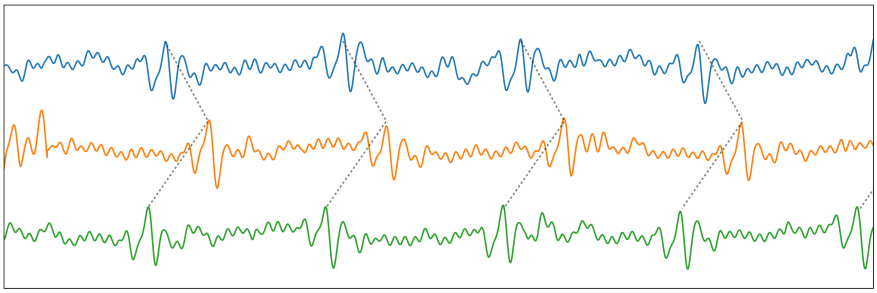


Figure 1: Example of static misalignment

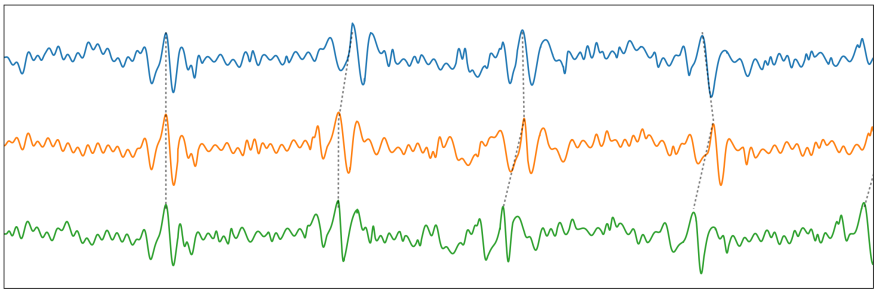


Figure 2: Example of elastic misalignment

context of second-order attacks, in [WW04], authors made use of a cross-correlation that was later on improved in [BBB+16] with complementary variants of frequency domain transformations.

1.1 Misalignment Issue

Misalignment may have different causes. It may come from an inaccurate trigger, resulting in a simple trace shifting, as depicted in Figure 1. When a recognizable feature can be chosen as reference, it is possible to align the traces together. Unless there is no feature, the effort is limited and alignment can be achieved.

More difficulties come when random jitter is observed all along the sensitive process. As in Figure 2, the trace execution becomes elastic, meaning that the operation processing time is variable. This may come from specific hardware or software countermeasures, such as Random Delay Insertion (RDI), variable clock or dummy operations. Another reason is the speculative execution of complex devices (e.g. System-on-Chips), where the processing is optimized over the different resources available and may differ from one execution to the other.

In that conditions, alignment requires the identification, without error, of all valuable features in each trace. Then it becomes possible to align them together one after the other. This may be challenging for long operations, when clear features cannot be found and require different levels of signal processing, or especially when features does not contain the same number of oscillations.

Figure 3 represents another kind of misalignment, related to the shuffling [RPD09] or dummy operations countermeasures. The first aims at executing similar operations in a random orders. And the second adds fake but similar operations and randomly hide the

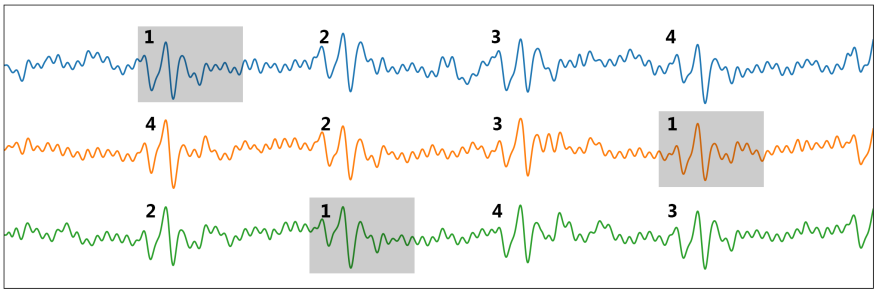


Figure 3: Example of shuffling misalignment

genuine within all executions. In both cases, the alignment becomes impossible as long as the targeted operation cannot be identified from the others.

From a theoretical point of view, the correlation coefficient is multiplied by $1/\sqrt{t}$ when the leakage is uniformly located over t different samples. The attack requires therefore roughly t times more traces to succeed [Sou11]. In [TGWC18], this ratio is shown to be less than t for first-order *Scatter* attacks integrating t samples. Exploiting this property may be useful for second-order attacks.

1.2 Contribution

This paper extends *Scatter* [TGWC18] to the scope of second-order side-channel analyses. It is particularly relevant when alignment is difficult. Indeed, aligning traces remains a challenge in a large number of practical second-order attacks. To overcome this issue, two selections of points are made and their joint distribution computed. Leakage points must be part of the selections, but their location does not matter. This means that the technique is insensitive to any kind of misalignment within the selection. Following this, a partitioning is done as described in [BGP+11]. Furthermore, the technique is able to leverage multiple points of leakage points available in the same data trace as well.

A theoretical formalization shows that *Scatter* integration remains effective even though the selected points hold few leakage-related components. A practical case is developed, highlighting that the technique is effective and that it achieves better results compared to all other techniques using time integration, such as the cross-correlation or Fourier transformations. The performance is particularly noticeable when centering is necessary but impossible to estimate accurately due to misalignment.

However, integrating non-leakage points does have a negative impact which can be partially compensated by applying specific preprocessing operation prior to any distinguisher. This paper introduces new techniques showing promising results. The value is not limited to *Scatter* and should be considered to enhance other partition techniques, such as second-order MIA.

Finally, we demonstrate that classical convolutions from the image processing can be used as distinguisher. Combined with a parametric projection, Sobel kernels showed very good results on the practical case developed in this paper, in which the second-order CPA is outperformed, even on aligned traces.

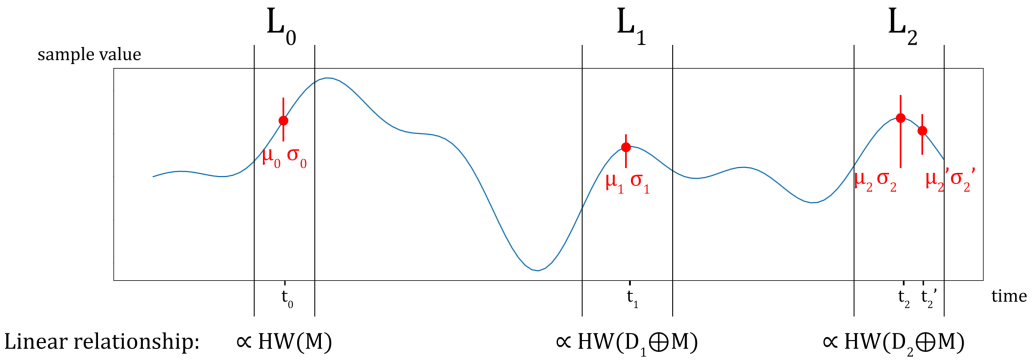


Figure 4: Example of a common second-order leakage scenario

2 Related Work

2.1 Second-Order Attacks

Second-order attacks typically target cryptography devices leaking sensitive data hidden behind a random *mask*. As depicted in Figure 4, two points represent leakages of two different intermediate data D_1 and D_2, protected by the same mask M. The analysis takes care of two instants t_1 and t_2 of a trace $\mathcal{T}_i$ where $D_{1,i} \oplus M_i$ and $D_{2,i} \oplus M_i$ are respectively manipulated, i being the index of the trace. Area L_0 is a leakage of the mask alone, which can correspond to the abstraction $D_{0,i} = 0$.

In practice, several leakage points for the same share can be found, especially at high sampling rate, as depicted in the trace area L_2. The most encountered leakage model is a linear relationship between the sample values and the Hamming weight of the processed data, but the linear coefficients are not necessarily identical between leakage samples. Indeed, the average μ and standard deviation σ are likely to differ, e.g. $(\mu_2, \sigma_2) \neq (\mu_2', \sigma_2')$.

