Spatial Dependency Analysis to Extract Information from Side-Channel Mixtures

Aurélien Vasselle[1,2], Hugues Thiebeauld[1] and Philippe Maurine[2]

[1] eShard, Pessac, France, aurelien.vasselle@eshard.com
[2] Laboratoire d'informatique, de robotique et de microélectronique de Montpellier (LIRMM), Montpellier, France, philippe.maurine@lirmm.fr

Abstract. Practical side-channel attacks on recent devices may be challenging due to the poor quality of acquired signals. It can originate from different factors, such as the growing architecture complexity, especially in System-on-Chips, creating unpredictable and concurrent operation of multiple signal sources on the device. This work makes use of mixture distributions to formalize this complexity, allowing us to explain the benefit of using a technique like Scatter, where different samples of the traces are aggregated into the same distribution. Some observations of the conditional mixture distributions are made in order to model the leakage in such context. From this, we infer local coherency of information held in the distribution as a general property of side-channel leakage in mixture distributions. This leads us to introduce how spatial analysis tools, such as Moran's Index, can be used to significantly improve non-profiled attacks compared to other techniques from the state-of-the-art. Exploitation of this technique is experimentally shown very promising, as demonstrated on two AES implementations including masking and shuffling countermeasures.

Keywords: Side-channel Analysis · System-on-Chips · Mixture distribution · Interaction Information · Spatial Analysis · Moran's Index · ASCAD

1 Introduction

Side-channel attack is a threat to any system implementing cryptographic operations and exposing unintentional leakages related to its internal processing. The development of embedded electronics and IoT translates into a growing number of systems deployed on the field subject to this risk. It concerns different products with various Integrated Circuit technologies, such as secure controllers, ASICs, System-on-Chips or FPGAs. Successful attacks have been demonstrated on different products, such as IoT [RSWO18], smartphones [GPP+16, LMPT15, AKD+18, VMC19] or laptops [GPT15].

Depending on the device, the practical realization significantly differs and may raise several technical challenges. This results in different levels of difficulty to make a successful attack. The first point concerns the source of leakage. Traditionally, attacks leverage physical measurements capturing the internal activity of the chip. Currently, the most efficient technique is often the one using near-field electromagnetic probes, since it is possible to capture a local activity related to a specific part of the circuit, like described in [LMPT15, VMC19]. Indeed, with devices growing in complexity, the whole activity can quickly overwhelm the leakages which are by essence of low amplitude. More recent works has made use of cross channel techniques, and more specifically exploit internal chip sensors. The work of [GDTL20] sketches a potential for remote attack leveraging delay lines available on different SoC devices.

Once the signal is available, the complexity of the devices may significantly affect the quality of the information. This is due to the nature of the device itself, where concurrent and predictive execution creates uncertainties about the exact time of a target operation. In addition, the rest of the circuit generates an activity whose signal can be overwhelming compared to the useful signal. This is even more challenging with fast clocks, where a hardware execution of an algorithm lasts a few nanoseconds only. To these considerations, we shall mention internal protections that can be added by the designer or developers to make side-channel exploitation more tedious.

As a result, it is reasonable to assume that practical side-channel attacks deal with signals of poor quality, due to a lack of alignment or to an overwhelming noise. To cope with this trend, the analysis techniques should evolve. Some propositions have been developed to overcome this. Automatic alignment does not turn out to be very efficient [vWWB11], for the simple reason that it targets the shape of the signal and not the information itself. Some other techniques demonstrated that they can tackle the alignment problem by removing the time factor, at least partially. This is the case of frequency based analyses [BBB+16], distribution based analyses [TGWC18] or deep learning techniques using Convolutional Neural Networks [PSB+18]. They all have their pros and cons, and should certainly all be considered.

In this paper, we aim to better set the problem of attacks dealing with complex signals. We come with a formalization of this complexity, by expressing it into statistical mixtures. This helped us to better characterize the problem and introduce an extension of scatter attacks leveraging spatial analyses. The benefit is demonstrated with practical implementations of second-order multidimensional attacks.

2 Problematic of Mixture Distributions

The acquisition of data from a side-channel results in a collection of traces, associated with the metadata used for each experiment (*e.g.* plaintext and/or ciphertext). The trace set can be statistically represented by a random process (*i.e.* stochastic process), that is a collection of random variables X, defined over a common probability space Ω, indexed by the time dimension T:

$$\{\ X(t,\omega) : t \in T\ \} \tag{1}$$

With t fixed, $X(t,\cdot)$ is a random variable that can be denoted X_t for any instant $t \in T$. With ω fixed, $X(\cdot,\omega)$ is a single trace, also called a realization of the random process.

For most practical attacks, the data is digitized in both dimensions by an oscilloscope. As a result, the random process usually has a discrete-state (*e.g.* 8-bit values) and discrete-time (defined by the sampling rate and the number of time-samples) that approximate the physical and continuous nature of the measurements.

Sometimes, a trace realization does not perfectly correspond to other measurements. For instance, an internal or external signal source may create noise, or a cache miss may delay the valuable information. The environment also plays a role, because a shift in temperature or the stability of power supply are known to significantly reduce the repeatability of side-channel measurements. It mainly causes changes in voltage and frequency of operation, resulting in jitter in horizontal direction and noise in vertical direction. This is especially true when acquiring side-channel data on complex devices such as those embedded in our smartphones.

With eventual side-channel countermeasures on top of this; it appears very unlikely that classical side-channel analyses can be applied without trouble on these complex use cases.

In fact, the distribution of a side-channel variable X_t is rather expected to be a mixture of multiple signals varying from one measurement to the other. In such a case, classical techniques analyzing a single time-sample, and assuming all realizations of X_t originate from the same source, end up poorly operating, because X_t incorporates either non-informative samples, or leakage samples with different characteristics. Consequently, univariate analyses are sub-optimal, because valuable information is often missed, or mixed with uninformative data.

In the following of this paper, we develop a descriptive model of such composite mixture of signals, and show that most distinguishers are not well suited to extract information from them. Then we propose to investigate two alternative approaches to analyze them efficiently. First, as jitter spreads the information over multiple time-samples, we applied the *Scatter* technique [TGWC18], which gathers several sources (*i.e.* X_t) into the same statistical distribution to counteract jitter. Second, as an innovative approach in side-channel, we introduce spatial dependency analysis, which was found efficient to extract information hidden in statistical mixtures. As a benefit, this method allows to counteract the vertical spread of information in the measured distributions.

3 Side-channel Mixture Model

In statistics, latent random variables are not directly observable, but can eventually be inferred from other observable variables via a mathematical or statistical model. When Z is a continuous latent variable, the mixture (or compound) distribution takes the form of:

$$P(X = x) = \int P(Z = z) \cdot P(X = x|Z = z)\, \mathrm{d}z \tag{2}$$

$P(Z)$ is referred to as the mixing distribution, and $P(X|Z = z)$ as a mixture component for source z.

The latent variable of a mixture distribution can help modeling the initial state of side-channel experiments. Two different aspects are taken in consideration.

First, Z can encompass the device micro-architectural state. For instance, it can be associated to the core voltage of the target CPU. $P(X = x|Z = z)$ therefore represents the distribution of the side-channel obtained at a given voltage z. The same can be said about clock frequency, cache and branch prediction status, etc. Besides, external and environmental parameters can influence the observed traces, thus Z can also model temperature or supply voltage variations, etc.

More generally, the mixing variable will be the Cartesian product of each individual effect. Analyzing signals from a device with complex features is equivalent to handling a side-channel mixture of sub-datasets, captured with different initial conditions, scrambled together. Actually, the number of initial states, *i.e.* outcomes of Z, is such that the number of observable configurations of $X(\cdot, \omega)$ can quickly surpass the number of acquired traces, meaning that traces are all different. Yet, in any case, the components of the mixture distribution can either be:

- **Noise or non-informative signal components**, independent of the leaking data,
- **Leakage components**, having possibly different distributions, including different noise.

Most often, leakage components are relatively close to each other, in both time and leakage model, which explains why classical techniques may work, given sufficiently large datasets. But our goal is to show they are sub-optimal in the context of mixture distributions.

4 Mitigating Mixing

Usually, separation of individual components of a mixture is managed manually, with techniques such as signal alignment and sometimes outlier exclusion. These operations often turn out to be tedious, and require human expertise and time, spent for each acquisition. Moreover, this may lead to discard part of the traces, with many false-positive and false-negative. Indeed, the discarded data may contain valuable information, and the remaining dataset may still include non-informative components if the alignment is not perfect.

A particular effort can be made to model horizontal jitter in the measurements as an operation on mixture distribution. The main difference is that jitter shifts samples in time and therefore mixes different neighbor instants of the acquisitions.

Assuming a static misalignment, represented by the random variable J. Each measured trace is shifted by a random offset, drawn from J. This generates a different mixture component for each possible outcome of jitter j. In that case, the random variable X_t associated to the measurement of traces at time t is a mixture of points taken from the aligned traces T (ground truth) in the neighborhood of t.
More precisely, the measured mixture probability density function (pdf) is a convolution of the jitter pdf p_J and the random process pdf, analyzed in the time dimension $T(t, \cdot)$:

$$\begin{aligned} \forall t \in \mathbb{R}, \quad P(X_t = x) &= \int P(J = j) \cdot P(X_t = x | J = j) \, \mathrm{d}j \qquad (3) \\ &= \int P(J = j) \cdot P(T_{t-j} = x) \, \mathrm{d}j \\ p_{X_t} &= (p_J * p_T)(t) \qquad (4) \end{aligned}$$

When the measurements are subject to two independent static jitter sources, the resulting mixing variable is the sum $Z = J_1 + J_2$. Its probability density function is the convolution of both pdfs:

$$\begin{aligned} P(Z = z) = P(J_1 + J_2 = z) &= \int_{-\infty}^{\infty} P(J_1 = t, J_2 = z - t) \, \mathrm{d}t \qquad (5) \\ &= \int_{-\infty}^{\infty} P(J_1 = t) \cdot P(J_2 = z - t) \, \mathrm{d}t \\ &= (p_{J_1} * p_{J_2})(z) \end{aligned}$$

4.1 *Scatter* Approach

To avoid separation of leakage components and enable the analysis of the whole leakage usually spanned over several consecutive samples, *Scatter* technique [TGWC18] offers an interesting alternative leveraging mixtures. Its main rationale comes from the observation that measured side-channel traces can be too complex to be finely aligned and efficiently analyzed with statistical tools manipulating only unidimensional variables, like the correlation coefficient. An information spread over time-samples cannot always be identified precisely. Therefore, univariate analyses are losing exploitable information when not considering several time-samples in the attack. This means that the adversary needs to make an educated guess to find out where the main part of the information stands, and incorporates multiple samples of each trace into a mixture distribution estimate, based on its selection of points of interest.

