SCATTER : A New Dimension in Side-Channel

Hugues Thiebeauld[1], Georges Gagnerot[1], Antoine Wurcker[1], and Christophe Clavier[2]

[1] eshard,

Martillac, France

[2] Université de Limoges, XLIM-CNRS

Limoges, France

hugues.thiebeauld@eshard.com
georges.gagnerot@eshard.com
antoine.wurcker@eshard.com
christophe.clavier@unilim.fr

Abstract. Side-channel techniques have well progressed over the last few years, leading to the creation of a variety of statistical tools for extracting the secrets used in cryptographic algorithms. Such techniques are taking advantage of the side-channel traces collected during the executions of secret computations in the product. Noticeably, the vast majority of side-channel attacks requires the traces have been aligned together beforehand. This prerequisite turns out to be more and more challenging in the practical realisation of attacks as many devices include hardware or software countermeasures to increase this difficulty. This is typically achieved applying random jittering or random executions with fake operations. In this paper, we introduce *scatter*, a new attack which has the potential to tackle most of the alignment issues. scatter brings a new dimension to improve the efficiency of existing attacks and opens the door to a large set of potential new attack techniques. The effectiveness of scatter has been proven on both simulated traces and real word secure products. As a result, scatter is a new side-channel technique particularly powerful when the trace alignment represents an issue, or even when considering low-cost attacks, as the requirements in terms of equipment and expertise are significantly reduced.

Keywords: side-channel, scatter, mutual information, Pearson chi-squared

1 Introduction

Over the past few years, *Side-Channel Attacks* have been proven to be effective in attacking a wide range of hardware devices [17–19]. Recently their application to compromise obfuscated ciphers including whitebox cryptographic libraries has been proven. Therefore, software based products can be subject to these attacks [5]. When successful, the impact of side-channel attacks is severe as it leads to the disclosure of the secret cryptographic key. In case of partial recovery, several techniques [34] can be used to achieve the whole key recovery. As a result, a flaw can be exploited and leads to losses. On the other hand, the perception of the product security can be dramatically undermined. In order

to avoid any exposure on the field, right protections must be implemented and validated by practical testing.

There are a number of side-channel attack techniques to take into consideration. Beginning with the original *Simple Power Analysis* (SPA) [23] and *Differential Power Analysis* (DPA) [24], several other techniques have been developed over the past few years. The most famous distinguishers are CPA [6], standing for *Correlation Power Analysis* and exploiting Pearson coefficient, the MIA [20], standing for *Mutual Information Analysis* and exploiting the Shannon entropy and the LRA for *Linear Regression Analysis* [14, 37]. These attacks can be run on a device without any preliminary knowledge about their implementation. Some other attacks (such as the *Templates Attacks* also named *Profiled* Attacks) [7] make use of a profiling stage used to target other similar devices through a matching phase. Finally, another testing methodology with the *T-Tests* have been introduced [4, 15, 21, 22] to characterise potential leakages in an efficient manner.

In the same period several countermeasures have been regularly published and improved in order to defeat these attack techniques. Most common ones consist in misalign different executions with hardware and software techniques [9, 38, 11] and/or to de-correlate the information from the traces using random values for masking the data manipulated. In the masking case, more complex but realistic, attacks named *Higher-Order Side-Channel Analysis* introduced by Messerges [29], studied [35] and later improved [3, 27, 30, 33, 39] is still applicable when some mask values are identical at some points of interest for the attack in the traces. Generally speaking, traces alignement remains a main technical issue when performing first and higher order side-channel attacks in practice.

We present in this paper a new side-channel analysis technique named *scatter* which has the potential to tackle most of the alignement issues including random jittering and random order execution. We consider *scatter* opens doors to improvements in side-channel analyses including higher order side-channel attacks and requires the development of innovative countermeasures to strengthen the security of existing products. This new technique has been proven efficient and is presented with practical experiments.

This paper is organised as follows. Section 2 gives necessary background on side-channel analysis. We present in Section 3 the principles of our scatter method. Section 4 presents a first validation of scatter efficiency based on simulations when practical results on physical measurements from a real hardware device are given in Section 5. Section 6 shows a comparison with some other window-based techniques, such as Fast Fourier Transform (FFT) or average. Section 7 introduces how the technique can be extended to higher order attacks. We discuss the impact of this new technique on secure products and state-of-the art countermeasures in Section 8, and conclude in Section 9.

2 Side-Channel Analysis Practical Issues

Quite remarkably, except for simple side-channel attacks using one single trace, all attack techniques mostly rely on the assumption that the measurement or data traces have been aligned as much as possible before applying a statistical analysis tool. In other words, it requires the estimated variable to be located at the same (X-axis) index along a certain number of traces - being the minimum number of aligned traces required to exploit the leakage with the statistical test. Without this prerequisite condition, none of the attack listed previously leads to a success. And this condition becomes more and more an issue when the cryptographic algorithm is executed on recent secure devices and on complex devices such as a SoC (System on Chip) in a mobile platform, which includes more complex mechanisms such as multi-threading.

When conducted practically, the alignment step may be time consuming and difficult. This requires a specific expertise and may increase dramatically the number of traces required to expose the key when not properly done. Moreover, the alignment represents a significant part of the effort for performing an attack. Conscious of this, some interesting, but limited, work has been developed to investigate ways to automate the alignment. We can quote the elastic alignment [40] exploiting fast Dynamic Time Warping. Some other techniques explored the use of wavelets [13, 26, 32]. All these techniques represent good tools, but they remain hard to apply in a generic way and sometimes they turn out to be inefficient.

On the other hand, some other studies investigated the opportunity to work in the frequency domain via Fourier transformations. Interesting results were obtained on second order attacks in [3]. Indeed, Fourier transformations represent a valuable tool for tackling a misalignment as it integrates a piece of trace in time domain into its equivalent in frequency domain. All values gathered within the window, even non-aligned, are spread over their corresponding set of frequencies. Practically, this technique shows interesting results, but it remains inefficient in a lot of practical cases. The downside of this technique is mostly related to the nature of Fourier transformations. Indeed, the information is spread over the corresponding sets of frequency and can face a lack of alignment in case of variable clocks. A second issue concerns the relationship between the window size and the information sampling in the frequency domain. Last but not least, the notion of frequency is not all the time applicable, such as for studies of whitebox cryptography implementations, where the side-channel traces are made of data traced from their execution in a emulated environment [5].

Since the alignment is a strict condition for most side-channel attacks, a large set of protections aim at making this task as difficult as possible. Excluding the inherent protocol based countermeasures (padding, session-keys) we can categorise the side-channel countermeasures in the two following categories:

- **Signals desynchronisation**: it aims at avoiding as much as possible an efficient alignment between the same point of interest of the execution among the different traces (executions). This can be achieved with hardware security features: noise generators, dummy cycles, clock jittering or power

filtering. It can also be done using software security measures: dummy operations, inserting fake or variable instructions amongst the real one, execute the operations in random order but constant time [9, 11, 38]. Doing so, it makes the alignment task difficult or even impossible when the same operation is hidden in the middle of fake but similar operations.