Additionally, it is important to mention that in a black-box approach, without knowing the mask value for each trace, an analyst cannot easily identify the leaking time samples location. Thus, an educated guess has to be made on the leakage locations L_1 and L_2. This may require to try a lot of different timings to select the right points of interests, increasing the computation effort.

Masked implementations were shown to be leaking the secret $D_1 \oplus D_2$ by combining the corresponding samples together [Mes00]. There are several ways to combine those instants, but for Pearson's correlation, the most effective way is proven in [PRB09] to be the Centered Product where the combination is implemented as follows, with t_1 and t_2 two sample indexes and E the mathematical expectation:

$$\text{CenteredProduct}_i(t_1, t_2) = (\mathcal{T}_i(t_1) - \text{E}[\mathcal{T}(t_1)]) \cdot (\mathcal{T}_i(t_2) - \text{E}[\mathcal{T}(t_2)]) \tag{1}$$

Centering the set requires the computation of a single-sample average, thus a fine alignment of every trace $\mathcal{T}_i$ for both instants t_1 and t_2. In an optimal scenario, the attack is expected to require $\mathcal{O}(N^2)$ traces where N is the number of traces necessary to detect the corresponding first-order leakages [PRB09].

The performance is however decreasing significantly when a fine alignment cannot be achieved. Indeed, the impact is twofold. First, centering the traces is biased since the average trace is no longer accurate. Second, the combined point is incorrectly located when one of the initial sample is misaligned. As a result, the incidence of jitter is quadratically worsened, which explains why alignment is so critical to succeed second-order attacks.

As detailed in Section 2.2, one way to overcome this is to consider window-based approaches.

In another vein, MIA embraces a different approach. As described in [BGP+11], different variants are explored to apply Mutual Information (MI) combining two leakage points. They target the mutual information of the joint leakage between the random variables X_1 and X_2 related to the leakage samples at times t_1 and t_2 and the estimation Y being the random variable of the estimation $D_1 \oplus D_2$. This technique raised a clear theoretical interest but has remained relatively unused in practical testing as CPA showed better result for linear leakages.

2.2 Windowed Time Integration

Making use of the information covered by several time-samples is a central issue in side-channel analysis as it encompass the integration of multiple leakages as well as potential signal misalignment. While not related to second-order attacks, this topic becomes predominant in such context, where the trace dimensions become considerable. Indeed, after combination of two areas of size n and m, the attacks must process traces of size $(n \cdot m)$.

A first method is to average a window of points after the second-order combination. The difficulty is to select the right area. It requires to finely select the areas of interest beforehand, otherwise the non-leaking points turn out to be quadratically overwhelming and drown the information. The choice of a subset is up to the analyst, as it implies to get the right feeling of where the information stands. Finding points of interest for high-order attacks was studied in [RGV12], but the question of a subset suitable for an average remains unresolved.

The issue of $(n \cdot m)$ complexity of second-order attacks was addressed in [WW04] then improved in [BBB+16]. Circular cross-correlation is used over the selected areas. For the cross-correlation, sample combination is processed by multiplying all points from the two vectors and summing the related outcome into a single value. Then, a shift is applied to cover all pairwise combinations. This combination can be computed quickly using a multiplication in the Fourier domain, thanks to the cross-correlation theorem:

$$\text{with two trace slices } L_k[t] = \mathcal{T}_i[t_k + t] \text{ and } |L_k| = n, \quad \forall t \in [0, n[\,,\ \forall k \in \{1, 2\},$$

$$\begin{aligned} C_{\text{x-corr}}(L_1, L_2) = (L_1 \star L_2)[t] &= \sum_{\tau=0}^{n-1} L_1[\tau] \cdot L_2[(\tau + t) \bmod n] \\ &= \sqrt{n} \cdot \text{FFT}^{-1}(\overline{\text{FFT}(L_1)} \cdot \text{FFT}(L_2)) \end{aligned} \tag{2}$$

The paper lacks explanation when dealing with two leakage areas of different sizes, as the circular cross-correlation is only defined to combine two vectors of the same length. To a certain extent, this combination function comes back to the previously discussed averaged CPA, as it implies to average a subset of points after their combination with the multiplication operator.

Therefore, the cross-correlation combination is expected to get much more efficient results if traces are centered beforehand, prior to applying the Fourier transformation. Thus, losing the benefits of the ability to handle misaligned traces.

Other FFT or time-frequency combination variants are proposed in [BBB+16], based on the Discrete Fourier or Hartley Transformation of two areas of interest. The most

effective variants in our practical case were kept:

$$C_{\mathrm{FFT}}(L_1, L_2) = \Re(\mathrm{FFT}(L_1||L_2)) \;\; || \;\; \Im(\mathrm{FFT}(L_1||L_2)) \;\; || \;\; |\mathrm{FFT}(L_1||L_2)| \tag{3}$$

$$\begin{aligned} C_{\mathrm{FHT}}(L_1, L_2) &= \mathrm{FHT}(L_1) \cdot \mathrm{FHT}(L_2) \\ &= (\Re(\mathrm{FFT}(L_1)) - \Im(\mathrm{FFT}(L_1))) \cdot (\Re(\mathrm{FFT}(L_2)) - \Im(\mathrm{FFT}(L_2))) \end{aligned} \tag{4}$$

As explained by the authors, frequency domain techniques, if the phase is ignored, can withstand static desynchronization. Still, more complex misalignment scenarios are not considered in their analysis.

A first limitation concerns frequency domain techniques: the window must be contiguous. This may compromise an attack when dealing with a shuffling or dummy operations protection. Indeed, it may be valuable to select the same points over the different patterns to capture the supposed leakage areas. Taking a non adjacent selection of points makes frequency analysis non sensible.

Secondly, all techniques presented in this section have the same issue: centering has to be done before computing the average, cross-correlation or FFT transformation. Indeed, centering a point requires to know the average value at a given time. Once the transformation applied, this information can no longer be retrieved. This makes the centering impossible to process accurately and explains why these techniques are limited when an alignment cannot be achieved. A practical example is given in Section 4.

In the following, it is shown that converting selection of points into their distribution removes these limitations and extend the attack possibilities on misaligned traces.

2.3 First-Order *Scatter*

As introduced in [TGWC18], the principle behind *Scatter* is to make a selection of points in a trace and exploit their distribution. A distribution is computed by counting how many times a value is found within a given range. Doing so, the information is sorted in bins. Maximal information of a n-bit digitized signal would be achieved with a number of bin $\mathrm{N_{bin}} = 2^n$. For instance, 8-bit data representation gives 256 bins. It is possible to tune down $\mathrm{N_{bin}}$, which gathers the information together and mitigate some noise effect. This is also a way to decrease the memory footprint of the attack.

As a result, leakage and non-leakage samples are integrated together losing the information of their location in time. Consequently, alignment is no longer required within the selection providing that leakages are included.

As with other side-channel techniques, such as MIA, the distinguisher step makes use of partitioning against the estimation to derive a score and highlight the secret. When partitioning, the estimation must be computed with a surjective but non-injective function [SGV08].

The attack starts with the allocation of a set of $|\mathcal{G}| \cdot |\mathcal{H}|$ accumulators, one for each pair of key guess g and estimation h. Then, each trace $\mathcal{T}_i$ is processed as follows:

- select a set of points $\mathcal{S}_i$ within a trace $\mathcal{T}_i$, the location of leakage samples has no incidence within this selection,
- translate the selection $\mathcal{S}_i$ into its corresponding distribution $\mathcal{D}_i$,
- for each key guess $g \in \mathcal{G}$, compute the estimation $h \in \mathcal{H}$ (e.g. the Hamming weight) of the sensitive data using the input data and the key guess, and add the distribution $\mathcal{D}_i$ into the accumulator corresponding to (g, h).

Once enough traces are processed, apply a statistical function and assign a score to each g. The aim is to find the remarkable set of distributions, the outlier, corresponding to the correct key guess. Literature has already explored efficient distinguishers, such as Mutual Information (MI) or Chi-squared (χ^2). In the context of *Scatter* attacks, they both demonstrated the ability to highlight the secret key, even though large selections were made, integrating almost exclusively noisy points.