In the mixture model, the integration of points boils down to crafting a mixture of hand-selected components. To that end, let S be the latent variable representing the index within the dataset from which the sample is drawn when selected.

A simple *Scatter* selection can for example be a window of consecutive points around the presupposed leakage area, having a width approximately equal to the observed jitter span. This results in incorporating leakage samples together with non-informative samples. It can be valuable to analyze the effect of such a selection on the resulting mixture.

It can be observed on a simple case with a static misalignment, *i.e.* a simple trace shifting. The jitter J and point selection S are independent. The resulting mixing variable is $Z = J + S$. Leveraging Equations (4) and (5), the random process yields:

$$\forall t \in \mathbb{R}, \quad P(X_t = x) = (p_Z * p_T)(t, x) = (p_S * p_J * p_T)(t, x) \tag{6}$$

and the mixing distribution after *Scatter* transform is:

$$P(Z = z) = \int P(S = t) \cdot P(J = z - t)\,\mathrm{d}t \tag{7}$$

In other words, *Scatter* can be seen as an augmentation of measurement realizations by adding jitter to the traces. In that case, knowing the jitter profile, it is possible to compute the probability that a point ends up in the selection, and thus the influence it has on the transformed distribution at time t.

Figure 1 illustrates this relationship with a jitter profile composed of two Normal distributions, associated with a continuous *Scatter* window selection of width 12. The resulting probability estimate has a flatter mixing distribution, meaning that samples are more equally represented. Moreover, the estimate is done with 12 times more samples. The main drawback is the number of different components, increasing slightly due to border effects of the integration. This is especially visible for components 6 to 10 that have a high probability to be incorporated in the scatter sample selection.

Intuitively, if there were 10 leakage samples in this example, the transformation would bring a lot of information to the expense of a little bit of noise components in the mixture. The benefits of this technique in practice are later discussed in Section 7.

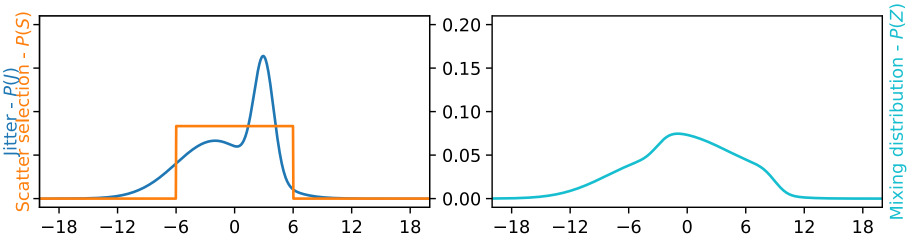


Figure 1: Effect of a *Scatter* window on a mixture subject to arbitrary jitter. (left) Probability distributions of jitter and *Scatter* selection. (right) Resulting mixing distribution, *i.e.* distribution of mixing variable Z.

Overall, when facing mixture distributions with complex jitter, alignment quickly becomes tedious and time consuming, if not impossible. In that case, *Scatter* is a valuable tool to transform the mixing proportions, and increase the number of samples available.

Nevertheless, with or without this preliminary transformation, extracting information from mixture distributions with classical side-channel distinguishers can be troublesome. In the next section, we show that most distinguishers are not suited to mixture analyses, and propose an enhanced processing specifically designed for this context.

5 Classical Distinguishers' Limitations

Once enough traces are acquired and processed, one has to apply a distinguisher, assigning a score to each key candidate. In order to discriminate the secret key among all possible candidates, the distinguisher should be chosen to reach maximum score for the right key guess to the detriment of the wrong guesses.

In practice, the mixtures are sometimes comprised of samples close to each other, usually having strong linear relationships that can be exposed through cross-correlation. In that scenario, classical side-channel distinguishers such as Pearson's Correlation are supposed to continue operating correctly while withstanding a small amount of jitter.

In the following, we illustrate that even assuming there are only leakages components, looking a lot alike, the mixture of their distributions might not be easily exploitable with such distinguisher. To that aim, let us consider a mixture of two distinct linear Hamming weight leakages:

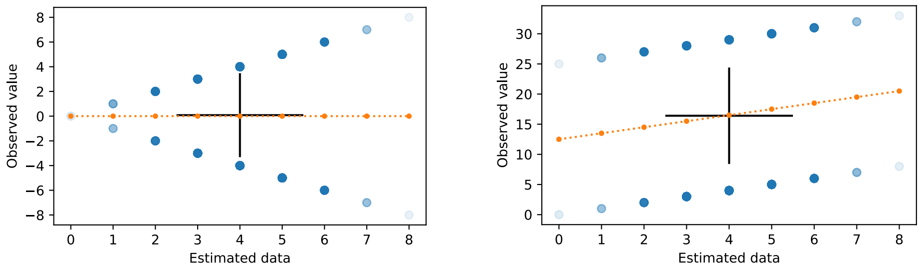
$$P(X) = \lambda \cdot P(T_0) + (1-\lambda) \cdot P(T_1) \tag{8}$$

$$\forall i \quad T_i = \alpha_i \cdot \mathrm{HW}(L) + \beta_i + \sigma_i \qquad \text{with } (\alpha_0, \beta_0, \sigma_0) \neq (\alpha_1, \beta_1, \sigma_1) \tag{9}$$

with two random variables associated to a leaking data L, and arbitrary noise σ_i, where α_i, β_i are the coefficients of the linear relationships and λ the mixing coefficient.

Using this example, it appears that cases such as $\alpha_0 = -\alpha_1$, if equally represented in the mixture (*i.e.* $\lambda = 0.5$), completely annihilates, by symmetry, the information carried by the first statistical moment. As depicted in orange in Figure 2a, the observed conditional average value will be equal for any HW, resulting in failure of classical distinguishers, including: correlation (CPA [BCO04]), as well as F-test variants (ANOVA [SGV08], NICV [BDGN13], SNR), and the linear regression (LRA [DPRS11]).

This can occur as soon as positively correlating leakages are followed by negatively correlating ones, and jitter creates a mixture of both.



(a) $\alpha_0 = -\alpha_1$

(b) $\beta_0 = \beta_1 + 25$

Figure 2: Two mixtures of noiseless leakages, linear in Hamming weight, difficult to exploit with classical distinguishers.

A more subtle issue can be encountered with an offset between leakage components. This happens when the device spontaneously changes its operating voltage due to a

DVFS mechanism [SIH, YWV$^+$05], or equivalently with leakages located on the jitting slope of a carrier signal. Figure 2b depicts a mixture model with two perfect linear leakages ($\alpha_0 = \alpha_1 = 1$), without noise ($\sigma_0 = \sigma_1 = 0$) but with an offset between the two ($\beta_0 = \beta_1 + 25$). The offset being high enough, the two components are clusterable and there is no overlap between observations.

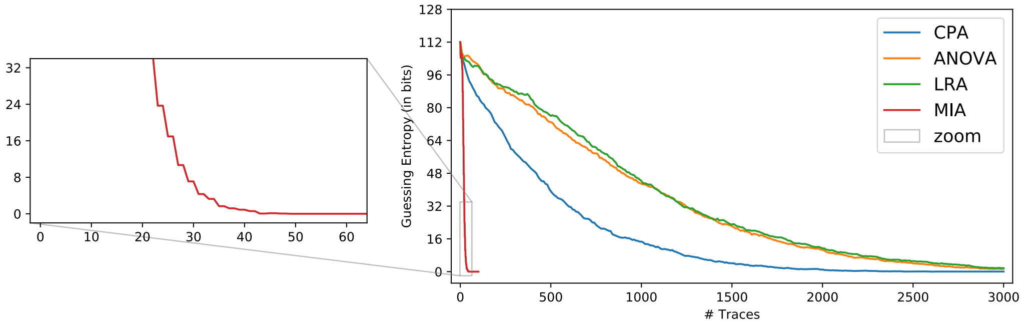


Figure 3: Comparison of distinguishers: average Guessing Entropy of 20 simulated attacks on a mixture of linear noiseless leakages with an offset.

In that case, the performance of the previously listed distinguishers also drops significantly. Figure 3 shows the average Guessing Entropy after 20 trial of simulated attacks based on correlation, F-test and linear regression, compared to Mutual Information. One can observe that our simulated mixture distribution induces a high bias in the statistics of the first three methods, which turns out to have a significant impact on the attack performance. In term of number of traces required, the CPA, ANOVA and LRA are two order of magnitude worsen, dropping from 10 traces when there is a single component to more than 3000 when analyzing this simple mixture. Indeed, these three methods are based on comparisons to the average as a reference, which is biased by the presence of multiple component in the mixture distribution. See Appendix A for an analysis of the limitations, detailed on CPA as an example.

