Formal Framework for the Evaluation of Waveform Resynchronization Algorithms

Sylvain GUILLEY[1], Karim KHALFALLAH[2],
Victor LOMNE[2] and Jean-Luc DANGER[1]

[1] TELECOM-ParisTech, CNRS LTCI, FRANCE.
[2] ANSSI (Agence Nationale pour la Sécurité des Systèmes d'Information), FRANCE.

Abstract. In side-channel analysis, the waveforms can be acquired misaligned. Several algorithms have been put forward to resynchronize signals, as a pretreatment before the attack proper. In this article, we examine two of them, namely amplitude-only and phase-only correlation (abridged AOC and POC), and introduce a third one, called threshold-POC (T-POC) that corrects a flaw of the phase-only correlation. Those three resynchronization algorithms are computationally efficient insofar as they find the correct displacement in $\mathcal{O}(n \log n)$ steps per waveform made up of n samples.
Former studies on resynchronization algorithms quantified their quality by their indirect effect on side-channel attacks. We introduce in this article a formal framework for the evaluation of the resynchronization algorithms *per se*. A benchmarking on representative waveforms shows that there is an adequation between the waveforms and the most suitable resynchronization algorithm. On unprotected circuits, the intra-waveform similarity in amplitude or in phase determines the choice for either the AOC or the POC algorithm. Circuits protected by hiding countermeasures have their amplitude made as constant as possible. Therefore, the intra-waveform similarity in amplitude is lowered and the POC is better. Circuits protected by masking countermeasures have their amplitude made as random as possible. Therefore, even if the intra-waveform similarity in amplitude is high, the inter-waveform similarity is reduced; hence a trade-off between AOC and POC, namely T-POC, is the most adequate resynchronization algorithm.

1 Introduction

Side-channel analysis starts with the acquisition of a collection of waveforms, corresponding typically to the measurement of the power or to the radiated electromagnetic (EM) field of a targeted device. However, these measurements can be desynchronized for several reasons. Very often, the attacker does not have an access to a signal that indicates that the operation to be spied is beginning. Instead, the attacker can approximate the operation boundaries indirectly, for instance by sending a request and observing the response. Most embedded systems react in non-deterministic timing because they must handle internally asynchronous buffering and interruptions. In some other cases, the delay between

the external trigger and the operation processing results from a countermeasure, such as *instructions shuffling* [17] or *random delay insertion* [21,3].

Strictly speaking, misalignement of measurements, either due to approximate synchronization between the acquisition apparatus and target or to intentional desynchronization, does not prevent attacks. It is shown in [2] that the averaging of the curves is a solution to get round these drawbacks. Let us assume the desynchronization results from a displacement of the waveforms by a number of clock periods that varies in the interval $[\![0, t[\![$. We say that $t \in \mathbb{N}^*$ is the size of the desynchronization window. Then, in the extreme case where the desynchronization is uniformly distributed over $[\![0, t[\![$ (which is almost achieved by [3]), the correlation ρ between the waveforms and a leakage model with the misalignment is equal to $1/\sqrt{t}$ times that without any misalignement. Now, the speed of a correlation power analysis (CPA [1]) is directly linked to these correlation coefficients. More precisely, the average number of waveforms required to break a cryptographic implementation is equal to [12,13]:

$$3 + 8\left(\frac{Z_{1-\alpha}}{\ln\left(\frac{1+\rho}{1-\rho}\right)}\right)^2 , \tag{1}$$

where $Z_{1-\alpha}$ is the quantile of a normal distribution for the 2-sided confidence interval with error $1-\alpha$. For low values of ρ, the Eqn. (1) is $\propto \rho^{-2}$. Therefore, all in one, the number of traces to break a cryptographic implementation with a misalignment window t is roughly multiplied by $\left(1/\sqrt{t}\right)^{-2} = t$. This shows that the countermeasure is not very impeding.

Nonewithstanding, it is better for an attacker to get rid off the misalignement, so as to attack in the best conditions. Conversely, from the evaluator's standpoint, it is important to know if a prospective attacker can indeed manage to revert the misalignement. Therefore, we focus in this article on the algorithms to resynchronize the side-channel waveforms, and forget the attack or the analysis that follows.

In the sequel, we are interested in resynchronizing waveforms that have been translated in time by an integer number n of acquisition samples. This is a more general case than the abovementioned displacements of integer number of clock cycles. Indeed, modern oscilloscopes digitize waveforms at a very high sampling rate, so that many samples are captured per clock period. Additionally, we assume the clock frequency is stable and we do not address the reversal of the varying clock (VC [16,8,22]).