Doing so, it is possible to overcome alignment issues. Moreover, even when alignment is achievable, *Scatter* simplifies the attack process by performing the attack directly on acquired data without preliminary processing.

However, when facing Boolean masking protections, the technique is no longer effective since the partition against g and h does not fit the manipulated value. Section 3 shows how *Scatter* can be extended to deal with high-order leakages.

3 High-Order Scatter Theory

3.1 Principle

Second-order *Scatter* targets two areas from a single trace $\mathcal{T}_i$ and exploits the information present by computing and processing their joint-distributions. The first step is to make two selection of points $\mathcal{S}_{1,i}$ and $\mathcal{S}_{2,i}$ within $\mathcal{T}_i$. They are chosen so that leakages of $D_{1,i} \oplus M_i$ and $D_{2,i} \oplus M_i$ are likely to be present in $\mathcal{S}_{1,i}$ and $\mathcal{S}_{2,i}$. They can be indifferently located within the selections. It is expected that selections incorporate points non-related to the leakage as well.

As depicted in Figure 5, both $\mathcal{S}_{1,i}$ and $\mathcal{S}_{2,i}$ selections are translated into a distribution. Subsequently, every point of the first distribution is paired with every point of the second one, yielding to a joint-distribution. This is a matrix representing the frequencies of all pairs of values within the selections. In the following, these matrices will be indexed by letters u and v and conveniently considered as images when needed.

Each selection is converted into their distribution. Interestingly, this removes the time representation. As a result, any misalignment within the selection has no influence on the attack. In other words, the resulting distribution is exactly the same regardless the location of all samples in the selection. Even a random permutation of the selected points would yield to the same distribution. This represents a great value for second-order attacks, which are so sensitive to misalignment.

As with other integration techniques, a second benefit lies in the presence of multiple points of leakages. Indeed, $\mathcal{S}_{1,i}$ and $\mathcal{S}_{2,i}$ may include several points related to $D_{1,i} \oplus M_i$ and $D_{2,i} \oplus M_i$ respectively. If they have the same characteristics, they contribute to the same part of the distribution. Otherwise, they are located in different parts. In both cases, an exploitation of multiple points of leakages can be made.

The trace set is processed by accumulating the joint-distribution of each trace $\mathcal{T}_i$ into the Accumulator$_{(g,h)}$ corresponding to the estimation $h = D_{1,i} \oplus D_{2,i}$. This partitioning must be performed for each key guess g.

Partitioning the joint-distributions by their associated estimation intends to emphasize a remarkable piece of information for the correct key guess. This concept will be more detailed in Section 3.3 on distinguishers and is equivalent to resolve an outlier problem.

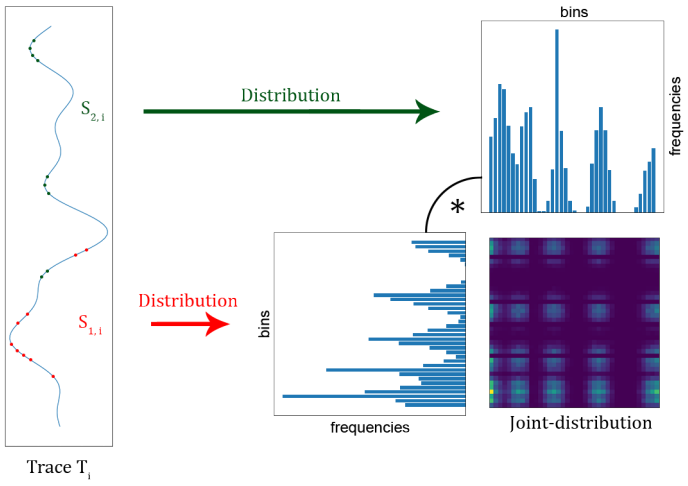


Figure 5: Principle of joint-distribution representation

The whole operation requires the memory allocation of $|\mathcal{G}| \cdot |\mathcal{H}|$ accumulators, respectively the number of key guesses and the number of possible estimations. The memory footprint has a square size $\mathcal{O}(\mathrm{N_{bin}}^2)$. Some optimization can be made by tuning $\mathrm{N_{bin,1}}$ and $\mathrm{N_{bin,2}}$ and better adjust each distribution. We noticed that smaller resolutions significantly accelerated computations without loss of efficiency.

Lastly, the accumulators must be converted into joint-probability density functions which are, by definition, matrices of non-negative real numbers summing to 1. It requires to divide the accumulated joint-distributions by their occurrence count in the trace set. The latter is conveniently contained in the sum of the accumulator itself:

$$\mathrm{pdf}_{(g,h)}[u,v] = \frac{\mathrm{Accumulator}_{(g,h)}[u,v]}{\sum_u \sum_v \mathrm{Accumulator}_{(g,h)}[u,v]} = P((X_1,X_2)=(u,v) \mid Y=h)$$

The proportion of traces accumulated in partition h is then:

$$\forall g \in \mathcal{G}, \quad P(Y=h) = \frac{1}{|\mathcal{T}|} \frac{1}{|\mathcal{S}_1| \cdot |\mathcal{S}_2|} \sum_u \sum_v \mathrm{Accumulator}_{(g,h)}[u,v]$$

with $|\mathcal{T}|$ the number of traces in the set and $|\mathcal{S}_1|, |\mathcal{S}_2|$ the number of points in both selections.

This description is given for a second-order analysis. Straightforwardly, the same can be applied to higher orders by dealing with n-dimensional joint-distributions, partitioned according to the estimation of $D_1 \oplus ... \oplus D_n$ under any leakage model.

3.2 Analysis in Common Cases

In the following section, the paper is elaborated assuming a classical 8-bit Hamming weight leakage to give a visual example of the accumulation step outcome. Other models could be explored similarly, as the attack is not tied to any particular one.

Figure 6 shows the distribution of the leakage points in a perfect Hamming weight model: the traces contain only two samples which are exactly the Hamming weight of the

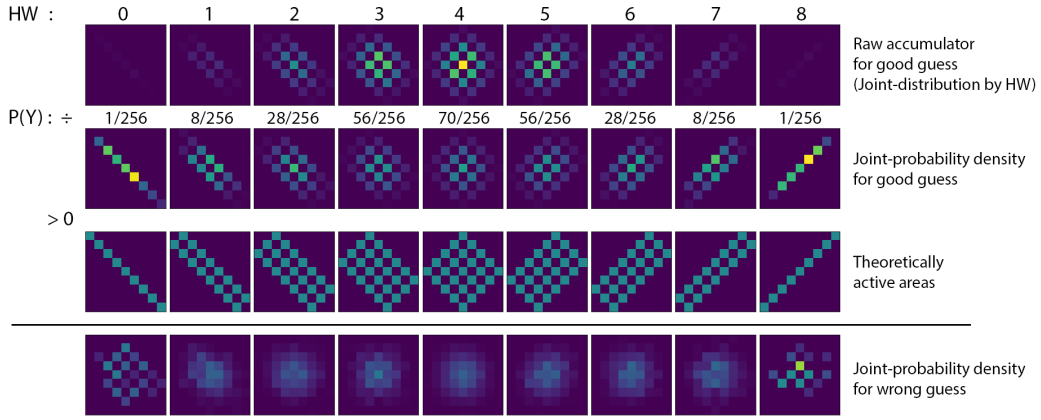


Figure 6: Theoretical joint-distribution and probability density and comparison of the secret key with a wrong guess

mask and of the masked data, standing between 0 and 8. The color intensity on each row is chosen according to the extreme values over all partitions. Interestingly, different patterns can be observed. They are highlighted on the third row, showing a particular shape for each Hamming weight, with a symmetrical property against the central estimation. Appendix A details a formal proof of this distribution, which was confirmed by simulations.

However, a wrong partition related to an incorrect guess results in blurry joint-distributions, as shown in the last row. This shows that remarkable shapes can be highlighted for the right key guess compared to all other key guesses.