On the other hand, MIA performance remains unchanged by the mixture effects. The main difference with Mutual Information is that it identifies the most remarkable set of conditional distributions, corresponding to the secret key, by independently analyzing each possible outcome of the probability space $\mathcal{X} \times \mathcal{Y}$. To evaluate dependence of observed traces with respect to the estimated leakage, it compares the conditional and marginal distributions together. As it does not use global properties of the distribution, Mutual Information is said a model agnostic distinguisher.

Yet, the MIA quickly becomes impractical with increasing noise, requiring a lot more traces. For instance, on the same simulation, adding a Gaussian noise, of only $\sigma = 3.5$, the MIA and CPA both require approximately 2200 traces to succeed, showing that MIA is less resilient to noise.[1]

As a result, on practical use-cases, classical distinguishers are all struggling to retrieve information from noisy mixture distributions. To better capture the leakage in such context, we first need to analyze in details what causes spreading of the leakage in the probability space. The main goal is to infer general properties of mixture distribution leakages that could be exploited by a distinguisher.

[1] At this noise level, the two mixture components remain non-overlapping, excluding bias due to complex mixing.

6 Understanding Leakage Spatialization

A first degree of leakage spatialization takes origins in the use of Boolean masking to split the secret into shares. In the following, we are going to have a look at second-order leakages to avoid missing this unavoidable point. For sake of clarity, this analysis starts with unidimensional leakages, and progressively move towards mixture leakages in Section 6.2

Assuming that two univariate shares of the secret intermediate value Y are observed without noise in X, that is $X_1 = X(t_1, \cdot)$ and $X_2 = X(t_2, \cdot) = X_1 \oplus Y$ respectively, the resulting theoretical conditional joint distributions show unique leakage patterns for each Hamming weight, as depicted in Figure 4. While in first-order the leakage takes form of a single point per Hamming weight in the probability space, Boolean masking spreads it over multiple values. Indeed, all possible ways to mask a given data of Hamming weight h lead to $(9-h) \cdot (1+h)$ different outcomes. This spatialization grows exponentially with the number of shares used for masking.

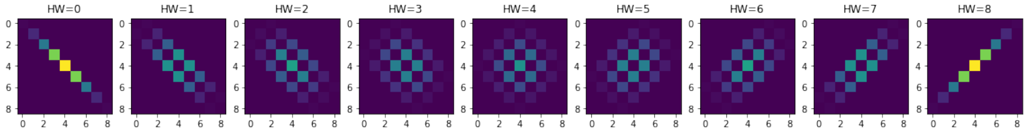


Figure 4: Theoretical conditional joint-probability distributions $P((X_1, X_2) = (x_1, x_2)|Y = h)$ under linear Hamming weight leakage model.

One can also notice that neighboring representations are totally disjoint from each other. This observation is no longer valid as soon as Gaussian noise is added to the shares. Figure 5 shows that instead of checkerboard, ellipsoidal shapes are emerging from the correct partitioning when the trace set is subject to Gaussian noise. More precisely, the theoretical conditional joint probability distributions are convoluted with a Gaussian kernel corresponding to the noise characteristics.

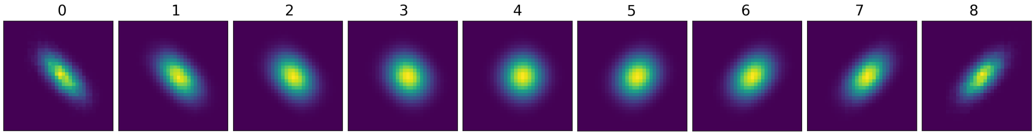


Figure 5: Conditional joint probability distributions under linear Hamming weight leakage model with Gaussian noise $\sigma = 1$.

The probability distributions for HW<4 and HW>4 are respectively leaning towards the left and right. It reveals a covariance and surely explains why techniques leveraging the centered product work.[2]

6.1 Differential Form Representation

A more interesting view can be obtained when using a differential form of usual distinguishers. The score of the attack targeting intermediate value Y for key candidate k, resulting from the application of distinguisher D on samples of $\boldsymbol{X}$ can be written:[3]

$$\text{Score}_D(\boldsymbol{X}; k) = \sum_{h \in \mathcal{Y}} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} \Delta_D(\boldsymbol{X} = \boldsymbol{x}, Y_k = h) \tag{10}$$

[2] The second-order CPA identifies a linear growth of the covariance from -1 to 1.

[3] for second-order, $\boldsymbol{X} = (X_1, X_2)$

where Δ_D is called the differential form of distinguisher D. It consists in the inner part of the sum leading to the final score, and is thus defined over the probability space $\boldsymbol{\mathcal{X}} \times \mathcal{Y}$.

In the following, while the approach can be extended to several distinguishers such as L_1 or χ^2, we will mainly use the differential form of Interaction Information [McG54, BGP+11] to represent the information contained locally in conditional joint-probability distributions. This distinguisher, extending the Mutual Information to more than 1 variable, turned out to be the most efficient in a lot of practical second-order scenarios we explored (a comparison can also be found in [GBPV10]). It is defined as follows:[4]

$$\begin{aligned}
-I(X_1;X_2;Y_k) &= I(X_1;X_2|Y_k) - I(X_1;X_2) \\
&= \sum_h P(Y_k = h) \cdot (I(X_1;X_2|Y_k = h) - I(X_1;X_2)) \qquad (11)\\
&= \sum_h \sum_{x_1} \sum_{x_2} \Delta_I(X_1 = x_1, X_2 = x_2, Y_k = h)
\end{aligned}$$

and its differential form is therefore:

$$\begin{aligned}
\Delta_I(X_1,X_2,Y_k) = P(Y_k) \cdot \Bigg[& P(X_1,X_2|Y_k) \log\left(\frac{P(X_1,X_2|Y_k)}{P(X_1|Y_k)P(X_2|Y_k)}\right) \\
& -P(X_1,X_2) \log\left(\frac{P(X_1,X_2)}{P(X_1)P(X_2)}\right) \Bigg] \qquad (12)
\end{aligned}$$

As shown in Equation (12), Interaction Information as a distinguisher is mainly based on the comparison of each conditional joint distribution to the average joint distribution over all Hamming weights, *i.e.* the marginal joint distribution. As this comparison is achieved by a difference, the differential form can take positive and negative values.

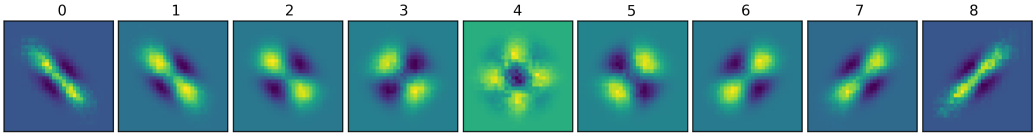


Figure 6: Conditional distributions of Δ_I computed from the joint probability distribution shown Figure 5.

In Figure 6, this representation reveals clear patterns, with areas in yellow (resp. dark blue) corresponding to outcomes more likely (resp. less likely) to be observed for a given Hamming weight. Notice the large similarities between consecutive Hamming weights. They are even more emphasized for a growing Gaussian noise, as it can be seen on the Figure 7. We can notice that the median Hamming weight does no longer seem to bring any valuable information, even though it is the most frequent observation. At this noise level, its leakage pattern is too close to the marginal average to be clearly identified. It is worth noting that a Gaussian of $\sigma = 3$ is still considered low noise in the context of side-channel measurements.

Most importantly, Δ_I mainly shows two clusters where the samples are more likely to be observed (yellow), as well as two clusters unlikely to be observed (blue). These clusters are similar for all Hamming weights strictly below the median 4, and symmetric strictly above the median HW value. However, the leaning effect is not obvious anymore. Indeed, the center of each conditional distribution brings very little information, *i.e.* it is hard to

[4] As the secret key minimize Interaction Information, a sign inversion is applied to reverse the candidates ranking.

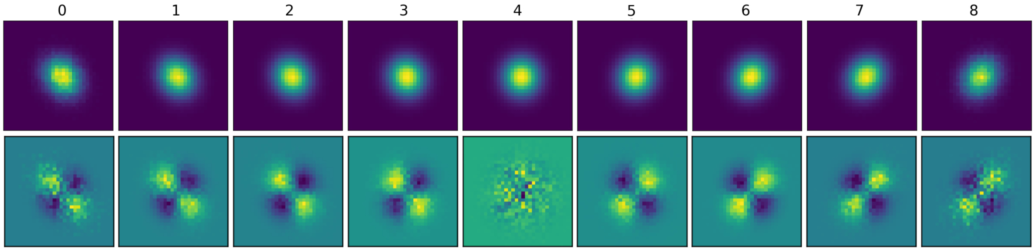


Figure 7: (top) Conditional joint probability distributions under linear HW leakage model with Gaussian noise $\sigma = 3$ (bottom) Conditional joint distributions of Δ_I.

decide if centers are yellow/blue in the differential form representation.

6.2 Mixture Spatialization

The previous results were obtained for a single pair of samples. Extending this to many leakage pairs with different characteristics should lead, for the right key guess, to a mixture composed of several patterns similar to those Figure 7, placed at different locations of the probability space. Indeed, mean value of each component determines the pattern location, and leakage amplitude its scale. Non-leakage components are not creating noticeable patterns but can add significant noise. The final pattern shape is intricate and hard to predict, still, a coherency should be observed for the right key guess, unlike for other guesses.

Besides, estimating probability distribution of a mixture with histograms requires additional care in the choice of bins. Modern data-based approaches were designed to choose the optimal number of bins for arbitrary distributions: Knuth's rule [Knu06] for equal-width bins and Scargle's Bayesian Blocks [SNJC13] for adaptive widths, which can help from a memory and performance standpoint, at the expense of implementation complexity. A simplified rule is to follow the pigeonhole principle by choosing bin edges ranging between minimum and maximum of measured samples, with a single bin per possible observation.