The rest of the article is organized as follows. In Sec. 2, the state-of-the-art resynchronization algorithms, namely AOC and POC, are introduced. One flaw of POC is described, and the threshold-POC (called T-POC) is defined. The complexity of the three algorithms is shown to be optimal. A formal framework for the evaluation of resynchronization algorithms is described in Sec. 3. The three algorithms are evaluated based on real side-channel waveforms captured from representative circuits, without and with side-channel countermeasures. Finally, the conclusions and the perspectives are given in Sec. 4.

2 Resynchronization Algorithms

2.1 Problem Statement

Theoretically, a waveform X is a series of real values, *i.e.* an element of $\mathbb{R}^{\mathbb{Z}}$. We note X_i the sample of X at date $i \in \mathbb{Z}$. Now, the measured waveform Y is said be desynchronized by an offset of k samples with respect to X if it satisfies: $\forall i, Y_i = X_{i-k}$. In practice, the (unshifted) reference X is unknown and the acquisition is limited in time. Thus, a waveform will rather be a finite set of n samples, belonging to $\mathbb{R}^n$. Given a collection of misaligned waveforms, the resynchronization problem consists in finding the correct offset for each of them. In fact, a relative offset is sufficient, because it allows to bring all the waveforms in phase; whether they are collectively offset by a constant time shift is generally not an issue. Indeed, most side-channel attacks consist in validating an hypothesis based on the maximization of a distinguisher over both time samples and key hypotheses. Thus an arbitrary collective offset in time does not change the side-channel attack's outcome. More specifically, in this paper, we focus on the resynchronization with respect to one reference waveform. The resynchronization thus comes down to the unitary problem of resynchronizing waveform Y knowing one reference waveform X.

2.2 AOC: Amplitude-Only Correlation

The cross-correlation $X \star Y$ between two waveforms X and Y is a new waveform, whose sample $i \in [\![0, n[\![$ is defined as: $(X \star Y)_i \doteq \sum_{j \in \mathbb{Z}_n} X_j \cdot Y_{j+i}$. In this notation, the time indices are considered not in the bounded interval $[\![0, n[\![$, but in the additive group $\mathbb{Z}_n$. Strictly speaking, we choose to consider the sample indexes modulo n to ease the computations, for instance in the identity involved in Eqn. (4). But in practice, it also makes "physical" sense, for instance if a waveform consists in the superposition of the clock activity and some extra signal incurred by cryptographic operations. This likely scenario is sketched in the leftmost part of Fig. 1. The straightforward cross-correlation algorithm would discard non-overlapping samples, resulting in a cross-correlation estimation over $n - k$ samples when testing for a k-sample offset. This sub-optimal solution is depicted in the middle part of Fig. 1. To avoid this loss of samples in the cross-correlation, we suggest to fold the shifted wave. The folded part, provided it contains only non-cryptographic information, will consistently match the beginning of the waveform, all the more so as the number of samples n divides the number of clock periods in the waveform. This advantageous situation is described in the right part of Fig. 1. We focus on this strategy in rest of the article.

The cross-correlation[3] can be used to recover the offset by guessing $\hat{k}$, as the offset that maximizes the cross-correlation between X and Y. Formally,

$$\hat{k} = \text{argmax}_{k \in \mathbb{Z}_n} (X \star Y)_k \quad . \tag{2}$$

[3] We would like to make clear that we name $X \star Y$ *auto-correlation* and not *correlation* to avoid the confusion with Pearson correlation coefficient.

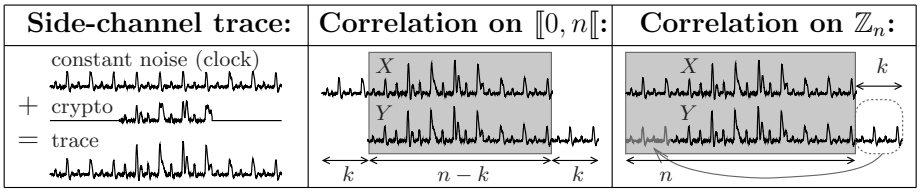

Fig. 1. Typical trace, exhibiting a special cryptographic zone (*left*), cross-correlation-based resynchronization without folding (*center*) and *idem* with folding (*right*). The shaded zone is the interval on which the "scalar product" can be computed between X and Y shifted by k samples.

Let us note ROR_k the samples circular right shift operation: $\forall i, \mathrm{ROR}_k(X)_i = X_{i-k}$, and $A \cdot B$ the coordinate-wise product: $(A \cdot B)_i = A_i \cdot B_i$. The resynchronization algorithm of Eqn. (2) is said sound, since it indeed recovers the correct offset when Y is equal to the reference waveform X circularly shifted by k':

$$\mathrm{argmax}_{k\in\mathbb{Z}_n} \left(X \star \mathrm{ROR}_{k'}(X)\right)_k = \mathrm{argmax}_{k\in\mathbb{Z}_n} \sum_{j\in\mathbb{Z}_n} X_j \cdot X_{j+(k-k')} = k' \ .$$

This result comes from the application of the Cauchy-Schwarz theorem to an auto-correlation.