– **Signals de-correlation**: the principle is to make the leakage independent from the sensitive data and to prevent attackers from predicting intermediate value manipulated during the known algorithm execution. Masking and randomisation techniques are in this category. It consists of the application to the sensitive data of a randomly chosen value, named the mask [23, 1, 8, 10, 16, 36]. We will define those methods under the term *masking* in the following.

As a result, there remains a significant technical challenge of extracting a secret key when the information is present in the traces but the alignment is not obvious. Scatter addresses this technical problem, by removing or at least reducing the need of alignment.

Notations. In this paper the following notations are being used:

- $C_i = E(P_i, K)$ where $E(.,.)$ is an encryption function with plaintext P_i and a secret key K,
- $\mathcal{P}$ denote the set of the n plaintext values $\{P_0, \ldots, P_n\}$,
- $\mathcal{S}$ denote the set of n side-channel traces $\{\mathcal{S}_0, \ldots, \mathcal{S}_n\}$ collected from the measurements of $E(P_i, K)$ on the targeted device,
- $\mathcal{U}$ denote the set containing all the possible ordinates values of points of $\mathcal{S}$,
- $\#\mathcal{S} = \#\mathcal{P} = n$: the number of elements (traces) of the sets $\mathcal{S}$ and $\#\mathcal{P}$,
- $\mathcal{S}_{i,j}$ the j^{th} point of i^{th} trace $\mathcal{S}_i$ of the set $\mathcal{S}$,
- T_i denote the set of $\#T_i$ Points of Interest in the trace $\mathcal{S}_i$, so the points related to the side-channel leakage,
- $\mathcal{O}_i$ denote the set of $\#\mathcal{O}_i$ Points of No-Interest in the trace $\mathcal{S}_i$, so the points not related to the side-channel leakage,
- $\mathcal{S}_i = T_i \cup \mathcal{O}_i$,
- $G = \{g_0, \ldots, g_{b-1}\}$ is the set of $b = 2^\ell$ possible guesses for a targeted secret key ℓ-bit value k of whole key K,
  $\Rightarrow$ e.g. $G = \{0, \ldots, 255\}$ for an 8-bit guess and secret key k.
- g or g_i is one guess value and $g* = k$ is the good ℓ-bit guess
- $f(P_i, g)$ is the intermediate calculation targeted for the statistical analysis, for instance the output of the SubBytes in the first round of the AES,
- $w(.)$ is the function used to model the way the information leaks, for instance $w(x)$ is the Hamming weight of point value x in case of a Hamming weight based leakage,
- H is the set of possible values $h = w(x)$ for $x = f(P_i, g)$, for instance $H = \{0, \ldots, 8\}$ for $h = w(x)$ in the case where x are byte values.
- $\mathcal{I}_{(w)(\mathcal{P}_i, g)}$ is the set of corresponding modelized intermediate values $w\ o\ f(P_i, g)$ for the guess g and trace S_i (and plaintext P_i).

3 scatter Principle

scatter lies on the exploitation of side-channel leakages using an innovative representation of the measurements points and the way to process them. More

particularly it integrates all measurements within a window of interest and convert the data into their distribution depicting the number of times each value occurred. The distribution method is given in Algorithm 1. It is processed for each trace by choosing a relevant window of interest to target the leakage. Figure 1 shows the window selection in traces and the corresponding conversion into their respective distributions. We will denote hereafter *distribution of a trace* the outcome of the distribution process for a trace.

Algorithm 1: Distribute Trace Data

Input: trace $\mathcal{S}_i$
Output: $DTD(\mathcal{S}_i) = D_{\mathcal{S}_i}$

```
1 begin
2 |   D_{S_i} ← {0, ..., 0};
3 |   for j ∈ [0, ..., #S_i − 1] do
4 |   |   D_{S_i}[S_{i,j}] += 1;
5 |   end
6 end
```

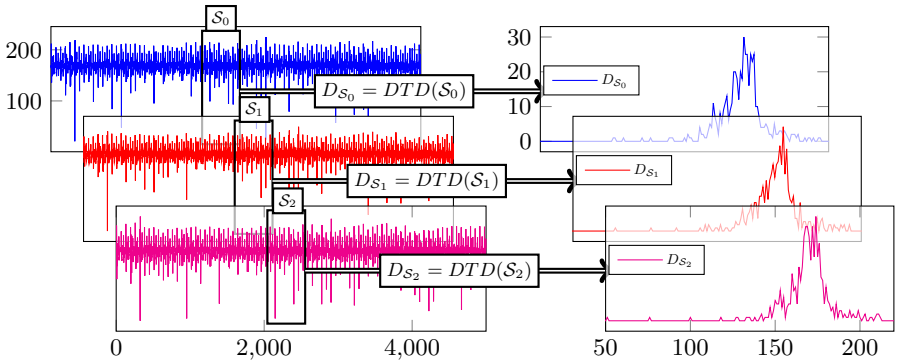

Fig. 1: Illustration of transformation from temporal traces portions to untemporal distributions.

Doing so, a new dimension is created. Indeed the useful information counts the same wherever it stands in the window. Furthermore it lies together with adjacent measurements, that are mostly independent from the estimated variable. In the following, we use the defined notation T_i for the leakage related measurement part and not related part in a trace $\mathcal{S}_\rangle$, $\#T_i$ the corresponding number of leaking points, $\#O_i$ the corresponding number of not leaking points.

The corresponding distribution for the i^{th} realisation can be expressed as:

$$DTD(\mathcal{S}_i) = DTD(T_i) + DTD(O_i) \tag{1}$$

For the next step of the attack, the distribution of each trace needs to be sorted against the estimated value over all key guesses. The estimated value

is denoted h. In this paper, h will be chosen as the hamming weight of the targeted variable. This choice is not restrictive; other models could be applied, such as the value itself or any subset. Moreover, it is important to mention that scatter is built in such a way that it does not assume anything about the linearity of the model. In other words, scatter works indifferently on either linear or non linear leakages.

Sorting the traces requires the creation of so called *accumulators*. An accumulator $Acc_{(g)(h)}$ is a three dimension vector defined for each guess g and for each value $h \in H$. This $Acc_{(g)(h)}$ is collecting trace distributions added on top of each others for each guess g and value h on all traces of $\mathcal{S}$ for the $\#U$ possible points values. To discriminate the useful information, the sorting has to be to done for each key guess g and the proper h value given by the estimated current value $w\ o\ f(P_i, g)$. Denoting $\#G$ the number of key guesses and $\#H$ the number of potential values h, the attack requires therefore the allocation of $\#G \times \#H$ accumulators. Taking the example of an 8-bit variable $f(P_i, g)$ estimated from an 8-bit key chunk (ie. byte) value k, scatter requires then the allocation of 256×9 accumulators, as $\#G = 256$ and $\#H = 9$.

In the course of the attack, each trace, once distributed, is processed $\#G$ times by accumulating it into the corresponding $Acc_{(g)(h)}$ by implementing Algorithm 2.