To illustrate spatialization of the leakage in probability mixtures, a reference trace set widely used by the side-channel community was considered: the so-called ASCAD dataset [PSB$^+$18]. It concerns an AES algorithm with second-order leakage of the SubBytes output. Interestingly, the public repository provides two data sets: one with traces being properly aligned (ASCAD.h5), and a second being misaligned (ASCAD_desync100.h5), with a uniformly random shift of less than 100 Samples to the left. With the knowledge of both shares, Figure 8 shows that leakages are present and significant.

Scatter was applied, following its extension to second-order [TVW19], on two frames of 100 points selected as depicted in Figure 8. They encompass approximately 4 clock cycles. Figure 9 depicts the mixture leakage in the probability space, obtained with the 50 000 traces provided. Each row displays the differential form of interaction information, Δ_I, for three key guesses and over the nine Hamming weights. Among them, `0xE0` is the correct key value, while `0x2B` and `0x80` were chosen randomly.

A zoom on Hamming weight 6 is given Figure 10 for the key and a wrong candidate. Looking at the bottom right corner, a clear pattern appears on the correct key guess, and this pattern is found in every image of the correct row in Figure 9. They are coherent over the different Hamming weight representations: it is highly similar for the first half

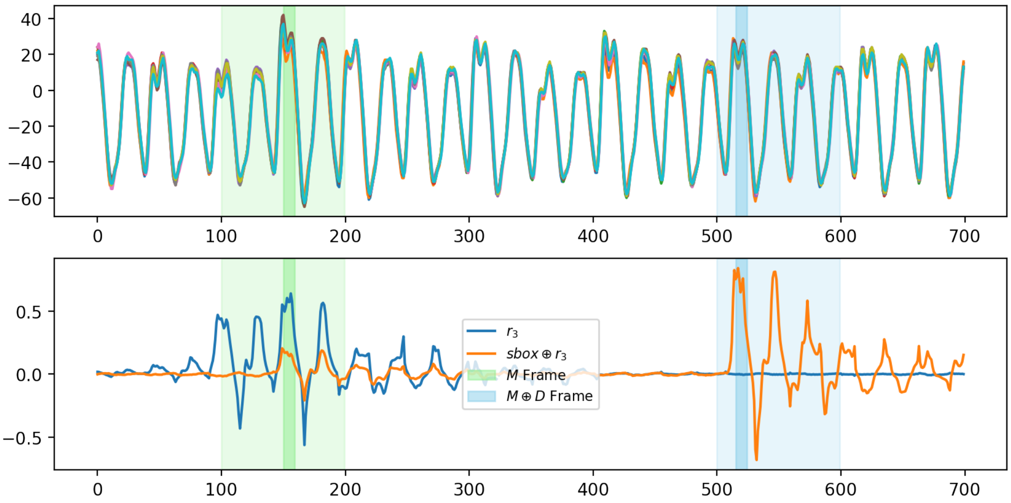


Figure 8: (top) Some traces from ASCAD aligned dataset. (Bottom) Correlation between traces, the mask X_1 and the masked SubBytes output X_2.

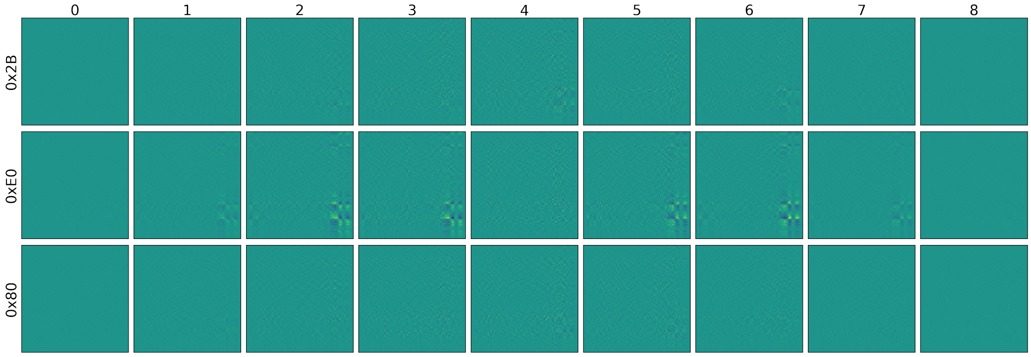
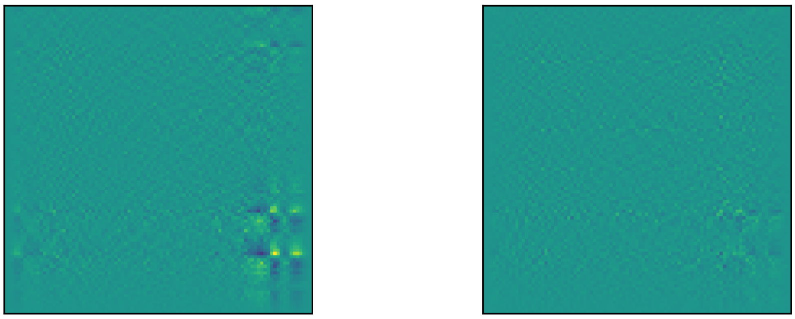


Figure 9: Conditional distributions of Δ_I for ASCAD traces after selection of two time windows of length $l_1 = l_2 = 100$ for X_1 and X_2, over 95 × 95 bins.



(a) Secret key `0xE0` (b) Wrong candiate `0x2B`

Figure 10: Zoom on Δ_I Hamming weight 6.

values and the second half values respectively. There is no clear pattern for the Hamming weight 4 either. For the wrong key guess, there is no similar observation, indicating these patterns constitute an exploitable leakage.

The pattern shape is complex due to the aggregation of many leakage samples of different characteristics in the probability mixture. Indeed, Figure 8 shows that there is a high number of leakages over multiple clock cycles. All have different means and variance, and also correlate distinctively. In fact, for both shares, samples leaking the most are located on top of signal peaks, thus their pairwise combinations creates an intricate leakage pattern in the bottom-right corner of the joint-distribution.[5]

Now the evidence is made that an exploitable information is present, as the secret key is found with the naked eye. Yet, efficiently extracting the information from the mixture distribution using a distinguisher remains a challenge.

At this stage, it is important to mention a drawback of Mutual Information (or Interaction Information) as a distinguisher. Indeed, these distinguishers, as described in the literature so far, implies a sum over the probability space as in Equations (11) and (12). When computing the score, both positive values (in yellow) and negative values (in blue) are summed together. It results that a significant amount of useful information is lost since the clear pattern is summarized into its too simplistic expression of a greater sum of positive values. The presence of negative information is also an indication of leakage, and could be exploited to find the key as well. In our opinion, this partially explains why MIA did not perform as expected, even if its fundamental benefits remain.

This led us to consider a more optimal way to exploit the available information. The problematic can be expressed as follows: when a coherency can be found across the conditional joint distributions, it is necessarily related to the secret key. Since the mixture leakage model is not predictable, the criterion must assume that there is a pattern, but its shape cannot be anticipated. This statement is particularly important when the selection of points incorporates many leakage samples. The next section introduces the notion of spatial analysis as a way to measure this coherency.

7 Spatial Dependency Analysis

Instead of scoring individually Δ_I over the entire probability space, spatial dependency analysis aims at considering the whole information held by a key guess, and find a way to extract any coherency in a subpart of the probability space.

As seen beforehand, the main properties identified from the analysis of spatialized leakages are the following:

1. Boolean masking creates spatialization of the leakage in the probability space.

2. Noise transforms the checkerboard patterns into imprecise clusters of information.

3. Existence of several non-informative and leaking components creates numerous leakage areas, placed at unknown locations of the probability space.

4. Areas of leakage are coherent for low and high Hamming weights, with a sign inversion with respect to the median.

5. Incoherent patterns can appear in wrong guesses distributions.[6]

6. Some distinguishers, such as Mutual Information, do not leverage negative differential to their advantage.

[5] The images origin is located in the top-left corner.

[6] Appendix B gives a finer example of such incoherence.

As depicted on Figure 11, we are only interested in finding a remarkable subpart of the probability space. To achieve this, we suggest to exploit a spatial auto-correlation measure, for instance by using Moran's Index, which can be used to efficiently extract information from side-channel mixtures.

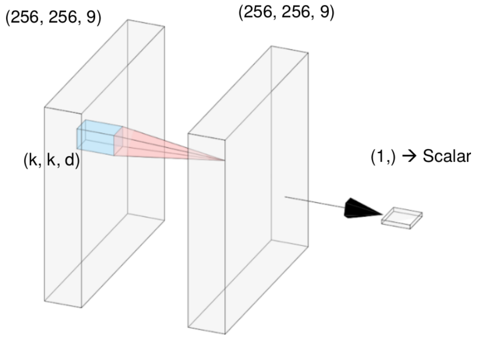


Figure 11: Definition of a vicinity kernel in the probability space, and visualization of local and global Moran's Index.

Moran's Index has been recently used to analyze the behaviour of Physically Unclonable Functions (PUFs) [VDTH16, WGP18]. This index was initially suggested [Mor50], and later refined in [Get95], to study the spatial correlation of measures such as the distribution of soil fertility over a field, or meteorological phenomena. The goal is to quantify if there are cold and hot spots in a map, *i.e.* several regions of lesser/greater intensity in the measurement, that are located close to each other. In this regard, Moran's formula gives a scalar index, that is:

- close to 1 in case of large and coherent hot/cold spots.
- close to 0 for random values.
- close to -1 for spatially incoherent measurements (interspersed).