The cross-correlation between two curves can be computed very efficiently using the discrete Fourier transform (DFT). The definition of the DFT and of the inverse DFT (IDFT), as per the library FFTW3 [4], is:

$$\begin{cases} \mathrm{DFT}\,(X)_i \doteq \sum_{j=0}^{n-1} X_j \cdot \exp\left(-2\pi j i \sqrt{-1}/n\right) \ , \\ \mathrm{IDFT}\,(X)_i \doteq \sum_{j=0}^{n-1} X_j \cdot \exp\left(+2\pi j i \sqrt{-1}/n\right) \ . \end{cases} \tag{3}$$

The definition of Eqn. (3) is not normalized, since it implies that: $\mathrm{DFT} \circ \mathrm{IDFT} = \mathrm{IDFT} \circ \mathrm{DFT} = n\mathrm{Id}$. In these equations, expressions are waveforms, *i.e.* elements of $\mathbb{R}^n$. Then, we have the following property: $\mathrm{DFT}(X \star Y) = \overline{\mathrm{DFT}(X)} \cdot \mathrm{DFT}(Y)$. It allows to rewrite the cross-correlation as:

$$X \star Y = \mathrm{IDFT}\left(\overline{\mathrm{DFT}(X)} \cdot \mathrm{DFT}(Y)\right)/n \, .$$

We also call the algorithm presented in this section the "amplitude-only correlation" (AOC):

$$\mathrm{AOC}(X;Y) \doteq X \star Y = \mathrm{IDFT}\left(\overline{\mathrm{DFT}(X)} \cdot \mathrm{DFT}(Y)\right)/n \, . \tag{4}$$

2.3 POC: Phase-Only Correlation

The AOC can be contrasted with the phased-only correlation (POC), described in [6,15,7]. In POC, the DFT of the reference X and desynchronized Y waveforms

are normalized prior to being multiplied. The computed quantity is:

$$\mathrm{POC}(X;Y) \doteq \mathrm{IDFT}\left(\frac{\overline{\mathrm{DFT}(X)}\cdot\mathrm{DFT}(Y)}{\left|\overline{\mathrm{DFT}(X)}\right|\cdot|\mathrm{DFT}(Y)|}\right)/n\,. \tag{5}$$

The POC is also sound, since if $Y = \mathrm{ROR}_{k'}(X)$, then:

$$\mathrm{argmax}_{k\in\mathbb{Z}_n}\,\mathrm{POC}\left(X;\mathrm{ROR}_{k'}(X)\right)_k = k'\ . \tag{6}$$

Indeed, $\mathrm{DFT}\left(\mathrm{ROR}_{k'}(X)\right)_i = \mathrm{DFT}\left(X\right)_i \cdot \exp\left(-2\pi k' i\sqrt{-1}/n\right)$. Let us note U the vector of components $U_i = \exp\left(-2\pi k' i\sqrt{-1}/n\right) \in \mathbb{C}$. Then

$$\mathrm{POC}(X;\mathrm{ROR}_{k'}(X)) = \mathrm{IDFT}\left(\frac{\overline{\mathrm{DFT}(X)}\cdot\mathrm{DFT}(X)\cdot U}{\left|\overline{\mathrm{DFT}(X)}\right|\cdot|\mathrm{DFT}(X)\cdot U|}\right)/n = \mathrm{IDFT}\left(U\right)/n\,.$$

The result of Eqn. (6) comes from the fact that:

$$\mathrm{IDFT}\left(U\right)_i = \sum_{j=0}^{n-1}\exp\left(+2\pi j(i-k')\sqrt{-1}/n\right) = n\cdot\delta_{i-k'}\ , \tag{7}$$

where δ is the Kronecker symbol, that satisfies $\delta_i = 0$ if $i \neq 0$ and 1 otherwise.

Compared to the AOC, the authors of the POC underline that the former is able to resynchronize with a resolution that is below the sampling rate. In this article, we consider only the resynchronization problem stated in Sec. 2.1, *i.e.* with an accuracy equal to that of the sample. We address the comparison of the AOC and POC algorithms empirically in the next section 2.4.

2.4 POC Flaw and Threshold-POC

We base our empirical study on waveforms taken from the DPA contest [20]. The first line of Fig. 2 shows three waveforms to resynchronize. The leftmost waveform, called $X[0]$, is the reference. On its right, $X[1]$ and $X[2]$ are two other waveforms from the same campaign that use different plaintexts, and that have been shifted artificially in time by respectively 31 and 195 samples. The exact details of these acquisitions is given in Tab. 1. These curves represent one DES encryption, that computes one round per clock period. The sampling rate is 20 Gsample/s and the DES is cadenced at a clock frequency of 32 MHz. Hence, one clock period lasts 625 samples. The waveforms are made up of $n = 20,000$ samples, thus representing 32 clock periods. The 16 clock periods where the DES hardware accelerator is computing are in the middle of the waveforms.