These primary shapes represent the theoretical joint-probability density of the expected leakage. Still, in practice, it is usual to select several points of leakage. Extending this representation to multiple pairs of leakage reveals the same shapes with different characteristics:

- Duplicated: p leakage samples in the first selection, and q in the second, generate $p \cdot q$ patterns.
- Shifted towards the index (μ_1, μ_2), corresponding to the average value of respective leakage samples.
- Scaled in (x, y) directions according to the leakage samples standard deviations (σ_1, σ_2) respectively.
- Blurred as a result of trace variations such as noise.

To illustrate these multiple transformations in Figure 7, simulated traces were used containing a carrier signal and two leaking points in each selection. The carrier brings a variable sample average, and leakage points are set with one having a double amplitude compared to the others. Signal to Noise Ratio (SNR) is chosen high enough to witness the good guess by eye.

In the following example, most shapes are not overlapping to each other. The meaningful patterns are visible on the different corners, and their properties are related to the chosen average and variance for the leakage points.

From these observations, we can conclude that remarkable features can be extracted when the correct key guess led to a good partitioning. The features are multiple when

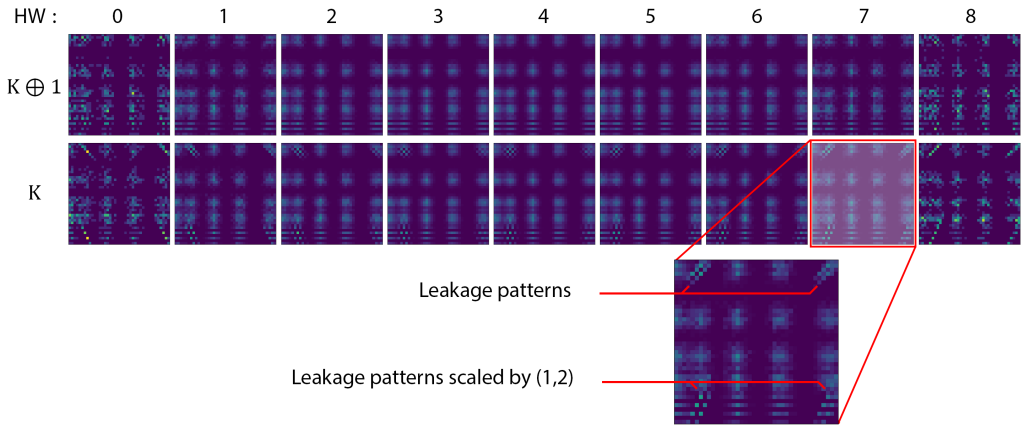


Figure 7: Joint-probability density function for wrong (top) and good (bottom) guesses of our simulated scenario

different leakage points were selected. To identify the right key value over all other guesses requires to solve an outlier problem, which makes use of an efficient statistical function, commonly named distinguisher.

3.3 Distinguishers

Resolving an outlier problem means to look for the most remarkable $\text{pdf}_{(g,h)}$ over the key guesses $\mathcal{G}$ for a given h. As in [TGWC18], the problem can be expressed by introducing a distance $D(g,h)$. In this context, the vector space has the joint-distribution dimension $\text{N}_{\text{bin}}{}^2$, and the guesses data set is composed of $|\mathcal{G}|$ points. For example, there are 256 points in a vector space of dimension $9 \cdot 9 \cdot 9 = 729$ in the perfect Hamming weight leakage.

The distance must be chosen to reach the maximum for the right key guess to the detriment of the wrong guesses. The corresponding distances can be gathered into a single score expressed as:

$$\text{Scatter}(g) = \sum_h P(Y = h) \cdot D(g,h) \tag{5}$$

Where $P(Y = h)$ denotes the occurrence probability of each estimation and is arbitrarily chosen to account for the poor SNR of unlikely observations, such as $\text{HW} = 0$ or $\text{HW} = 8$.

Different distances can be chosen and some suggestions are made in Section 3.3.2. This paper does not aim to provide an in-depth study of their respective performance, particularly because it might highly depend on the practical use case.

The joint-probability density of the entire trace set can be used as a reference point. It aims at describing the background distribution held in the traces. Additionally, it is by construction less noisy than the distribution for each h individually, and is identical for all guesses. In order to compute such background easily, as each trace is processed once per guess, averaging the accumulators for any guess leads to the whole trace set joint-probability density:

$$\forall g \in \mathcal{G}, \quad \overline{\mathrm{pdf}}[u,v] = \sum_{h \in \mathcal{H}} P(Y=h) \cdot \mathrm{pdf}_{(g,h)}[u,v] \tag{6}$$
$$= P((X_1, X_2) = (u,v))$$

3.3.1 Influence of Accumulating Non-Leakage Samples

This section presents a first level of proof that secret information can be extracted from joint-probability density, providing that enough traces are available. And this can be achieved in spite of the non-leakage points influence.

Each probability density function $\mathrm{pdf}_{(g,h)}[u,v]$ is split in leakage ($\mathcal{L}$) and non-leakage ($\mathcal{O}$) distributions:

$$\mathrm{pdf}_{(g,h)}[u,v] = \mathcal{L}_{(g,h)}[u,v] + \mathcal{O}_{(g,h)}[u,v]$$

Using equation (6):

$$\forall g \in \mathcal{G}, \quad \overline{\mathrm{pdf}}[u,v] = \sum_{h' \in \mathcal{H}} P(Y=h') \cdot \mathcal{L}_{(g,h')}[u,v] + P(Y=h') \cdot \mathcal{O}_{(g,h')}[u,v]$$

Then the difference between each partition and the background is:

$$\mathrm{pdf}_{(g,h)}[u,v] - \overline{\mathrm{pdf}}[u,v] = \mathcal{L}_{(g,h)}[u,v] - \sum_{h' \in \mathcal{H}} P(Y=h') \cdot \mathcal{L}_{(g,h')}[u,v] + \tag{7a}$$
$$\mathcal{O}_{(g,h)}[u,v] - \sum_{h' \in \mathcal{H}} P(Y=h') \cdot \mathcal{O}_{(g,h')}[u,v] \tag{7b}$$

Focusing on the non-leakage samples influence in equation (7b), the contribution of h can be extracted from the sum to obtain two independent parts:

$$\varepsilon_{(g,h)}\ [u,v] = (1 - P(Y=h)) \cdot \mathcal{O}_{(g,h)}[u,v] - \sum_{h' \neq h} P(Y=h') \cdot \mathcal{O}_{(g,h')}[u,v]$$

By nature, the non-leakage samples are independent from any guess or intermediate value. Thus, their estimation converges to an arbitrary distribution:

$$\forall (g,h) \quad \mathcal{O}_{(g,h)}[u,v] \underset{|\mathcal{T}| \to +\infty}{\longrightarrow} \overline{\mathcal{O}[u,v]}$$

As the limit of a sum is the sum of its limits, it can be deduced:

$$\forall (g,h), \quad \forall (u,v) \quad \lim_{|\mathcal{T}| \to +\infty} \varepsilon = (1 - P(Y=h)) \cdot \overline{\mathcal{O}} - \sum_{h' \neq h} P(Y=h') \cdot \overline{\mathcal{O}}$$
$$= P(Y \neq h) \cdot \overline{\mathcal{O}} - \overline{\mathcal{O}} \cdot P(Y \neq h)$$
$$= 0$$

On the other hand, distributions $\mathcal{L}_{(g,h)}[u,v]$ converge towards different values depending on g and h. This is illustrated in Appendix A for the correct key guess. Therefore, equation (7a) does not converge to zero.

As a result, the estimation error tends towards zero as the number of traces grows ($\varepsilon \underset{|\mathcal{T}| \to +\infty}{\longrightarrow} 0$). This means that non-leakage samples influence is eventually lower than the leakages samples contribution. This allows an attacker to make use of this difference in

any distinguisher.

In the next sections, the validity of the suggested distances is tested on the previous simulation depicted in Figure 7, but with higher noise and a single leakage point in the selection. The overall noise level is chosen so that the worst distinguisher results are barely capable of finding the secret key.