It is defined as follows:

$$\mathrm{I_{Moran}}(x) = \frac{N}{W} \frac{\sum_i \sum_j w_{ij}(x_i - \overline{x})(x_j - \overline{x})}{\sum_i (x_i - \overline{x})^2} \tag{13}$$

with N the number of spatial sites of x, w_{ij} the expected interactions between sites i and j, and W the sum of all weights w_{ij}. It can also be conveniently expressed with standardized values, using μ and σ respectively the sample average and standard deviation of x:

$$\text{let } z = \frac{x-\mu}{\sigma}, \quad I_{\mathrm{Moran}}(x) = \frac{1}{W} \sum_i \sum_j w_{ij} \cdot z_i \cdot z_j$$

By definition, applying Moran's Index requires setting the interaction w_{ij} between each sites. In geographical analysis, this operation can be quite complex, as there are various kinds of spatial sites, with different areas, borders and content. But in general, the probability space studied during side-channel analyses, usually estimated via histograms, is regular (constant bandwidth), which greatly simplifies the process of defining the interaction between sites.

Following the numerous observations we did and the modeling of the spatial expression of the leakage in Section 6, but also in order to reduce the number of parameters to be defined, it appears that a vicinity of interaction, which is a rectangular cuboid of size (k_1, k_2, d), is sufficient to efficiently extract the leakage. Indeed, the area of spatial spreading of the information induced by masking and noise is counterbalanced by a kernel of size (k_1, k_2), while coherency between neighbor Hamming partitions can similarly be taken into account by extending the kernel with a third dimension by setting d. The simplest interaction model is to choose $w_{ij} = 1$ in the cuboid, except in its center where it should be 0. This cuboid is then slided through the differential form in order to analyze local dependencies and find hot/cold spots of interest. Doing so, Moran's Index can be computed as a stencil operator [RMKB97], which is easy to implement, and efficiently parallel:

$$\mathrm{I}_{\mathrm{Moran}}(\Delta_I) = \sum_{(i,j,k)} \sum_{\substack{0\leq\ i'<k_1 \\ 0\leq\ j'<k_2 \\ 0\leq\ k'<d}} \Delta_I^*(i,j,k)\cdot\Delta_I^*(i+i'-\frac{k_1-1}{2}, j+j'-\frac{k_2-1}{2}, k+k'-\frac{d-1}{2}) \tag{14}$$

where Δ_I^* is the standardized differential form of Interaction Information. Values outside the definition range are considered equal to zero.

In order to illustrate the discriminating power of Moran's Index applied to the differential form of Interaction Information, we drew Figure 12, using 4000 traces of ASCAD dataset and time-windows of length $l_1 = l_2 = 100$. On the left part, the leakage pattern is very subtle, especially when compared to Figure 10a with 50 000 traces. Yet, local Moran's Index obtained with $(k_1, k_2, d) = (7, 7, 5)$ strongly highlights a pattern in the interesting area of the probability space, while anywhere else the noise has been suppressed because of its incoherence. Moreover, notice the differential form is transformed to leverage negative information as a benefit, with mostly yellow regions after application of Moran's Index.

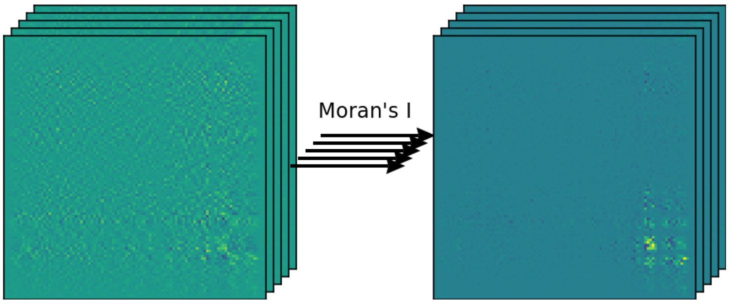


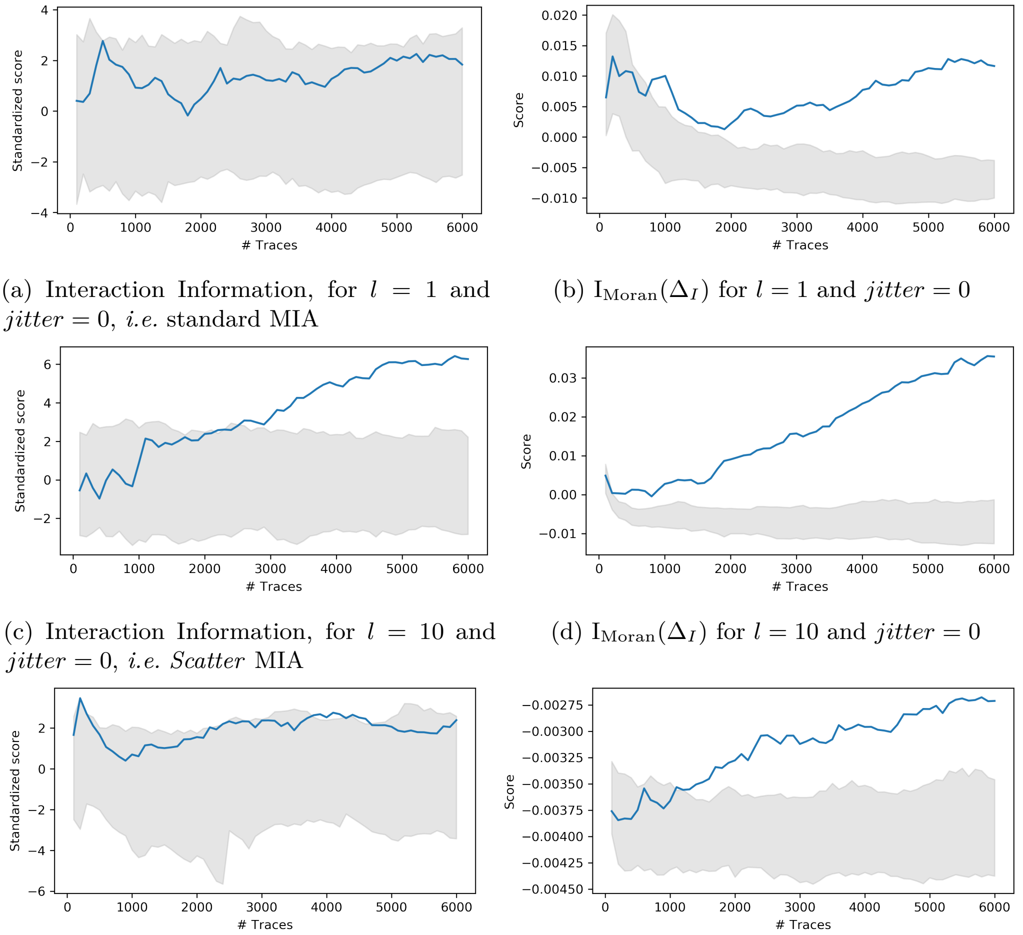
Figure 12: Local Moran's indexes after the processing of 4000 traces of the ASCAD dataset with $(k_1, k_2, d) = (7, 7, 5)$.

To sustain this result and the ability of the proposal to efficiently process complex mixtures, Figure 13 displays some comparative results for different time-window length $(l_1 = l_2)$ on ASCAD data set with and without jitter. To see when the key emerges with higher clarity, results without Moran's Index were standardized.

These results all confirm a strong benefit expressed both in terms of number of traces required to retrieve the secret and of distinguishability of the secret key compared to all other candidates. One can observe that only the correct key leads to a positive Moran's Index, meaning that a coherency is found, and that wrong guesses are perceived as incoherent, as they show a negative index. This confirms that incoherent leakage patterns can only be found in the distribution of wrong key guesses.

In the cases without jitter, the MIA is not able to recover the key given 6000 traces, while leveraging 10 leaking samples using *Scatter* allows to find it within 2500 traces. Moreover, the use of Moran's Index significantly reduces the number of traces required, to approximately 600 with a single leakage sample, and 200 with a mixture of 10.

Finally, with a jitter of 100 time-samples, the use of *Scatter* in conjunction with Moran's Index allows to recover the key in 1000 traces.



(a) Interaction Information, for $l = 1$ and $jitter = 0$, *i.e.* standard MIA

(b) $\mathrm{I}_{\mathrm{Moran}}(\Delta_I)$ for $l = 1$ and $jitter = 0$

(c) Interaction Information, for $l = 10$ and $jitter = 0$, *i.e. Scatter* MIA

(d) $\mathrm{I}_{\mathrm{Moran}}(\Delta_I)$ for $l = 10$ and $jitter = 0$

(e) Interaction Information, for $l = 200$ and $jitter = 100$, *i.e. Scatter* MIA