Algorithm 2: Accumulate Trace Data According to Guesses

Input: trace set $\mathcal{S}$,intermediate values $\mathcal{I}_{(w)(\mathcal{P}_i,g)}$
Output: accumulator set $Acc_{(g)(h)}$

```
1  begin
2      Acc_(g)(h) ← Table[null, 1, #G][null, 1, #H][null, 1, #U];
3      for i ∈ [0, ..., #S − 1] do
4          D_Si ← DTD(S_i);
5          for g ∈ G do
6              h = w o f(P_i, g);
7              Acc_(g)(h) += D_Si;
8          end
9      end
10 end
```

Once the accumulation step is performed, the corresponding values shall be normalised with the total number of point within the accumulator. Denoting X the random variable related to the measurement, and Y the random variable related to the estimation, the new expression leads to the probability density function $pdf(x) = P(X = x/Y)$.

It is important to note that scatter is not tied with a leakage model. It targets successfully hamming weight related leakage, for both linear or non linear model. Other models, such as the value itself or any subset can be considered assuming to adapt accordingly the accumulators and the way the distributions are sorted into them. For the sake of clarity, the hamming weight model is chosen in this paper. The corresponding accumulation algorithm depicted as

Algorithm 3: Normalize Accumulator

Input: accumulator $Acc_{(g)(h)}$
Output: normalized accumulator $pdf_{(g,h)}$

```
begin
    sum ← 0;
    pdf_(g,h) ← 0, ..., 0;
    for u ∈ U do
        sum+ = Acc_(g)(h)[u];
    end
    for u ∈ U do
        pdf_(g,h)[u] = Acc_(g)(h)[u]/sum;
    end
end
```

Algorithm 2 can be processed as follows returning 256×9 accumulators:

Having done this, it is worth exploring the expected behaviour in the resulted distributions $pdf_{(g,h)}$. Interestingly, different profiles should pop up depending on the correctness of the key guess. On one hand, the non informative points within O spreads themselves in a similar way regardless the key guess. More particularly, they are converging towards an asymptotical distribution related to the nature of the signal. On the other hand, each point belonging to T follows a Gaussian distribution for the wrong key guess, as they have not been sorted against their real value. This is not the case for the correct key guess, where the set of $\#T$ points sticks to the same area with a measurement data matching with the estimation. A simple graphical representation is depicted on Figure 2.

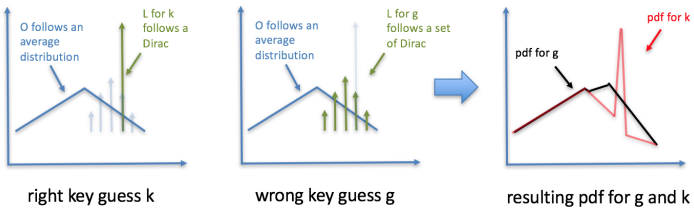

Fig. 2: Pdf functions for the right and a wrong key guess

Getting this behaviour, it becomes possible to discriminate $pdf_{(k,h)}$ related to the correct key from all other $pdf_{(g,h)}$ for a given estimation h. To achieve this, different distinguishers may be used. For instance authors in [25] suggested several ones to compare different distributions . In the context of this paper, two distinguishers will be explored using classical statistical tools in the information technology area. The first one makes use of Pearson's chi-squared (χ^2) statistical test. This test is a good fit as it expresses how much a distribution differs from a general distribution. The general formula is given as follows:

$$\chi^2 = \sum_{u \in U} \frac{(B[u] - E[u])^2}{E[u]}$$

$B[u]$ being the observed frequency of u and $E[u]$ the expected frequency of u. The application to scatter is pretty much straightforward, as it is providing a way to explore how much a accumulator $NormAcc_{(g)(h)}$ differs from the expected frequency that can be expressed as the average distribution:

$$\chi^2_{(g)(h)} = \sum_{u \in U} \frac{(pdf_{(g,h)}[u] - \frac{1}{\#G} \cdot \sum\limits_{g' \in G} pdf_{(g',h)}[u])^2}{\frac{1}{\#G} \cdot \sum\limits_{g' \in G} pdf_{(g',h)}[u]}$$

Following this, the factors can be combined over the different values h in order to provide a simple discriminant related to g:

$$\text{scatter}_{\chi^2}(g) = \prod_{h \in H} \chi^2_{(g)(h)}$$

A second distinguisher exploits the mutual information as introduced in [20]. The difference of entropy remains an appropriate factor, in spite of the presence of O. In the same way as the work described in this area, the difference of entropy is given by the formula:

$$\text{MI} = \sum_X P(X) \cdot \log(P(X)) - \sum_Y P(Y) \cdot \sum_X P(X/Y) \cdot \log(P(X/Y))$$

Which can be written in a different way expressing better the difference of behaviour we are looking for:

$$\text{MI} = \sum_Y P(Y) \cdot (\sum_X P(X) \cdot \log(P(X)) - \sum_X P(X/Y) \cdot \log(P(X/Y))$$

Mutual Information can be simply applied to scatter as any individual value in $pdf_{(g,h)}$ represents the probability $P(X/Y)$. The rest can be translated as follows: $P(Y) = P(Y = h)$ and $P(X) = Norm(\sum\limits_h Acc_{(g,h)})$ for any arbitrary g as this value yields the same regardless g. The resulting expression becomes:

$$\text{scatter}_{\text{MI}}(g) = \sum_{h \in H} P(Y = h) \cdot (\sum_{u \in U} P(X)[u] \cdot \log(P(X)[u]) - \sum_{u \in U} pdf_{(g,h)}[u] \cdot \log(pdf_{(g,h)}[u]))$$

In the following, both distinguishers are systematically kept as practical results have shown different levels of success for one or the other.

4 Attack Simulation

Practical validations of scatter effectiveness were first validated on simulated traces. This study was done to analyse scatter's resilience to an additive gaussian noise on one hand and to the level of misalignment on the other hand. Along the testing, the question of a fair comparison with existing techniques was raised. Indeed, the integration in time confers a new dimension not relevant for classical techniques. This explains why the focus has been made in extending the window of interest together with the addition of a random shift in time of the traces. By extending the size of the window of interest, this highlights how much scatter remains efficient in spite of the addition of a larger number of unrelated points (O). Since scatter is not influenced by the location of the leakage point within the window of interest, the traces were shifted in time in such a way that the leakage point always remains within the window with the same level of probability about its location. Doing so, it provides a way to analyse the technique robustness facing poorly-aligned traces, or when different patterns have to be integrated to tackle shuffled implementations. Different distinguishers or techniques have been created in the side-channel area, such as DPA, CPA, MIA, LRA. As they all assume an alignment prior to running the attack, only the CPA is used in this paper for the sake of comparison.

scatter concerns all algorithms subject to side-channel analyses. For the sake of consistency, all practical work described in this article focuses on the AES algorithm. This choice was made without removing the general impact of the new technique. Furthermore, simulation traces were generated to represent the Hamming weight of all intermediate values during an AES 128 encryption.

The simulations for widow size is f were generated following the process:

1. Generate a secret key, the value is kept for checking the validity of the results but is not exploited for the attack,
2. Generate a 16-byte long random plaintext,
3. Compute and save the output of one SBOX from SubBytes operation during the first round of AES 128 computation,
4. Generate and save $f - 1$ random bytes,
5. Convert all values into their Hamming weight,
6. Apply a chosen Gaussian noise level to simulate non-perfect measurements,
7. Apply countermeasures such as shuffling,
8. Go back to step 2 until enough simulation traces are generated.