For each distinguisher, *Scatter* attack is applied and the score for all 256 guesses is respectively depicted on the corresponding figure. The right guess appears in orange. To unify the scales, we decided to plot the deviation to the average score.

3.3.2 Examples of Known Distinguishers

Pearson's χ^2 was introduced to evaluate the likelihood of observing differences between sets solely by chance. It was shown to be an efficient way to detect the secret key distribution as an outlier in a first-order *Scatter* attack. It naturally applies to the second-order case as a dimensional extension. The χ^2 statistic corresponds to the Euclidean distance between the observed distribution and the set centroid, weighted by the expected value:

$$D_{\chi^2}(g,h) = \sum_u \sum_v \frac{(\mathrm{pdf}_{(g,h)}[u,v] - \overline{\mathrm{pdf}}[u,v])^2}{\overline{\mathrm{pdf}}[u,v]}$$

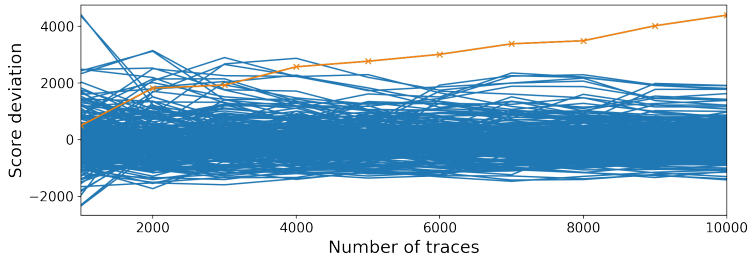


Figure 8: Result of χ^2 distinguisher on our simulation scenario

Mutual Information was proven to be meaningful in [BGP+11] and measures the difference of entropy between the joint-probability density and the its average representation:

$$\begin{aligned} D_{\mathrm{MI}}(g,h) &= \sum_u \sum_v \mathrm{pdf}_{(g,h)}[u,v] \cdot \log(\mathrm{pdf}_{(g,h)}[u,v]) - \overline{\mathrm{pdf}}[u,v] \cdot \log(\overline{\mathrm{pdf}}[u,v]) \\ &= \frac{1}{P(Y=h)} .(H(X_1,X_2) - H((X_1,X_2)|Y=h)) \end{aligned}$$

Please note that this expression is not a valid distance from a mathematical point of view: it leads to the Mutual Information when recombining over h as in Formula (5).

3.3.3 Pattern Detection

Unlike χ^2 and Mutual Information, 2D-convolution kernels are introduced as a new kind of distinguisher. Widely used in the image processing field, they are beneficial to extract specific features in the joint-probability representation. This is particularly valuable when multiple points of leakage lead to numerous shapes scattered in the image, as shown in

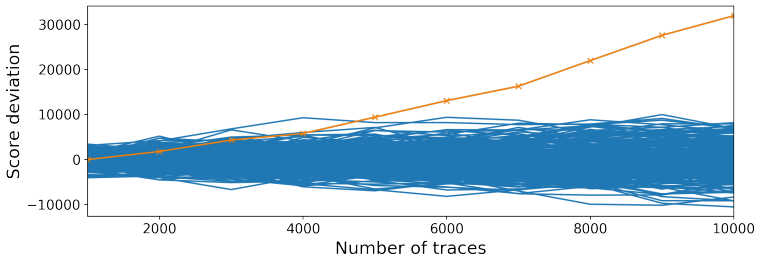


Figure 9: Result of Mutual Information distinguisher on our simulation scenario

Section 3.2 and Figure 7.

One major advantage of the convolution is to work only on local areas of the image, which are defined by the kernel size. It has the ability to capture leakage features and reject those emerging from wrong guesses or from non-leakage samples.

The choice of convolution kernels is specific to the leakage model. In a first approach, we found valuable to use a sharpening kernel on the accumulators since leakage patterns can be seen as a checkerboard:

$$P_{sharpened} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} * \text{pdf}_g, \quad D_{sharpness}(g) = \sum_u \sum_v |P_{sharpened}[u, v]|$$

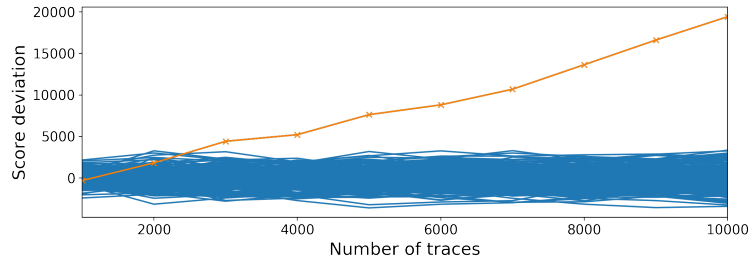


Figure 10: Result of Sharpness distinguisher on our simulation scenario

Such pattern detection does no longer rely on a distance from the background, taking away noise from the estimation of $\overline{\text{pdf}}$. In practice however, the leakage pattern is unlikely to fit the kernel size, and may result to poor detection levels. To take into account various leakage amplitudes and resolutions N_{bin}, the focus was made on diagonal lines in the joint-distribution. The choice of diagonals is made based on the observations of expected shapes in Section 3.2. Good results were obtained with an edge detection algorithm based on a Sobel Filter, defined as two 2D-convolutions with the following kernels:

$$G_x = \begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{bmatrix} * M \quad \textit{and} \quad G_y = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{bmatrix} * M$$

$\vec{\boldsymbol{G}} = (G_x, G_y)$ represents the horizontal and vertical line magnitude in the matrix M. In other words, the outcome of these two convolutions is a vector field, representing the strength and direction of lines identified at every pixel of our image.

Applied to our joint-probability density function pdf_g, the Sobel filtering rearranges the data in order to enhance lines and edges inside the joint-distribution. Summing over the filter output assesses the "quantity of lines" present in the guess distribution, and highlight the valuable information held by the leakage patterns:

$$\text{with } M = \text{pdf}_{(g)}, \quad D_{sobel}(g) = \sum_u \sum_v \sqrt{G_x[u,v]^2 + G_y[u,v]^2}$$

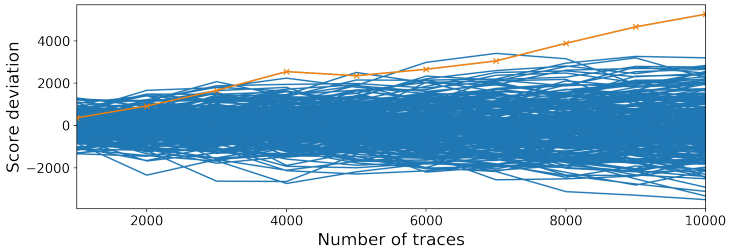


Figure 11: Result of Sobel distinguisher on our simulation scenario

Section 3.4 shows that it could be greatly improved by a specific projection to prioritize line directions fitting the leakage shape.

3.4 Preprocessing

A drawback of working with joint-probability density function is that information is scattered over high-dimensional representations. Indeed, the meaningful information is spread over different parts of the distribution and can therefore be hard to identify.

The role of preprocessing techniques is to gather as much information as possible and therefore to better focus on the leakage area.

3.4.1 Centering / Standardization

Centering (Standardization) is defined by removing the mean (and dividing by the standard deviation) at each trace sample. In the case of HO-CPA, these techniques are used to get consistent properties for both points, whereas in a *Scatter* attack, centering shifts each pair of points in the center of the joint-distribution. In the case of standardization, the points displacement between observations, induced by noise or information leakages, share the same reference amplitude.

It has the minor drawback to concentrate all noise in the same place. But on the positive side, it allows to refine the joint-distribution resolution N_{bin}. Indeed, this operation condenses the leakage of several points of interest, having different mean and standard deviation (μ, σ), in the same place, which might be highly advantageous in the context of non linear distinguishers (Euclidean distance, Mutual Information, ...).

In practice, this preprocessing brings value, as it increases the leakage contrast, making it easier to detect. However, it still requires the traces to be aligned in order to be efficient.

3.4.2 Projection

The projection described here is a method specifically designed to enhance the information contained in joint-distributions using the knowledge of the expected patterns. Although it is valuable to increase *Scatter* performance, the same could be applied to second-order MIA.