(f) $\mathrm{I}_{\mathrm{Moran}}(\Delta I)$ for $l = 200$ and $jitter = 100$

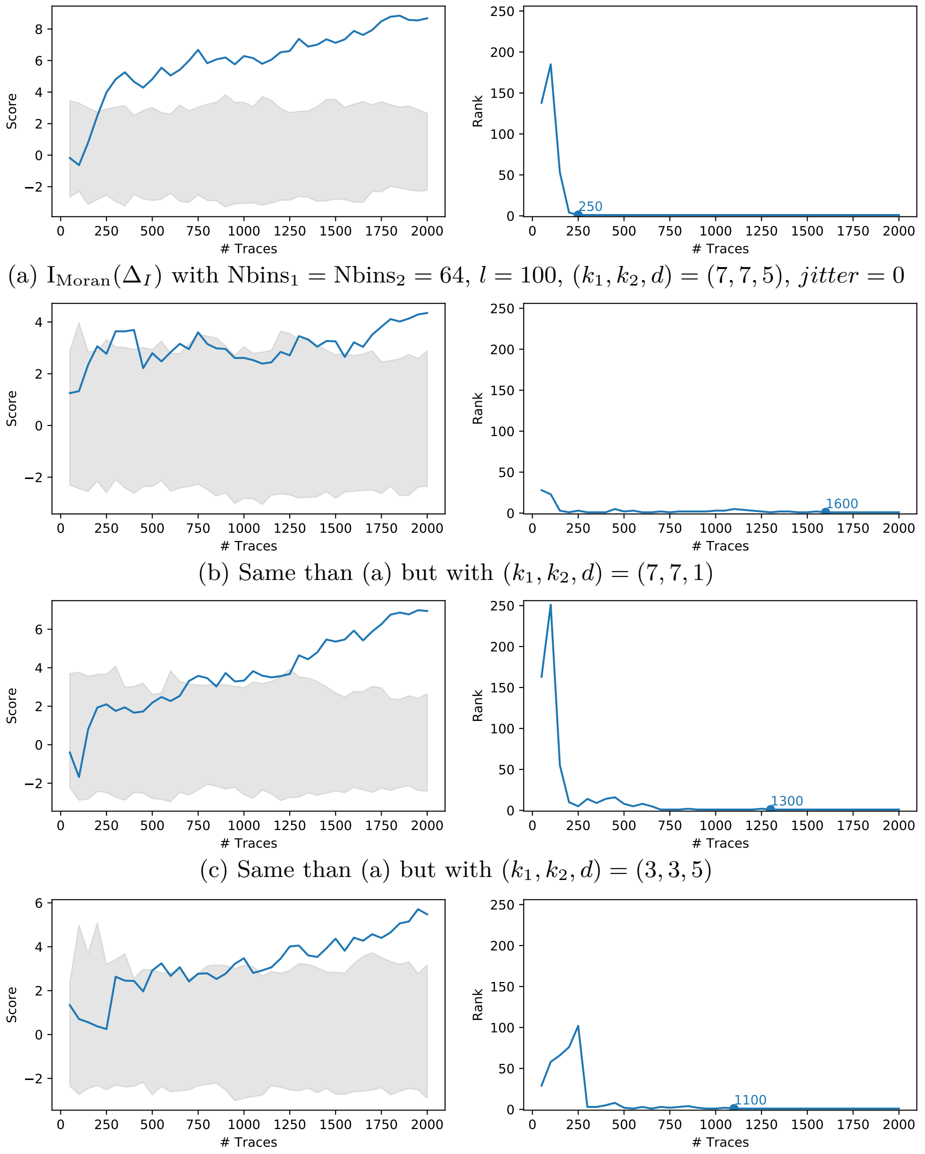
Figure 13: Comparison of attacks on ASCAD using Interaction Information distinguisher, with and without *Scatter* (left column) and Moran's Index (right column). Targeting the strongest point of leakage on aligned traces (1st row), 10 points around it on aligned traces (2nd row), and a window of 200 points when facing 100 points of jitter (3rd row). All results are the best of 10 tries with random traces order (kept identical between attacks).

For all these experiments, we looked for an optimum set of hyper parameters and more particularly an optimal kernel size (k_1, k_2, d); the computation effort remaining reasonable. We observed during this experimentation that the impact of kernel dimensions remains fairly limited: the number of traces required to retrieve the key varies only from 250 to 1600. Some results are displayed on Figure 14 showing the influence of the hyper-parameters choice on the distinguishability and the number of traces required to get the key. Notice that after 250 traces, all configurations rank the key among the 10 firsts, but setting the right parameters brings a significant margin of distinguishability to the correct key.

There are some rationales in the choice of these parameters:

- Figure 14b depicts the same attack than Figure 14a but with $d = 1$. This change implies completely ignoring coherency between conditional distributions of neighbor Hamming weights. It appears that the depth parameters d, plays an important role in the performance of the technique. Without this aspect, even if the key rank is low, the resolving power is significantly lessen.

- Figure 14c and Figure 14d give an appreciation of the influence of parameter $k = k_1 = k_2$. This should represent the feature size of leakages. A $k = 3$, chosen minimal, will not counterbalance noise spreading properly, and in the opposite, if chosen too large, it tends to incorporate positive and negative differential information in the same vicinity, annihilating information. In this example, the sweet-spot seems around 7×7 for 64×64 bins histograms. Note these values should be chosen relatively to the number of bins used to estimate the distributions of X_1 and X_2.



(a) $\mathrm{I}_{\mathrm{Moran}}(\Delta_I)$ with $\mathrm{Nbins}_1 = \mathrm{Nbins}_2 = 64$, $l = 100$, $(k_1, k_2, d) = (7, 7, 5)$, $jitter = 0$

(b) Same than (a) but with $(k_1, k_2, d) = (7, 7, 1)$

(c) Same than (a) but with $(k_1, k_2, d) = (3, 3, 5)$

(d) Same than (a) but with $(k_1, k_2, d) = (17, 17, 5)$

Figure 14: Effect of kernel choice hyper-parameter on $\mathrm{I}_{\mathrm{Moran}}(\Delta_I)$ distinguisher. Scores are standardized for comparison purposes.

8 Application to a Masked and Shuffled AES

To further sustain the technique, it was applied on a second practical use-case. The aim was to clearly exhibit a use case where the new technique has a strong interest. It concerned a software implementation of an AES-128 encryption. The code implemented a classical Boolean masking algorithm. We checked that there was no remaining first-order leakage. An additional protection was added and could be enabled: the random shuffling of SubByte operations.

The corresponding binary was loaded on a microcontroller based on Cortex-M4. More specifically, we used a STM32F446 32-bit MCU, with a clock setup at 30 MHz. This frequency was chosen as the highest leading to negligible clock jitter in the measurement. A near field EM acquisition was made using a Langer probe RF-B 0.3-3. The DUT package was not removed.

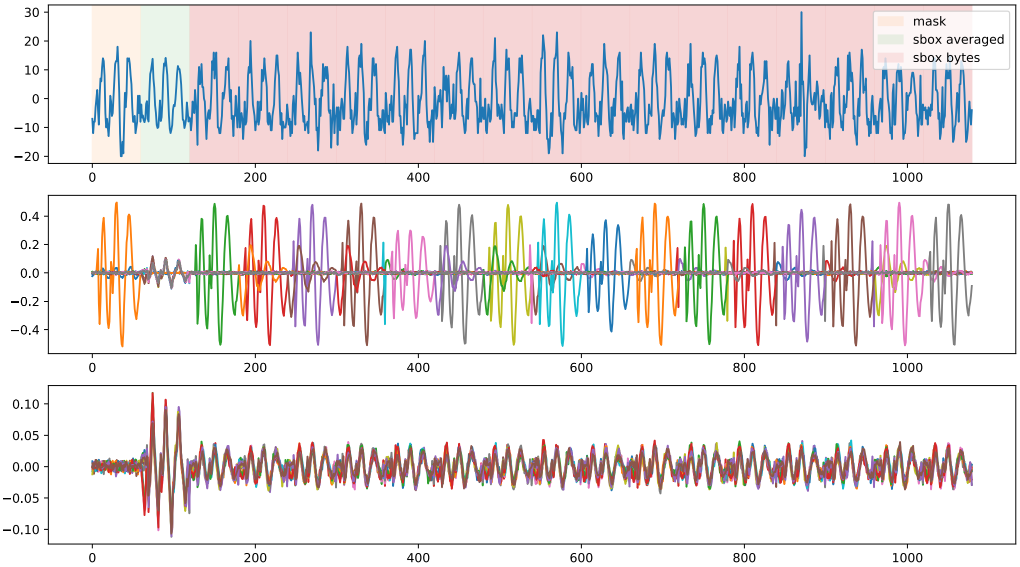


Figure 15: Overview of SubBytes leakages. (top) Trace portions with areas highlighted. (middle) Correlation between traces and intermediate values of mask and masked SubBytes output, obtained with knowledge of random mask values, and shuffling disabled. (bottom) Correlation of masked SubBytes with shuffling enabled.

With knowledge of the random numbers used for masking, and without shuffling enabled, it was possible to identify all shares location. We then built a traceset, visible in Figure 15, with the concatenation only of the mask area (orange), the average of 16 SubBytes operations (light green), followed by the 16 non-averaged SubBytes operations (light red).

When facing a shuffling protection, the attacker may have no choice than using techniques such as averaging all 16 occurrences together and apply the analysis on this average signal. Performing this operation transforms the mixture of 16 leakages components into a single averaged component. Note that SubBytes pattern averaging was performed using the leakage patterns, which required knowledge of the key and the mask to characterize properly the signal. Without this knowledge, SubBytes pattern averaging would have been tedious or arguably impossible.

Figure 15 showing correlation values with known mask, confirms that enabling the shuffling reduces the first-order leakage by a factor 16. However, when averaging the SubBytes patterns together, some leakage remains although it is significantly lowered. Only the shuffled traces will be considered in the following of this section.