The Gaussian noise was added using the following formula:

$$T_j = \alpha \times (HW(\text{data}) + \text{noise}(\sigma)) + \beta \tag{2}$$

$$O_j = \alpha \times (HW(\text{random data}) + \text{noise}(\sigma)) + \beta \tag{3}$$

All points share the same α and β parameters and σ is the standard deviation of the Gaussian noise applied with a mean set to 0.

The simulations results have been averaged over different campaigns in order to smooth down potential statistical inconsistencies. Within the window of interest, a random shift in time between 0 and $f - 1$ is applied so that the leakage point falls randomly with the same probability. As a result, the maximum number of traces with the leakage point properly aligned converges to

$1/f$. On following figures, the X-axis represents, in logarithmic scale, the size f of the window of interest for scatter methods results (bottom scale of the trace) and $1/f$ the portions of related points at each instant of the studied area for CPA results (top scale of the trace). The Y-axis represents the number of traces necessary to extract the key value. The Y-axis is in normal scale, except in Figure 4 where the logarithmic scale is used. The score is defined when the correct key value remains above all guesses for the given key byte. One can notice that the lowest this number is, the most successful the attack.

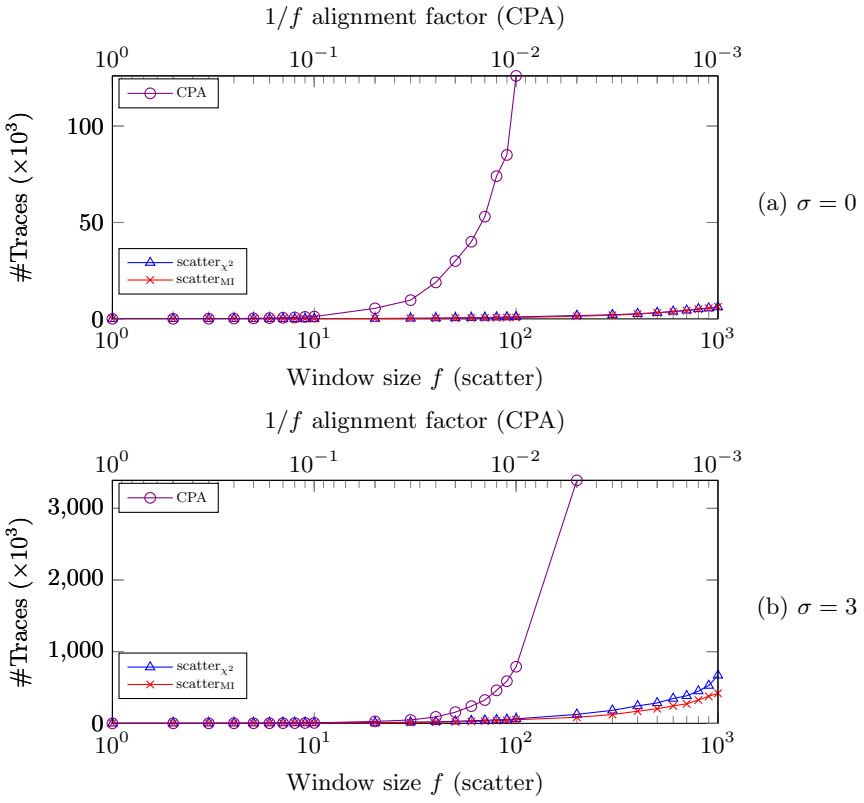

Fig. 3: Simulation of the impact of window size parameter f onto CPA and scatters methods under $\sigma = 0$ and $\sigma = 3$ noises

4.1 In Time Integration: Keep Information

Figures 3a and 3b illustrate the effectiveness of the techniques with different levels of Gaussian noise (respectively $\sigma = 0$ and $\sigma = 3$). All computations were made using both scatter_{χ^2} and $\text{scatter}_{\text{MI}}$ distinguishers. Results are depicted together with CPA computations, where the highest outcome was taken into account within the window. For both scatter techniques, all points in the window were integrated. Remarkably, scatter's outcomes remain solid with a growing window size, even though it implies the integration of an increasing number of

non informative points (O). In this configuration, $\text{scatter}_{\text{MI}}$ configuration stays slightly better than scatter_{χ^2}, but stays comparable. Unsurprisingly, CPA results decrease significantly (the number of traces needed grows faster) when the shift in time grows. Indeed, a poor alignment quickly undermines the effectiveness of the attack.

Depending on the level of noise σ, the number of traces necessary to retrieve the key changes. However, the general outcome remains similar with scatter techniques showing valuable results even with large windows.

For small windows, and consequently a fairly good alignment, the CPA technique give better results. More generally speaking, this indicates that classical techniques remain relevant when the traces are well aligned. However, the results confirm that scatter represents a real value when the condition of alignment cannot be satisfactorily fulfilled, due to a poor quality of traces or shuffling countermeasures.

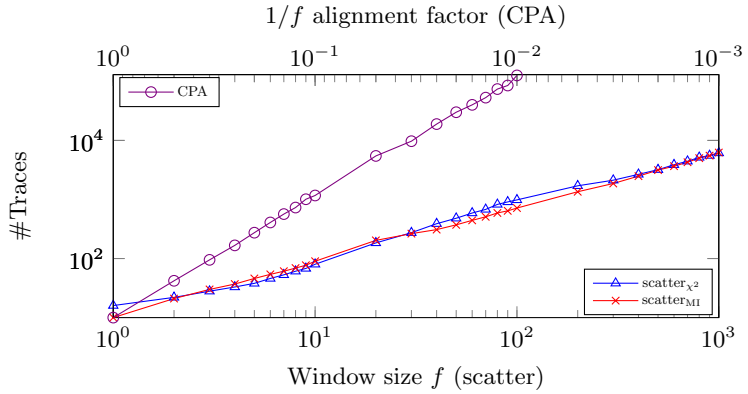

Fig. 4: Y-axis log scale applied onto Figure 3a showing results as linear

Finally, Figure 4 shows the same results as Figure 3a but with logarithmic scale on Y-axis. In that representation the results follow linear representation, illustrating scatter methods' resilience to window size f and shift in time. In this model, the behaviour is highly predictable.

4.2 In Time Integration: Accumulate Information

The previous simulations took into account one single leakage point. With the integration in time, scatter has the ability to combine different leakage points in time and consequently cumulate nicely the information available. As a result the technique gives stronger results. The benefit of accumulating different points of leakage is achieved, even though their measurement levels are not identical.

To illustrate this benefit, corresponding simulation traces were generated using the same methodology as previously. For this purpose, the traces were forged using two set of points of equal cardinality: $f/2$, the sets having point

parameters equal to (α_1, β_1) and (α_2, β_2) respectively. Compared to the previous analysis, the window of interest contains two leakage points and no longer one. Figure 5 captures the corresponding results computed with a noise $\sigma = 0$. In dashed, the outcome of the attacks using one single leakage point is depicted and, in plain, two leakage points are present within the window of interest.

The outcome of this analysis shows a positive impact of having two leakage points instead of one. Using either scatter_{χ^2} or $\text{scatter}_{\text{MI}}$ distinguishers, the gain is significant and reaches in this case a factor of at least 2. In other words, the key could be extracted with at least twice less traces in the same conditions. This result could be obviously magnified in the event of more leakages.