Assuming a linear leakage model, the joint-distribution is composed of multiple diagonal lines, whose direction are h-dependent. Indeed, the diagonals does not have the same direction when $h = 1$ or when $h = 7$. A way to better capture the information is to project the joint-distribution elements along a given axis. The projection axis is chosen according to the expected leakage model. The projection translates the two-dimensional matrix into one-dimensional vectors, that can be processed using first-order distinguishers.

Figure 12 illustrates the principle of a projection with a 45° angle. As a result of projection, the leakage pattern is supposed to be condensed and better overcome noise. Projecting along a given line is relevant regardless the location of the leakage pattern within the joint-distribution. As a result, it is expected to gather the information into a smaller number of bins, and consequently improve the distinguishability.

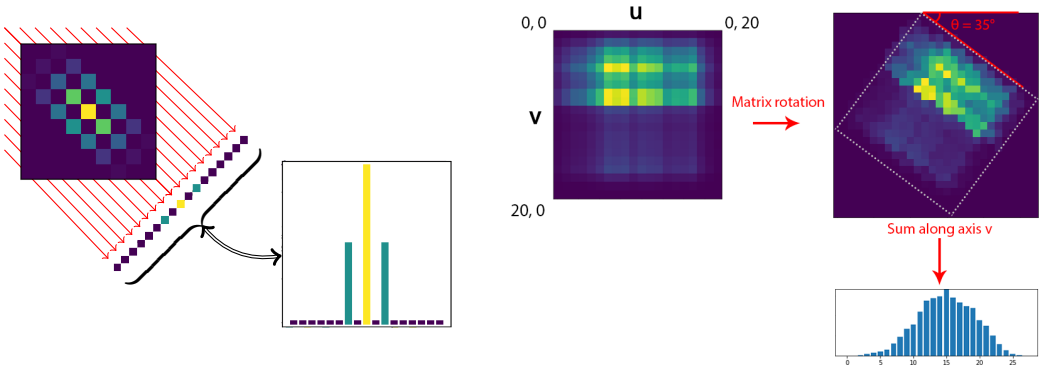


Figure 12: Principle of projection for dimensionality reduction, (left) at 45°, (right) at an arbitrary 35° angle

The projection angle can be fine-tuned for better fitting the leakage pattern. Figure 12 illustrates a projection for an arbitrary joint-distribution with a 35° angle. This is valuable when both leakage samples hold different strengths. In other words, this happens when the respective information behave with different standard deviations σ_1 and σ_2. It is always better to start with a first projection angle of $\theta = 45°$, assuming a similar leakage strength. In a second stage it can be readjusted by exploring empirically other values, or assessing the leakage variances.

Sobel distinguisher must be combined with another kind of projection. As mentioned earlier, it has a vectorial representation of the line direction, as it is composed of (x, y) features. The projection can therefore be achieved by computing an inner-product with a

projection vector:

$$\text{with } M = \text{pdf}_{(g,h)}, \quad \vec{\boldsymbol{G}}[u,v] = (G_x, G_y)[u,v] \text{ its Sobel vector field,}$$
$$\vec{p} = (p_x, p_y) \text{ a projection axis}$$

$$D_{\vec{p},Sobel}(g,h) = \sum_u \sum_v \frac{|\vec{\boldsymbol{G}}[u,v] \cdot \vec{p}|}{||\vec{p}||}$$
$$= \frac{1}{\sqrt{p_x^2 + p_y^2}} \sum_u \sum_v |p_x.G_x[u,v] + p_y.G_y[u,v]|$$

In the same way, the projection can be fine-tuned for an optimal result. By light computational steps, it is possible to apply different projection vectors to find the best ratio. Indeed, once $\vec{\boldsymbol{G}}$ is computed, any angle can be tested thanks to various projection vectors (p_x, p_y). Alternatively, one can try to estimate the ratio between σ_1 and σ_2 leakages by looking directly at the estimated line angles $\theta = \text{atan}(\frac{G_x}{G_y})$.

The next sections illustrate the interest of the projection. We noticed that projection brought a significant improvement in most of the testing we performed.

3.5 Influence of Non-leakage Points on Score

The selection size has a direct impact on the attack performance. Indeed, integrating non-leaking points creates a noise that may overwhelm the valuable information if not enough traces were processed. This is a particularly sensitive issue for a second-order attack integrating pairs of points. Figure 13 explores the attack performance on simulated traces for different distinguishers, and projected variants.

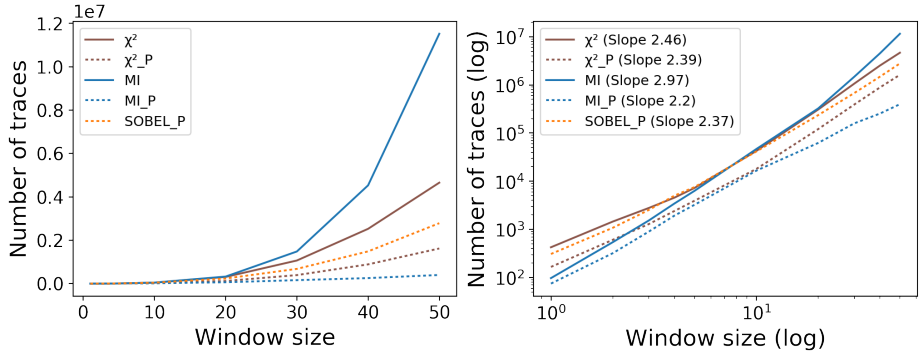


Figure 13: Influence of the integration size ("_P" and dashes indicates the use of projection)

It gives the number of traces necessary to recover a secret byte against the window size w. The traces are built such that only one pair holds the secret information. This means that one point holds the Hamming weight of the mask M within a window of size w and one point holds the Hamming weight of the estimated value $D \oplus M$ within another window of size w. All other points are set to the Hamming weight of random bytes. Computing the joint-distribution consists therefore of accumulating w^2 pairs. The same results are represented in linear (left) and logarithm (right) scales. The latter provides an indication of the order of the best fitting polynomials as linear regression slopes.

One can note that for $w = 1$, the Mutual Information distinguisher is a classical MIA, as only the leaking pair is accumulated. The influence of the window size w has a near square factor. This is expected as the noise related to the non-leakage samples represents $w^2 - 1$ pairs. Besides, the performance of the different distinguishers can be noted: the projection has a steep positive impact. Even though the respective performance of the distinguishers with a projection differs, this should not be taken for granted as it is highly tied to the use case. The practical example in Section 4 illustrates this.

4 Practical Results

The objective of this section is to confirm the validity on proposed techniques on a practical use case. Moreover, it provides a practical evidence that second-order *Scatter* brings value compared to other techniques from the state-of-the-art. For this, different tests were run on the same trace set to make a fair comparison.

The testing was performed on a mono-core 32 bits secure device. A set of traces was collected with a near-field electromagnetic probe located at the surface of the die with a sampling rate of 2.5 GSamples per second. The device implements hardware random jitter. The algorithm is an AES-128 software implementation with masking countermeasures preventing first-order attacks. The SubBytes operation is protected by a random mask value generated prior to the execution. For each encryption, a new random is generated and the masked SBox tables computed accordingly. Both leakage areas, the random mask and the output of the SubBytes operation, are identified. No first-order leakage was found but a linear relationship between the Hamming weight of the masked data/mask and the trace values can be observed. The sampling rate has been chosen above the clock so that several leaking samples are present in a cycle.

4.1 Validating the Distinguishers

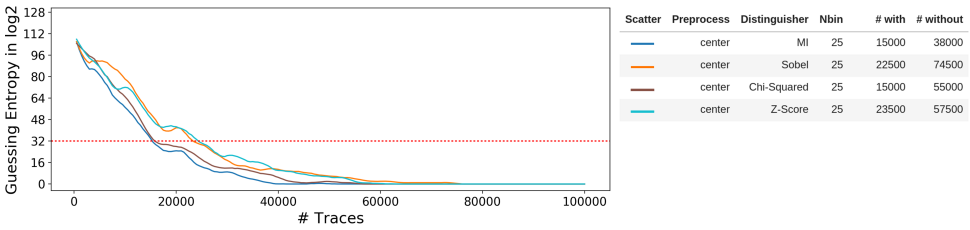
The first test aimed at validating that the technique behaves as expected from a practical point of view. The different distinguishers are applied on the same trace set on aligned and centered data.