After this preliminary analysis of the trace characteristics, we analyzed how the leakage expresses in the conditional distributions, using the differential form of Interaction Information. In Figure 16, a pattern is clearly visible for the right key guess `0x07` and it is coherent over the different Hamming weight values. Looking at other guesses, no discriminating characteristic appears, confirming without any doubt which key guess is the correct one. Notice that guess `0x15`, on the top row, shows false positive patterns with incoherence between HW 4 to 6.[6]

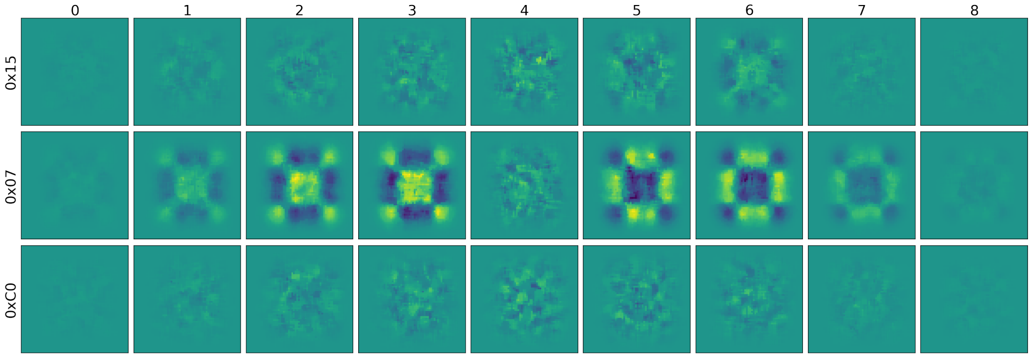
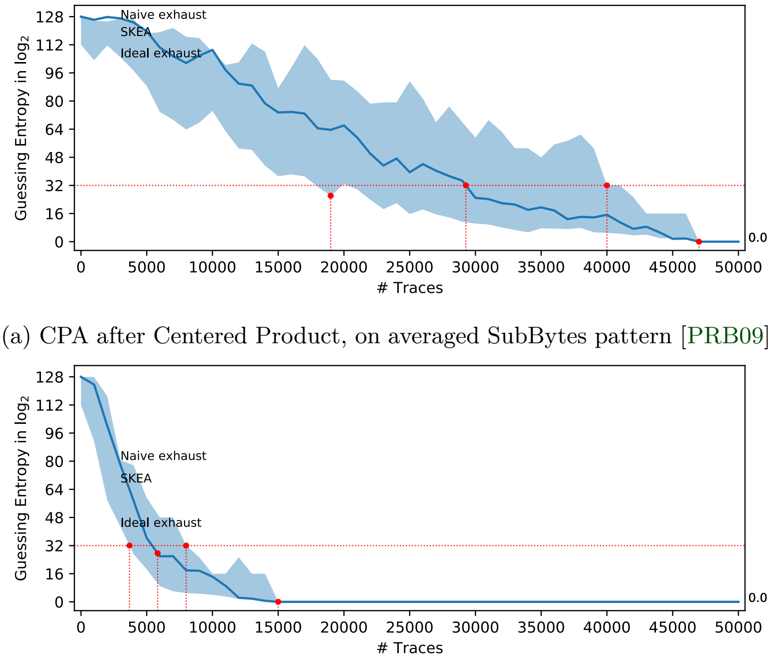


Figure 16: Conditional distributions of Δ_I for 3 keys guesses (`0x07` is the correct key byte).

The best results when applying classical attacks were obtained, as shown Figure 17a, with a univariate CPA on the averaged SubBytes signal, after CenteredProduct recombination. This attack was able to recover the secret key with 47 000 traces. In our testing, no time integration technique [CCD00, WW04, BBB+16] was able to improve this result.

On the other hand, with a *Scatter* approach, the process is much simpler: the analyst can select the whole signal area containing the 16 SubBytes computation and perform a single attack on the resulting complex mixture. Attacking the averaged leakage pattern makes less sense, but results are also given for comparison purpose. The performance of *Scatter* with Interaction Information distinguisher is already a bit better than the best CPA, finding the key with 42 000 traces. Yet, the use of spatial dependency analysis, and in particular Moran's Index applied to the differential form of the Interaction Information, further enhances the results. The Guessing Entropy, as shown on Figure 17b, drops below a bruteforceable threshold within 5000 traces, and the key is recovered with only 15 000.

Table 1 summarizes all results. It gives the Guessing Entropy and the related number of traces for each attack variant. The first column reports results obtained when considering the area enclosing all SubBytes. The second one gives results obtained when averaging all SubBytes. This clearly demonstrates the soundness of the scatter approach, as well as Moran's Index described in the present paper, and their efficiency to extract the leakage from complex mixtures of signals. On the last row of the table, averaging the SubBytes seems to reduce the leakage exploited by both *Scatter* attacks.



(a) CPA after Centered Product, on averaged SubBytes pattern [PRB09]

(b) $\mathrm{I}_{\mathrm{Moran}}(\Delta_I)$ on the whole SubBytes area (this paper)

Figure 17: Guessing Entropy of the best attacks against masked and shuffled AES implementation.

Table 1: Comparative summary of Guessing Entropies after application of different attack techniques targeting a STM32F4 running a masked and shuffled AES.

	No Averaging	**Averaging**
Centered Product CPA [PRB09]	$> 2^{90}$ at 50 000	**2^{32} at 18 000** **1 at 47 000**
Centered Product CPA + Integration [CCD00]	$> 2^{90}$ at 50 000	2^{32} at 34 000 2^{20} at 50 000
Cross-correlation (x-corr) [WW04, BBB+16]	$> 2^{100}$ at 50 000	2^{43} at 50 000
concatFFT [BBB+16]	$> 2^{100}$ at 50 000	2^{64} at 50 000
concatFHT [BBB+16]	$> 2^{100}$ at 50 000	2^{70} at 50 000
windowFFT [BBB+16]	$> 2^{100}$ at 50 000	$> 2^{100}$ at 50 000
windowFHT [BBB+16]	$> 2^{100}$ at 50 000	2^{60} at 50 000

Scatter with Interaction Information [TVW19]	2^{32} at 14 000 1 at 42 000	2^{32} at 28 000 $< 2^{10}$ at 50 000
Scatter and Moran's Index on Δ_I (this paper)	**2^{32} at 4000** **1 at 15 000**	2^{32} at 6000 1 at 18 000

9 Conclusion

This work attempted to highlight the benefits of using probability mixtures to model the different issues encountered when performing practical side-channel analyses. This model becomes unavoidable when dealing with datasets from complex devices or countermeasures such as shuffling. The aim is to provide new tools to understand and characterize the signal captured on these devices, especially when the initial state of the experiment is not properly controlled. Doing so, the benefit of incorporating many points of selection and working with mixture distributions appears more clearly. We then provided a better formalization of *Scatter* technique by expressing it in the context of mixture distributions.

Subsequently, this paper develops a spatialized leakage model, by exploring useful information present in the differential form of the Interaction Information. Indeed, a coherent information is only associated to the correct key guess. This led us to demonstrate why techniques, such as MIA, could be greatly improved.

An improvement is elaborated by introducing spatial dependency analysis to capture remarkable information related to the correct key guess. By leveraging Moran's index as a way to measure the spatial auto-correlation, the practical results highlighted that the technique can be very beneficial for running complex side-channel attacks with a significant factor of improvement, both in number of traces and distinguishability. We believe that the new technique is powerful enough to be part of the toolbox for anyone taking into consideration the non-profiled side-channel threat.

Finally, we strongly believe that the proposed spatial dependency analysis could be further enhanced, possibly by using tools developed in other scientific areas dealing with spatialized data and/or mixture distributions.

References

[AKD+18] Monjur Alam, Haider Adnan Khan, Moumita Dey, Nishith Sinha, Robert Callan, Alenka Zajic, and Milos Prvulovic. One&done: A single-decryption em-based attack on openssl's constant-time blinded RSA. In 27th USENIX Security Symposium (USENIX Security 18), pages 585–602, Baltimore, MD, August 2018. USENIX Association.

[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.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings, volume 3156 of Lecture Notes in Computer Science, pages 16–29. Springer, 2004.

[BDGN13] Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, and Zakaria Najm. NICV: normalized inter-class variance for detection of side-channel leakage. IACR Cryptol. ePrint Arch., 2013:717, 2013.

[BGP+11] Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual information analysis: a comprehensive study. J. Cryptol., 24(2):269–291, 2011.

[CCD00] Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential power analysis in the presence of hardware countermeasures. 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 252–263. Springer, 2000.

[DPRS11] Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. Univariate side channel attacks and leakage modeling. J. Cryptogr. Eng., 1(2):123–144, 2011.

[GBPV10] Benedikt Gierlichs, Lejla Batina, Bart Preneel, and Ingrid Verbauwhede. Revisiting higher-order DPA attacks:. In Josef Pieprzyk, editor, Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings, volume 5985 of Lecture Notes in Computer Science, pages 221–234. Springer, 2010.

[GDTL20] Joseph Gravellier, Jean-Max Dutertre, Yannick Teglia, and Philippe Loubet-Moundi. Sideline: How delay-lines (may) leak secrets from your soc. CoRR, abs/2009.07773, 2020.

[Get95] Arthur Getis. Cliff, ad and ord, jk 1973: Spatial autocorrelation. london: Pion. Progress in Human Geography, 19(2):245–249, 1995.

[GPP$^+$16] Daniel Genkin, Lev Pachmanov, Itamar Pipman, Eran Tromer, and Yuval Yarom. ECDSA key extraction from mobile devices via nonintrusive physical side channels. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016, pages 1626–1638. ACM, 2016.

[GPT15] Daniel Genkin, Itamar Pipman, and Eran Tromer. Get your hands off my laptop: physical side-channel key-extraction attacks on pcs - extended version. J. Cryptogr. Eng., 5(2):95–112, 2015.

[Knu06] Kevin H Knuth. Optimal data-based binning for histograms. arXiv preprint physics/0605197, 2006.

[LMPT15] Jake Longo, Elke De Mulder, Dan Page, and Michael Tunstall. Soc it to EM: electromagnetic side-channel attacks on a complex system-on-chip. In Tim Güneysu and Helena Handschuh, editors, Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings, volume 9293 of Lecture Notes in Computer Science, pages 620–640. Springer, 2015.

[McG54] William J. McGill. Multivariate information transmission. Trans. of the IRE Professional Group on Information Theory (TIT), 4:93–111, 1954.

[Mor50] Patrick AP Moran. Notes on continuous stochastic phenomena. Biometrika, 37(1/2):17–23, 1950.

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

[PSB$^+$18] Emmanuel Prouff, Rémi Strullu, Ryad Benadjila, Eleonora Cagli, and Cécile Dumas. Study of deep learning techniques for side-channel analysis and introduction to ASCAD database. IACR Cryptol. ePrint Arch., 2018:53, 2018.

[RMKB97] Gerald Roth, John M. Mellor-Crummey, Ken Kennedy, and R. Gregg Brickner. Compiling stencils in high performance fortran. In Proceedings of the ACM/IEEE Conference on Supercomputing, SC 1997, November 15-21, 1997, San Jose, CA, USA, page 12, 1997.