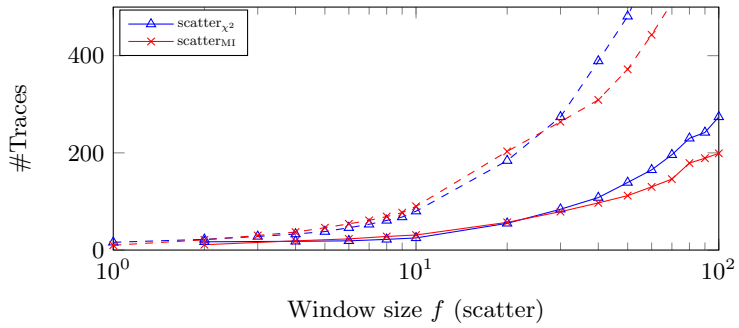

Fig. 5: Comparison results when integrating one point of leakage (dashed) vs two points of leakage (plains)

Appendix A shows other observations for different combination of α and/or β. They all lead to the same conclusion of a significant gain when dealing with more leakage points.

4.3 In Time Integration: Face the Shuffling Countermeasure

Shuffling countermeasure has been shown as an efficient way to protect algorithms. This is typically implemented by randomising the execution order of independent operations. As an example this works well when executing SBOX operations during the SubBytes of an AES. In the same vein, any sensitive operation can be concealed, by hiding it randomly in the middle of fake but similar operations. Doing this makes the identification of the correct operation difficult, or even impossible, and prevents the alignment between different execution traces.

By either integrating the whole area in the trace, or by picking up the small pieces for each individual operation, scatter has the potential to defeat such protection. Indeed, fake values are likely independent from the targeted value, and consequently do not interfere negatively during the discrimination of the right key guess. Equivalent to random noise, it can be considered that the results depicted in Figure 3 remain valid to highlight scatter performances in case of shuffling.

5 Practical Results

Further testing were performed on a hardware device. This was an unprotected AES 128 implementation running on an 8-bit AVR micro-controller. Physical measurements were performed using a low-end near-field electromagnetic probe and a low cost acquisition hardware. The global cost of the equipment used for this campaign turns around $1 000. The setup was chosen to represent a real case scenario with poor quality of traces.

The choice was made to acquire traces with an external trigger and no jitter. AES-128 encryptions were performed with a fixed key and random plain texts. In a post processing, the aim was to downgrade the initial alignment with a random shift in time as it was done on the simulation traces previously. As it can be observed on Figure 6, the quality of the traces is very poor. Without any signal processing, we did not find any way to align these traces since no pattern was found exploitable. With a random shift in time, the model looked realistic to represent an attack scenario with jittering and no good way to perform the alignment.

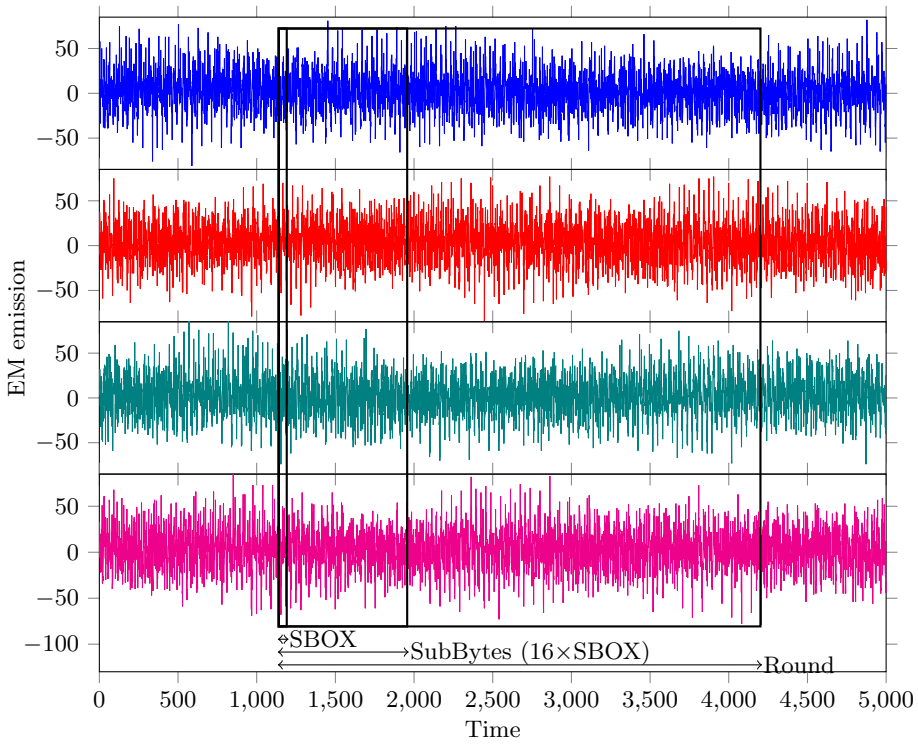

Fig. 6: Near field EM traces from a HW AES 128 execution showing that even synchronized traces do not share identifyable patterns

Figure 6 shows the processing of one round of AES 128 encryption. The operations are not visible but it gives an idea of the number of points involved.

In a similar way as the previous testing, scatter's performance is analysed with the integration of a growing number of points. A random shift in time is applied to spread the leakage uniformly within the window of interest. Once done, both scatter and CPA were applied yielding to the results depicted in Figure 7. X-axis is in a logarithmic scale and Y-axis shows the number of traces necessary to get the good key guess staying on top of all other guesses.