Figure 14 captures the results. Two selections of 100 points have been chosen. The number of traces required to find the key *without* exhaust or *with* a 2^{32} remaining guessing entropy indicates how fast the key is retrieved. An exhaust of 2^{32} is considered achievable as it requires less than a minute of bruteforce on a classical desktop computer.

These results confirm the validity of the distinguishers. All of them show similar performance. For the sake of simplicity, Sobel and Mutual Information distinguishers will be kept for the rest of the paper. This choice is justified by the fact that Sobel is a new distinguisher introduced in this paper and comes with a specific projection. Mutual Information is a reference, as exploited by the MIA.

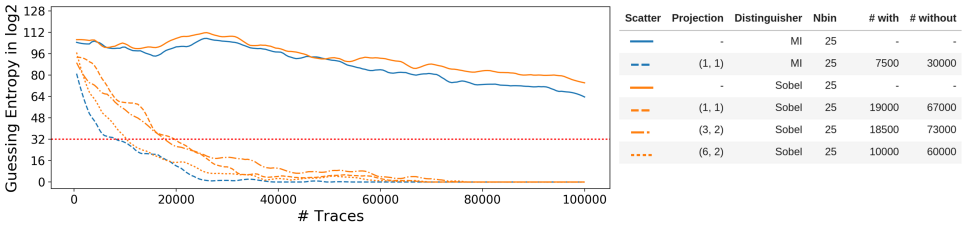
4.2 Projection

The second test was done to confirm the positive impact of the projection. Assuming a linear leakage model, projection was applied together with MI distinguisher. It resulted an 1-dimension vector for the MI distinguisher to process. On the other hand, Sobel distinguisher was associated to its related projection using different angles which kept the 2-dimensional representation.



Scatter	Preprocess	Distinguisher	Nbin	# with	# without
	center	MI	25	15000	38000
	center	Sobel	25	22500	74500
	center	Chi-Squared	25	15000	55000
	center	Z-Score	25	23500	57500

Figure 14: Comparison of distinguishers on a centered trace set



Scatter	Projection	Distinguisher	Nbin	# with	# without
	-	MI	25	-	-
	(1, 1)	MI	25	7500	30000
	-	Sobel	25	-	-
	(1, 1)	Sobel	25	19000	67000
	(3, 2)	Sobel	25	18500	73000
	(6, 2)	Sobel	25	10000	60000

Figure 15: Impact of projection preprocessing on *Scatter* performance

Testing were done without centering the traces, as the main objective is to handle misaligned traces. The same two selections of 100 points were chosen. Figure 15 reveals that the projection is effective to reduce the overall guessing entropy. Using Sobel, the best projection seems to be along the vector (6,2), which expresses that both mask and masked value leakages have different properties. In the following, this angle will be kept for Sobel projections. Mutual Information is projected with a 45° angle.

These results first highlighted the impact of centering the traces on this use case. The attack performance are significantly downgraded. Without projection, the secret key is no longer retrieved using the trace set. However, applying the projections related to MI and Sobel respectively improve significantly the results. They can be slightly improved by tuning the projection angle.

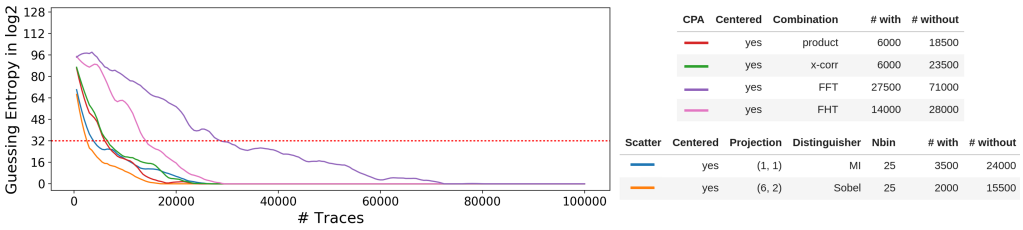
This confirms that the projection wears an important part of the attack performance. It will be systematically used in the following of this paper.

4.3 Comparative Study with Misalignment

Subsequent testing was performed to compare second-order *Scatter* techniques with others from the state-of-the-art, typically combination with the cross-correlation ($C_{\text{x-corr}}$), the FFT (C_{FFT}) and its variant FHT (C_{FHT}). Except for a straightforward CPA, the attacks were done with two windows of 100 adjacent points, which means 10^4 pairs of points. Traces were aligned and centered as well. Figure 16 summarizes all the results. It confirms that all techniques work well on this best-case scenario of aligned and centered traces. They all lead to the secret key extraction within the same range of performance.

It can be noticed that Sobel with the right projection performs even better than CPA, while the other integration techniques perform worse. It could be that *Scatter* is able to better leverage multiple points of leakage, having different distributions.

Then, the same test was performed without centering the traces. Figure 17 confirms that centering is necessary for most of the state-of-the-art attacks, including the classical

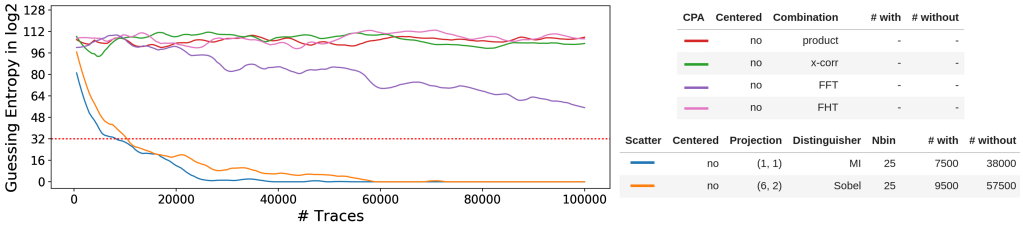


CPA	Centered	Combination	# with	# without
	yes	product	6000	18500
	yes	x-corr	6000	23500
	yes	FFT	27500	71000
	yes	FHT	14000	28000

Scatter	Centered	Projection	Distinguisher	Nbin	# with	# without
	yes	(1, 1)	MI	25	3500	24000
	yes	(6, 2)	Sobel	25	2000	15500

Figure 16: Comparative results on centered traces (jitter = 0)

CPA and the window-based techniques. C_{FFT} shows a slightly better behavior compared to C_{FHT} and $C_{\text{x-corr}}$. Only *Scatter* techniques led to the secret key extraction, MI and Sobel giving comparable results.



CPA	Centered	Combination	# with	# without
	no	product	-	-
	no	x-corr	-	-
	no	FFT	-	-
	no	FHT	-	-

Scatter	Centered	Projection	Distinguisher	Nbin	# with	# without
	no	(1, 1)	MI	25	7500	38000
	no	(6, 2)	Sobel	25	9500	57500

Figure 17: Comparative results on non centered traces (jitter = 0)

Adding a span of jitter helped us to see how the techniques behave on misaligned traces. Misalignment was created by an uniform random time shift of 50 points applied on the traces. We can note that more complex misalignment could have been applied. They would have been detrimental for frequency domain techniques and would not have any influence on *Scatter* attacks, as explained in Section 3.1.

In that case, it was not possible to center the traces accurately. Looking at the previous test results for C_{FFT}, C_{FHT} or $C_{\text{x-corr}}$ techniques, it was preferable to keep centering the traces, even though this operation is biased due to poorly aligned traces. However, no centering was applied to *Scatter* attacks.

Because the leakage was moving, it was chosen to increase the number of integration points. Each selection was made of 150 adjacent points. Figure 18 displays the tests results. In this comparison, CPA is discarded as this technique does not make much sense when traces are not aligned.

The outcome shows that all techniques are exploitable when using enough traces. But *Scatter* techniques perform significantly better than all other window-based techniques on this use case. The misalignment had almost no impact on *Scatter* performance.