[RSWO18] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. Iot goes nuclear: Creating a zigbee chain reaction. IEEE Secur. Priv., 16(1):54–62, 2018.

[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 Pil Joong Lee and Jung Hee Cheon, editors, Information Security and Cryptology - ICISC 2008, 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers, volume 5461 of Lecture Notes in Computer Science, pages 253–267. Springer, 2008.

[SIH] Diary R. Suleiman, Muhammed A. Ibrahim, and Ibrahim I. Hamarash. Dynamic voltage frequency scaling (dvfs) for microprocessors power and energy reduction.

[SNJC13] Jeffrey D Scargle, Jay P Norris, Brad Jackson, and James Chiang. Studies in astronomical time series analysis. vi. bayesian block representations. The Astrophysical Journal, 764(2):167, 2013.

[TGWC18] Hugues Thiebeauld, Georges Gagnerot, Antoine Wurcker, and Christophe Clavier. SCATTER: A new dimension in side-channel. In Junfeng Fan and Benedikt Gierlichs, editors, COSADE 2018, volume 10815 of LNCS, pages 135–152. Springer, Heidelberg, April 2018.

[TVW19] Hugues Thiebeauld, Aurélien Vasselle, and Antoine Wurcker. Second-order scatter attack. IACR Cryptol. ePrint Arch., 2019:345, 2019.

[VDTH16] Shrikant Vyas, Naveen Kumar Dumpala, Russell Tessier, and Daniel E. Holcomb. Improving the efficiency of puf-based key generation in fpgas using variation-aware placement. In Paolo Ienne, Walid A. Najjar, Jason Helge Anderson, Philip Brisk, and Walter Stechele, editors, 26th International Conference on Field Programmable Logic and Applications, FPL 2016, Lausanne, Switzerland, August 29 - September 2, 2016, pages 1–4. IEEE, 2016.

[VMC19] Aurélien Vasselle, Philippe Maurine, and Maxime Cozzi. Breaking mobile firmware encryption through near-field side-channel analysis. In Chip-Hong Chang, Ulrich Rührmair, Daniel E. Holcomb, and Patrick Schaumont, editors, Proceedings of the 3rd ACM Workshop on Attacks and Solutions in Hardware Security Workshop, ASHES@CCS 2019, London, UK, November 15, 2019, pages 23–32. ACM, 2019.

[vWWB11] Jasper G. J. van Woudenberg, Marc F. Witteman, and Bram Bakker. Improving differential power analysis by elastic alignment. In Aggelos Kiayias, editor, Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings, volume 6558 of Lecture Notes in Computer Science, pages 104–119. Springer, 2011.

[WGP18] Florian Wilde, Berndt M. Gammel, and Michael Pehl. Spatial correlation analysis on physical unclonable functions. IEEE Transactions on Information Forensics and Security, 13(6):1468–1480, Jun 2018.

[WW04] Jason Waddle and David A. Wagner. Towards efficient second-order power analysis. In Marc Joye and Jean-Jacques Quisquater, editors, Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings, volume 3156 of Lecture Notes in Computer Science, pages 1–15. Springer, 2004.

[YWV+05] Shengqi Yang, Wayne H. Wolf, Narayanan Vijaykrishnan, Dimitrios N. Serpanos, and Yuan Xie. Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach. In 2005 Design, Automation and Test in Europe Conference and Exposition (DATE 2005), 7-11 March 2005, Munich, Germany, pages 64–69. IEEE Computer Society, 2005.

A Details About CPA Limitations on a Simple Mixture

This annex details why a CPA attack requires much more traces to succeed when facing a simple mixture distribution. Let X be a mixture of two positive and linear Hamming weight leakages without noise:

$$P(X) = \lambda \cdot P(T_0) + (1-\lambda) \cdot P(T_1) \quad \text{with} \quad T_0 = T_1 + \beta = \mathrm{HW}(L) \tag{15}$$

Using an intermediate value estimate Y_k for a given key guess k, the CPA ranks keys according to Pearson's correlation:

$$\forall k, \quad \rho(X, Y_k) = \frac{\mathrm{cov}(X, Y_k)}{\sigma_X \sigma_{Y_k}} \tag{16}$$

First, the normalization term at denominator is identical for all key guesses, and does not affect the key ranking. Looking at the numerator, it follows for this mixture distribution that:

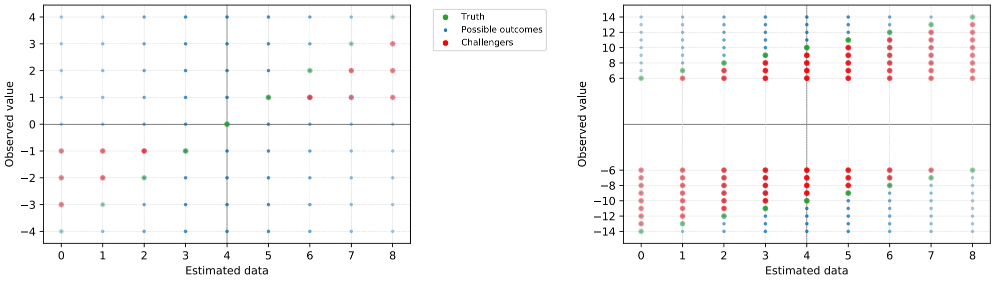
$$\begin{aligned}
\forall k, \quad \mathrm{cov}(X, Y_k) &= \mathrm{E}[XY_k] - \mathrm{E}[X]\,\mathrm{E}[Y_k] && (17)\\
&= \lambda\ \mathrm{E}[T_0 Y_k] + (1-\lambda)\ \mathrm{E}[T_1 Y_k] - \lambda\ \mathrm{E}[T_0]\,\mathrm{E}[Y_k] - (1-\lambda)\ \mathrm{E}[T_1]\,\mathrm{E}[Y_k] \\
&= \lambda\ \mathrm{cov}(T_0, Y_k) + (1-\lambda)\ \mathrm{cov}(T_1, Y_k) && (18)\\
&= \mathrm{cov}(\mathrm{HW}(L), Y_k) && (19)
\end{aligned}$$

Regardless of the key, the covariance computed with the mixture is thus equal to that with a simple leakage component.

To isolate each effect, the attack was attempted with a correlation using the theoretical population expectations and variances $E[X], E[Y_k], \sigma_X, \sigma_{Y_k}$ and the results are similar, requiring more than 3000 traces for a noiseless leakage. As a result, the difference of behavior must originate from larger errors in the estimate of sample covariance:

$$\forall k, \quad \widetilde{\mathrm{cov}}(X, Y_k) = \sum_x \sum_y (x - \mathrm{E}[X])(y - \mathrm{E}[Y_k]) \cdot \widetilde{P}(X = x, Y_k = y) \tag{20}$$

For a given amount of traces, the contribution a single observable value (x, y) brings to the covariance can sometimes be greater for an incorrect guess. In that case, the score of this guess challenges that of the secret key, which means more samples (traces) are required to differentiate the correct key. It turns out that the probability of being challenged is much higher in the mixture scenario, as depicted in Figure 18. The green dots are points of leakage obtained for the secret key, blue dots represent the outcomes obtained with incorrect key guesses, and red dots are all cases where the secret key is challenged for a given observation.



(a) Single leakage component

(b) Mixture of two offset components

Figure 18: Representation of the probability that an observation contributes more to the sample covariance for an incorrect guess than for the key.

To give an order of magnitude of the difference, we assume independence of the incorrectly estimated data and observation.[7] For a single leakage component, the probability of an observation being challenged by another key is 4.0%, while for our mixture scenario, it is 44.4%. As a result, with a given number of traces, the CPA in the mixture scenario is much less likely to succeed. Or in other words, retrieving the secret key from this mixture requires significantly more traces, which concurs with the simulation Figure 3.

B Evidences of Incoherent Patterns in Wrong Key Guesses Distributions

Even when targeting a highly non-linear intermediate data (SubBytes output) and processing more traces than necessary to find the secret key, incoherent leakage patterns are still found in the conditional joint-distribution of wrong key guesses. Figure 19 shows the conditional distribution of Δ_I for 3 wrong key guesses. The residual patterns are less potent than for the correct key, therefore the colormap was adjusted to highlight smaller variations.

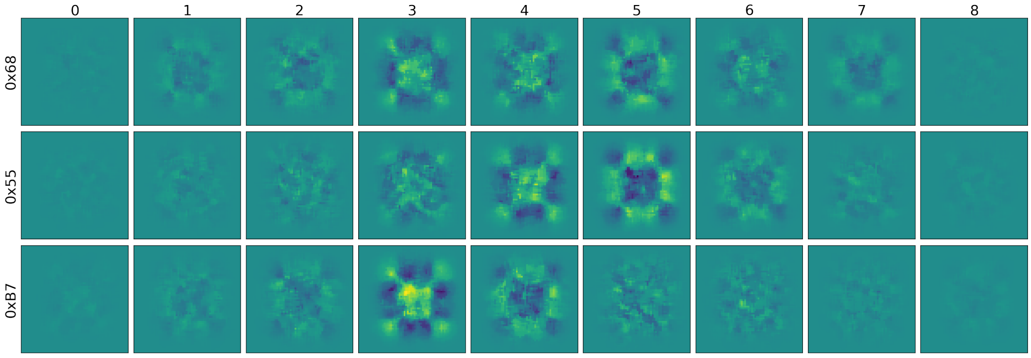


Figure 19: Example of incoherent patterns found in Δ_I for wrong key guesses on the STM32 AES implementation.

The patterns look exactly like the ones found in Figure 16. However, they are incoherent with previous observations. First, by construction, the HW 4 should not exhibit the strong leakages. Second, there are adjacent partitions showing sign opposition instead of a coherency between low and high HW.

[7] This hypothesis is false, even after non-linear transformations such as AES SBox, but allows to easily compute an approximate result.