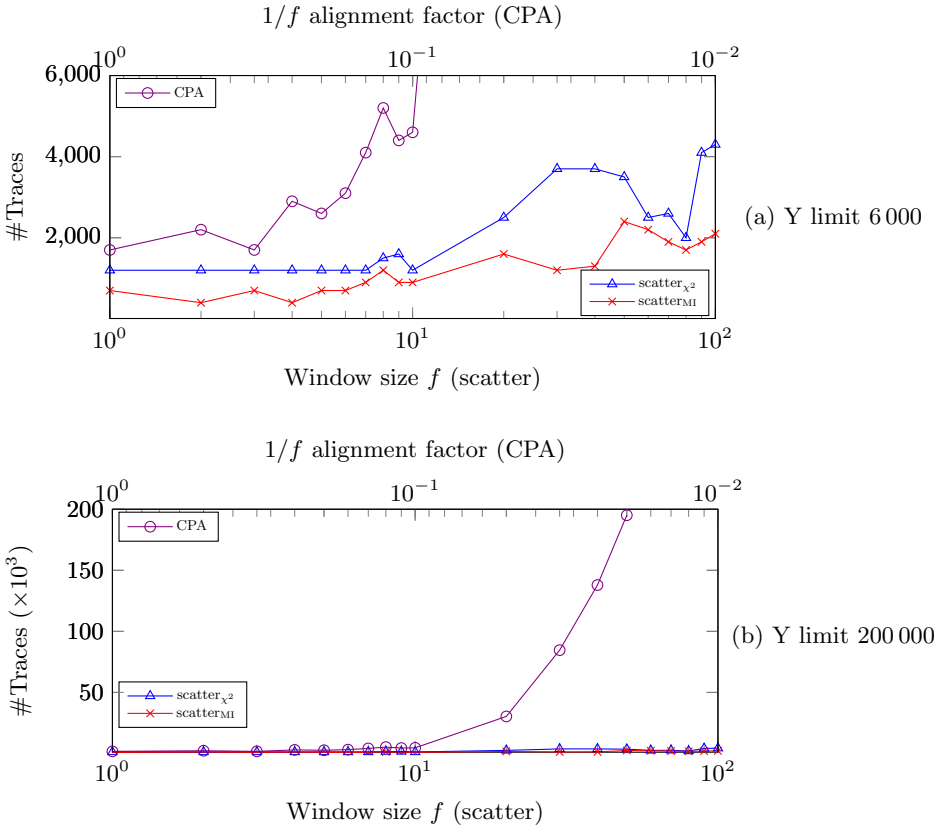

Fig. 7: Practical result similar to simulation results

These result confirm the observations made during the simulation campaign. All techniques expose the secret key. The results stay relatively in the same range for a low level of misalignment. A significant gap appears quickly showing a strong resilience of scatter techniques to the window size. The good results can be explained by multiple points of leakages present in the window of interest. In this case, both scatter_{χ^2} or $\text{scatter}_{\text{MI}}$ show good results, with slightly better outcomes for the $\text{scatter}_{\text{MI}}$ distinguisher.

For having a clear view of the integration in time, Figure 8 illustrates scatter techniques evolution together with the time representation of the AES encryption. Looking at this figure, it becomes clear that scatter techniques remains very efficient when integrating the whole SubBytes operation, including the 16

SBOXes. This means that any random order execution countermeasure would have no effect, and the attack results remain the same. Furthermore, the technique is still valid albeit less efficient when integrating the whole round operation. This gives a lot of latitude for exploring the leakage area with raw window sizes.

As a result, this practical session led to several valuable observations. First of all, scatter showed a great benefit when handling raw traces, where alignment is not obvious. This can be exploited to characterise the leakages in an exploration mode or even to break the implementation when a good alignment could not be performed. Besides, it proved that scatter has the potential to defeat several countermeasures, such as any random executions.

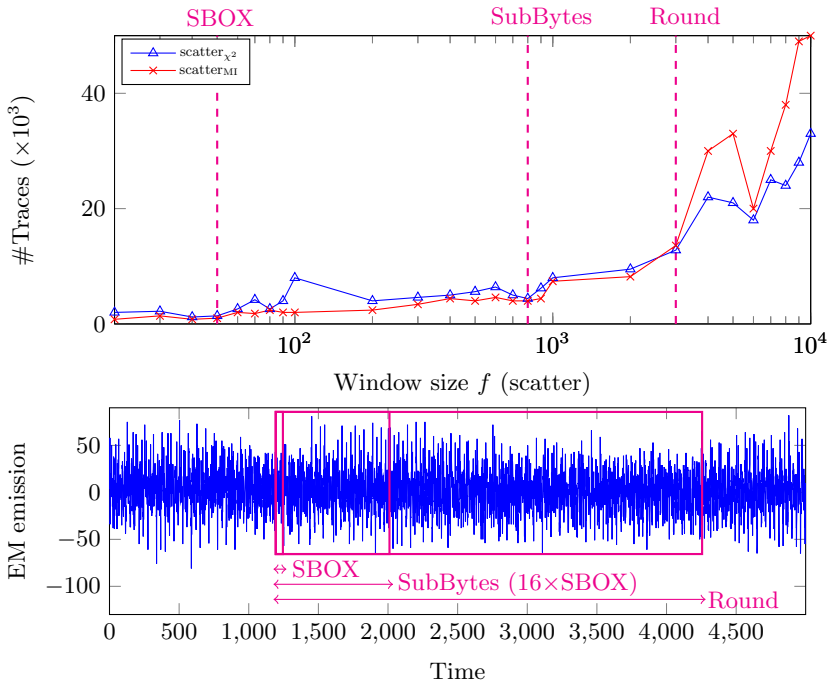

Fig. 8: Practical result of a growing window integration reachin sequencially the size of a suboperation(SBOX), algorithm step (SubBytes), a round and more

6 A comparison with window-based techniques

The comparison between scatter and preprocessing technique (such as average or FFT) is not easy. Indeed the results can vary significantly from one device to the other, or remain highly dependent on the model when dealing with simulation traces.

The main purpose of the following test was to define a realistic simulation model and highlight how much the different techniques perform in that case.

To serve this purpose a model with two sets of parameters was defined. One stating the jitter and the related window size. And a second describing the leakage model, including the value representation and the noise.

Different levels of jitter were chosen, more particularly 3, 10, 30 or 50 points. The maximum jitter value is denoted J and a dedicated set of test was performed for the given value J. With J defined, sets of traces were generated with $2 * J$ points per traces. The right value is located at the same index J for all traces. A random jitter was simulated by taking J points from an index randomly chosen between $[0, J]$. Doing so, the model integrates a jitter J and the information is always present once within the window of size J.

Regarding the trace profile, the intention was to remain generic and therefore to cover most of practical cases. The hamming weight of values was applied with couples (α, β), with (α, β) defined random in time but remaining the same from a trace to the other:

$$T_{i,j} = \alpha_j \times \mathrm{HW}(x_{i,j}) + \beta_j + \mathrm{Random}(\sigma)$$

with:

- $T_{i,j}$, the point j within the trace i
- $\mathrm{HW}(x)$ function returning the Hamming weight of the value x
- $\mathcal{N}(0, \sigma^2)$ representing a noise factor

The Gaussian noise has been chosen at $\sigma = 1$ representing a fair amount of noise without being excessive.

Taking into account that a trace is not only made of leakage points, even unrelated to the targeted variable, α was chosen equal to 0 for 70% of the points, and α randomly chosen between $[2, 6]$ for the 30% remaining points.

β was chosen randomly between $[50, 200]$ for all points. α and β range of values were defined with the aim to represent an information fitting an 8-bit oscilloscope with values staying within the range $[0, 255]$. The realization of the noise $\mathcal{N}(0, \sigma^2)$ being a real value, data in the traces were truncated to stay within the range $[0, 255]$.

$x_{i,j}$ are 8-bit values chosen randomly within $[0, 255]$. The point of interest is located at the index $j = J$.

For this study, our choice was to compare scatter with respectively a CPA average and the CPA FFT. For a proper comparison, it was important to keep the same window of points for all techniques. The window remains fix in this set of results. Applying a moving average would not help the comparison, as the same could be applied to all techniques.

For the CPA average, all points within the window were summed together and the CPA subsequently applied. On the other hand, a Fourier transformation was performed and the CPA applied to the resulting real value. The first point of the FFT was excluded as the frequency 0 is equivalent to the averaged window and therefore would give the same result as the CPA average.

The results shown below are the outcome of 200 runs with the same parameters. These runs aim at smoothing down the statistical discrepancy. The selected jitter levels J have been applied. When the maximum jitter level was increased, it resulted to get a larger window and therefore required more traces to get a strong success rate for all techniques. Doing so, the corresponding impact of the window size could be observed.