4.4 Second-order *Scatter* Benefits and Further Work

Based on the practical results shared in this paper, different advantages of using *Scatter* can be highlighted:

- The attack does not require a fine alignment. This is a strong added-value when alignment cannot be achieved due to the nature of the traces.

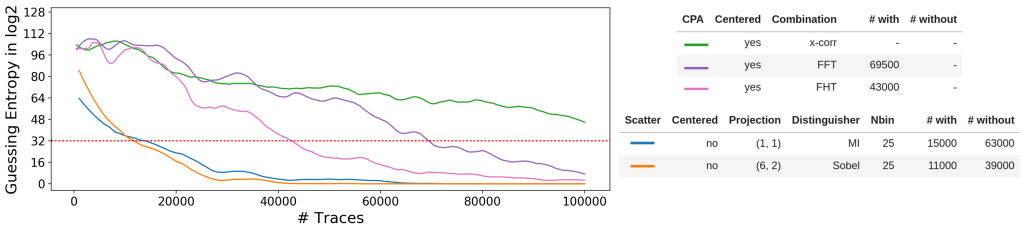


Figure 18: Comparative results of integration techniques (jitter = 50)

- Conducting a practical second-order can be made simpler, as no alignment is required for a reasonable processing effort.
- The technique works well to exploit multiple leakages by integrating several points of interest in the same joint-distribution.
- Hardware jitter may no longer represent a valuable protection in some cases. Countermeasures, such as shuffling, can be potentially defeated.

It is however worth mentioning that further work may be required to apprehend this technique in a more comprehensive way. A deeper formalization effort may help to better understand the impact of the non-leakage points depending on their numbers and characteristics. A second axis of work concerns the ability of the technique to combine several points of information present in the same trace: it would be valuable to estimate how much the integration of different points of leakage improves the attack results.

From testing prospective, it would deserve further practical validations to explore how much it defeats existing protections, particularly hiding countermeasures like shuffling or dummy operations.

5 Conclusion

This paper develops an extension to *Scatter* attacks to perform high-order analyses and offers a strong alternative to the state-of-the-art when dealing with complex signals. This is particularly valuable when traces are hard or excessively time-consuming to align. The simulated and practical results confirm that these new attacks achieve a similar efficiency as the best existing techniques, and is interestingly slightly better when no jitter is implemented. When alignment cannot be achieved, *Scatter* performs significantly better than other techniques.

With this new second-order attack, we show that another approach can be used to break protected cryptographic algorithms, particularly when facing misalignment. The paper develops different techniques showing that dealing with joint-distribution over selections of point can offer a large space of attack and research opportunities. Particularly, the preprocessing was shown to be a critical part of the attack. *Scatter* concept is flexible and we believe there is room for improving the techniques described in this paper.

Finally, we suggest to consider *Scatter* attacks when implementing or assessing secure cryptographic libraries claiming second-order resistance, either for software or hardware implementation.

References

[BBB+16] Pierre Belgarric, Shivam Bhasin, Nicolas Bruneau, Jean-Luc Danger, Nicolas Debande, Sylvain Guilley, Annelie Heuser, Zakaria Najm, and Olivier Rioul. Time-Frequency Analysis for Second-Order Attacks. *IACR Cryptology ePrint Archive*, 2016:772, 2016.

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

[CCD00] Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, pages 252–263, 2000.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 398–412, 1999.

[GBPV10] Benedikt Gierlichs, Lejla Batina, Bart Preneel, and Ingrid Verbauwhede. Revisiting Higher-Order DPA Attacks. In *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings*, pages 221–234, 2010.

[GBTP08] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual Information Analysis. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, pages 426–442, 2008.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 388–397, 1999.

[Koc96] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[Mes00] Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 238–251. Springer, 2000.

[PR10] Emmanuel Prouff and Matthieu Rivain. Theoretical and practical aspects of mutual information-based side channel analysis. *IJACT*, 2(2):121–138, 2010.

[PRB09] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.

[RGV12] Oscar Reparaz, Benedikt Gierlichs, and Ingrid Verbauwhede. Selecting Time Samples for Multivariate DPA Attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, pages 155–174, 2012.

[RPD09] Matthieu Rivain, Emmanuel Prouff, and Julien Doget. Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009.

[SGV08] François-Xavier Standaert, Benedikt Gierlichs, and Ingrid Verbauwhede. Partition vs. Comparison Side-Channel Distinguishers: An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices. In *Information Security and Cryptology - ICISC 2008, 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers*, pages 253–267, 2008.

[Sou11] Youssef Souissi. *Optimization methods for side channel attacks. (Méthodes optimisant l'analyse des cryptoprocesseurs sur les canaux cachés)*. PhD thesis, Télécom ParisTech, France, 2011.

[TGWC18] Hugues Thiebeauld, Georges Gagnerot, Antoine Wurcker, and Christophe Clavier. SCATTER: A New Dimension in Side-Channel. volume COSADE 2018 of *Lecture Notes in Computer Science*. Springer, 2018.

[WW04] Jason Waddle and David A. Wagner. Towards Efficient Second-Order Power Analysis. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 1–15, 2004.

A Theoretical Joint-distribution Under Linear Model

Single-Data Leakage

The observed repartition of data is explained by the relationship between the Hamming weight of the mask and the Hamming weight of the masked data. To compute this theoretical joint-probability, we assume uniform distribution of mask M and data D between 0 and 256.

We aim to give the proportions of masks leading to an accumulation in cell $[u, v]$ for any given D, which is literally:

$$\begin{aligned}\mathrm{pdf}_h[u,v] &= P((X_1,X_2)=(u,v)|Y=h)\\ &= \frac{1}{256}\cdot \#\left\{M \mid \left\{\begin{array}{l}\mathrm{HW}(M)=u\\ \mathrm{HW}(D\oplus M)=v\end{array}\right\},\ \forall D,\ \mathrm{HW}(D)=h\right\}\end{aligned}$$

We denote by $x = \mathrm{HW}(D \wedge M)$ the actual number of bits equal to 1 that will be reset during the Boolean masking. Thus:

$$\begin{aligned}\mathrm{HW}(D\oplus M) &= \mathrm{HW}(D)+\mathrm{HW}(M)-2\cdot\mathrm{HW}(D\wedge M)\\ &\Leftrightarrow v = h+u-2x\\ &\Leftrightarrow x = \frac{h+u-v}{2}\end{aligned}$$

We know that x bits over the existing h are flipped from 1 to 0 and also that $(u-x)$ bits over the existing $(8-h)$ are flipped from 0 to 1. When x is valid (i.e. $x \in \mathbb{N}$), the number of combination can therefore be computed, else, there is no valid solution:

$$\mathrm{pdf}_h[u,v] = \frac{1}{256} \cdot \begin{cases} \binom{h}{x} \cdot \binom{8-h}{u-x} & , x \in \mathbb{N} \\ 0 & , x \notin \mathbb{N} \end{cases}$$

Finally, this probability density function corresponds to the shapes depicted on Figure 19.

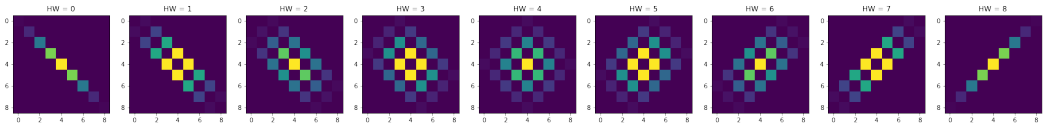


Figure 19: Joint-probability for second-order leakage under the Hamming weight model

Multiple-Data Leakage

Additionally, this paper considers the case of a multiple data leakage $(D_1 \oplus M, D_2 \oplus M)$. Such leakage leads to the exact same pattern as the previously described for single-data leakage, as it can be converted into a single data $D' = D_1 \oplus D_2$ equation by considering a mask $M' = D_2 \oplus M$:

$$(D_1 \oplus M, D_2 \oplus M) \Leftrightarrow (D' \oplus M', M')$$

As a single data leakage equation, it leads to the same shapes of joint-probability.