All results are captured on the following figures:

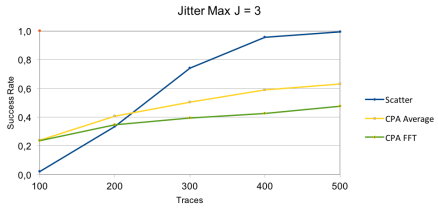

Fig. 9: comparison with jitter = 3

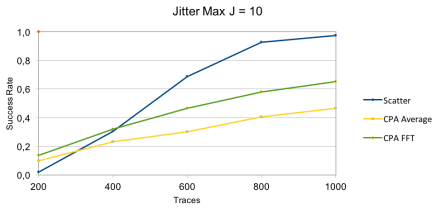

Fig. 10: comparison with jitter = 10

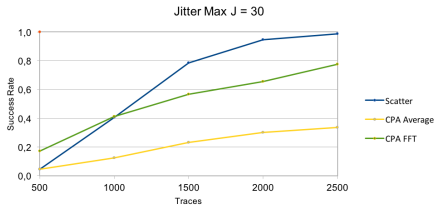

Fig. 11: comparison with jitter = 30

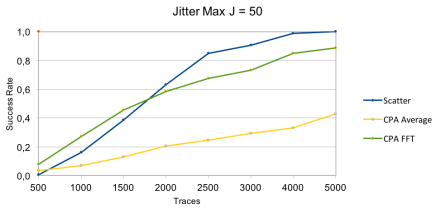

Fig. 12: comparison with jitter = 50

For all jitter levels, it can be observed that scatter success rate turns out to be higher than the other techniques after a reasonable amount of traces has been processed.

As a result, this case shows that scatter technique represents a value when jitter and desynchronisation could not be fully removed.

7 An Introduction to Higher Order Extension

As previously said, one of the most efficient countermeasure to protect products from (first-order) side-channel attacks uses random values for masking the operations (i.e. data manipulated) [36, 16, 12]. When properly implemented, masking countermeasures are effective and first order leakage is no longer available. As a result, scatter in the simple form does no longer work and needs to be extended into higher order attacks. In a same way, classical attacks had to

be adapted to extract the information when facing masking countermeasures. This led to the creation of dedicated techniques whose principle is to select and combine two (resp. n) pieces of the same trace: one targeting the value related to the secret, and the second targeting the mask value, we then perform a second-order (resp. n^{th}-order) side-channel analysis [3, 27, 29, 30, 33, 35, 39]. In these conditions, the secret can be extracted requiring a significant number of traces compared to the first order technique.

Except the techniques using Fourier transformations, second-order attacks practicality is significantly challenged by the alignment issue. Indeed, the alignment turns out to have a square complexity as it concernes both pieces of traces. As a result, the number of traces necessary to succeed a second order attack is even higher as the perfect alignment is most of the time impossible to achieve.

This is no longer the case with scatter at a higher order. Adding another dimension will create a scatter second order that will defeat all CPA leakage model and alignment restriction to achieve cutting-edge second order attack. This will be subject to further work.

8 Impact on Current State-of-the-Art

With the introduction of a new attack technique, it is important to consider the impact for secure implementations. First of all, the first order scatter technique only works for algorithm with first order vulnerabilities. However it is possible to extend the attack, this be subject to further work. The main impact concerns the hiding countermeasures that scatter may seriously threaten. Indeed, gathering the useful information within a window of interest may defeat the effort of desynchronisation, and more particularly when the shift in time remains limited, such as a clock jitter or some dummy instructions. Potentially the cost incurred by such hardware countermeasures may be questioned if scatter technique turns out to be efficient on the device.

Furthermore, practical testing on secure devices showed that scatter remains effective with large windows. This very much depends on the signal to noise ratio and the number of points of interest caught within the window. In that cases, some other well-known countermeasures may be defeated. This is the case of shuffling protections, hiding the right operation randomly within a significant number of fake operations.

The success of the attack will vary from one device to another, but no clear restriction seems to appear and scatter technique represents a clear threat for hiding type of protections. Taking into account this new paradigm, the choice of building a strong algorithmic protection looks like the right way for securing a software- or hardware-based cryptographic implementation.

9 Conclusion

This article introduces scatter, a new side-channel technique taking benefit of the integration in time of several data or measurements. On both simulated

and practical cases, the testing results have shown a high effectiveness, particularly when the set of traces has not been or could not be aligned prior to the attack. Unlike a large number of existing attacks, the scatter technique is still able to extract the secret key, even when the traces are non-aligned. As a result, the technical difficulties related to practical side-channel realisations are significantly lowered. This provides new opportunities of attack when alignment is not possible properly due to the nature of the traces. On the other hand, it makes the attacks cheaper in terms of equipment cost and expertise level. As a result, it is very likely that scatter will make practical realisations of side-channels more affordable.

The technique is generic and consequently concerns all algorithms, both symmetrical and asymmetrical. Furthermore, it opens up new exploitation opportunities, particularly when combining different pieces of information available during a sensitive algorithm execution. Extensions to higher order attacks are powerful and were briefly introduced in this article. Representing a new step in side-channels and bringing a complement to existing techniques, scatter questions the relevance of many countermeasures, more particularly those aiming at making the alignment difficult or impossible.

References

1. Mehdi-Laurent Akkar and Christophe Giraud. An implementation of DES and aes, secure against some attacks. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 2001.
2. Lejla Batina and Matthew Robshaw, editors. *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*. Springer, 2014.
3. 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.
4. Benjamin Jun and Pankaj Rohatgi. Is your design leaking keys? Efficient testing for side-channel leakage. RSA Conference, 2013.
5. Joppe W. Bos, Charles Hubain, Wil Michiels, and Philippe Teuwen. Differential computation analysis: Hiding your white-box designs is not enough. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 215–236. Springer, 2016.
6. 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.
7. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood*

Shores, CA, USA, August 13-15, 2002, Revised Papers, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.

8. Jean-Sébastien Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302. Springer, 1999.
9. Jean-Sébastien Coron. A new DPA countermeasure based on permutation tables. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *Security and Cryptography for Networks, 6th International Conference, SCN 2008, Amalfi, Italy, September 10-12, 2008. Proceedings*, volume 5229 of *Lecture Notes in Computer Science*, pages 278–292. Springer, 2008.
10. Jean-Sébastien Coron and Louis Goubin. On boolean and arithmetic masking against differential power analysis. 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 231–237. Springer, 2000.
11. Jean-Sébastien Coron and Ilya Kizhvatov. An efficient method for random delay generation in embedded software. 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 156–170. Springer, 2009.
12. Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In Moriai [31], pages 410–424.
13. Nicolas Debande, Youssef Souissi, M. Abdelaziz Elaabid, Sylvain Guilley, and Jean-Luc Danger. Wavelet transform based pre-processing for side channel analysis. In *45th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2012, Workshops Proceedings, Vancouver, BC, Canada, December 1-5, 2012*, pages 32–38. IEEE Computer Society, 2012.
14. Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. Univariate side channel attacks and leakage modeling. *J. Cryptographic Engineering*, 1(2):123–144, 2011.
15. Francois-Xavier Standaert. How (nnot) to Use WWelch's T-test in Side-Channel Security Evaluations. eprint, 2017.
16. Guillaume Fumaroli, Ange Martinelli, Emmanuel Prouff, and Matthieu Rivain. Affine masking against higher-order side channel analysis. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers*, volume 6544 of *Lecture Notes in Computer Science*, pages 262–280. Springer, 2010.
17. Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer. Stealing keys from pcs using a radio: Cheap electromagnetic attacks on windowed exponentiation. 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 207–228. Springer, 2015.
18. Daniel Genkin, Itamar Pipman, and Eran Tromer. Get your hands off my laptop: Physical side-channel key-extraction attacks on pcs. In Batina and Robshaw [2], pages 242–260.
19. Daniel Genkin, Adi Shamir, and Eran Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 444–461. Springer, 2014.

20. Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.
21. Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. stochastic methods. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, volume 4249 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2006.
22. Gilbert Goodwill and Benjamin Jun and Josh Jaffe and Pankaj Rohatgi. A Testing Methodology for Side Channel Resistance Validation. NIST Non Invasive Attack Testing Workshop, 2011.
23. 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.
24. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.
25. Yanis Linge, Cécile Dumas, and Sophie Lambert-Lacroix. Using the joint distributions of a cryptographic function in side channel analysis. In Emmanuel Prouff, editor, *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*, volume 8622 of *Lecture Notes in Computer Science*, pages 199–213. Springer, 2014.
26. Wei Liu, Liji Wu, Xiangmin Zhang, and An Wang. Wavelet-based noise reduction in power analysis attack. In *Tenth International Conference on Computational Intelligence and Security, CIS 2014, Kunming, Yunnan, China, November 15-16, 2014*, pages 405–409. IEEE Computer Society, 2014.
27. Victor Lomné, Emmanuel Prouff, Matthieu Rivain, Thomas Roche, and Adrian Thillard. How to estimate the success rate of higher-order side-channel attacks. In Batina and Robshaw [2], pages 35–54.
28. Stefan Mangard and François-Xavier Standaert, editors. *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*. Springer, 2010.
29. 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.
30. Amir Moradi, Oliver Mischke, and Thomas Eisenbarth. Correlation-enhanced power analysis collision attack. In Mangard and Standaert [28], pages 125–139.
31. Shiho Moriai, editor. *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, volume 8424 of *Lecture Notes in Computer Science*. Springer, 2014.
32. Ruben A. Muijrers, Jasper G. J. van Woudenberg, and Lejla Batina. RAM: rapid alignment method. In Emmanuel Prouff, editor, *Smart Card Research and Advanced Applications - 10th IFIP WG 8.8/11.2 International Conference, CARDIS 2011, Leuven, Belgium, September 14-16, 2011, Revised Selected Papers*, volume 7079 of *Lecture Notes in Computer Science*, pages 266–282. Springer, 2011.

33. Elisabeth Oswald, Stefan Mangard, Christoph Herbst, and Stefan Tillich. Practical second-order DPA attacks for masked smart card implementations of block ciphers. In David Pointcheval, editor, *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings*, volume 3860 of *Lecture Notes in Computer Science*, pages 192–207. Springer, 2006.
34. Romain Poussier, François-Xavier Standaert, and Vincent Grosso. Simple key enumeration (and rank estimation) using histograms: An integrated approach. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 61–81. Springer, 2016.
35. Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.
36. Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In Mangard and Standaert [28], pages 413–427.
37. Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.
38. Michael Tunstall and Olivier Benoît. Efficient use of random delays in embedded software. In Damien Sauveron, Constantinos Markantonakis, Angelos Bilas, and Jean-Jacques Quisquater, editors, *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems, First IFIP TC6 / WG 8.8 / WG 11.2 International Workshop, WISTP 2007, Heraklion, Crete, Greece, May 9-11, 2007, Proceedings*, volume 4462 of *Lecture Notes in Computer Science*, pages 27–38. Springer, 2007.
39. Michael Tunstall, Carolyn Whitnall, and Elisabeth Oswald. Masking tables - an underestimated security risk. In Moriai [31], pages 425–444.
40. 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.

A Low Influence of α and β Variations onto scatter Method

Figure 13 represent a comparison between Figure 3a results (repelled in dashed lines) where only one leakage point with parameter (α,β) value was used, and the results obtained with two leakage points with variations onto α, β, both or neither respectively on Figures 13a, 13b, 13c, 13d. Those results show that the modification of parameters have a reduced impact onto attack effectiveness. The improvement brought by a second leakage point with same parameters (Figure 13d) is lowered but stay strong when parameters are modified.

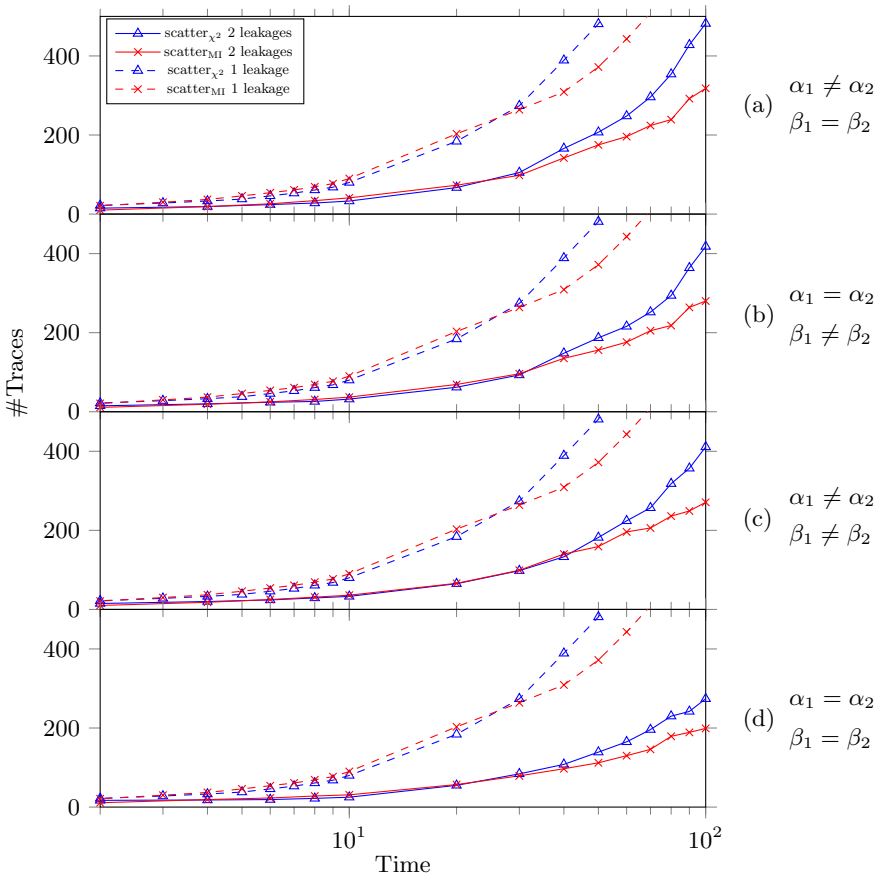

Fig. 13: Comparision one point leakage (dashed) vs two point leakages (plains), with variations of α and β coefficients.