In this section, we compare AOC and POC algorithms on $X[q]$, $q \in \{0,1,2\}$. The application of the first method is illustrated on the second line of Fig. 2. The three figures show the amplitude of the correlation for various offsets in $[\![0,n[\![$. It appears clearly that the auto-correlation $\mathrm{AOC}(X[0];X[0])$ is the greatest for

Table 1. Detail of the encryption whose side-channel is represented in the first line of Fig. 2.

Waveform	Key	Message	Ciphertext	Offset
$X[0]$	`0x6a65786a65786a65`	`0x67c6697351ff4aec`	`0xc54baee5fc80756a`	0
$X[1]$	`0x6a65786a65786a65`	`0x29cdbaabf2fbe346`	`0x857f106855100811`	31
$X[2]$	`0x6a65786a65786a65`	`0xabb2cdc69bb45411`	`0x04385795f886e215`	195

a null offset. However, the auto-correlation features peaks, of smaller amplitude, for non-zero offsets: there is a peak (local maximum) at each clock period.

Therefore, the computation of the correlations with the shifted curves is maximal at the "correct" offsets (31 and 195), but reveals also a local maximum at the same offsets modulo the clock period. We also notice an especially large peak at the correct offset plus 16 clock periods: the reason is that DES executes in 16 clock periods and that the acquisition window happens, by chance, to be exactly equal to 32 clock periods. There is therefore an ambiguity in the correct phase to choose for the resynchronization. Nonetheless, the maximum peak (indicated by a "$\bigoplus$" sign) coincides with the actual offset.

The POC's results are shown on the line below in Fig. 2. The POC alignment of the reference waveform $X[0]$ versus itself is, as expected, a real Dirac function. This was indeed proved theoretically in Eqn. (7). Hence, the POC might look better than AOC to distinguish the correct offset from offsets modulo one clock period. Indeed, the graphs $\mathrm{POC}(X[0];X[1])$ and $\mathrm{POC}(X[0];X[2])$ show a clear peak at the correct offsets. Although the noise of the POC is high, the correct offset clearly stands out. But spurious peaks appear at high offsets, especially for $\mathrm{POC}(X[0];X[2])$, where the greatest peak occurs at an offset of $n-1$ (indicated by a "$\bigotimes$" sign). The reason is the numerical instability, during the computation, of the ratio:

$$\frac{\overline{\mathrm{DFT}(X)}\cdot\mathrm{DFT}(Y)}{|\mathrm{DFT}(X)|\cdot|\mathrm{DFT}(Y)|}$$

for small modulus values of $\mathrm{DFT}(X)$ or $\mathrm{DFT}(Y)$, because of a floating point values resolution problem (we use the C type `double`).

In order to make up for this computational artifact, we resort to a trick that consists in preventing the division by too small a quantity if the DFT modulus is small. To make up for this issue, the denominator is added a small quantity $\epsilon > 0$. Thus, the threshold-POC is defined as:

$$\text{T-POC}(X;Y) \doteq \mathrm{IDFT}\left(\frac{\overline{\mathrm{DFT}(X)}\cdot\mathrm{DFT}(Y)}{\left|\overline{\mathrm{DFT}(X)}\right|\cdot|\mathrm{DFT}(Y)|+\epsilon}\right)/n\,. \tag{8}$$

The same empirical protection of the normalization has already been used in the correlation calculation [11]. Results are shown in Fig. 2 for $\epsilon = 10^{-3}$. The

auto-correlation has a less sharp contrast, but the spurious peaks have disappeared. From a theoretical perspective, the T-POC synchronization algorithm cannot be proved sound any longer.

The value of the positive constant ϵ to be added at the denominator in Eqn. (8) is not trivial to find. To have a better idea of the normalization factor, we have computed the spectrum of a waveform. It is shown in the left part of Fig. 3. The frequency range is limited to $[\![0, n/2[\![$ because on the other half $[\![n/2, n[\![$, the curve would simply be mirrored. This is due to the fact $X[0]$ is a real waveform; thus: $\mathrm{DFT}(X[0])_{n-i} = \overline{\mathrm{DFT}(X[0])_i}$, hence $|\mathrm{DFT}(X[0])_{n-i}| = |\mathrm{DFT}(X[0])_i|$. To choose ϵ methodically, we could opt to have it equal (by convention) to a fraction of the maximum peak. The log graph on the right of Fig. 3 shows that $|\mathrm{DFT}(X[0])|$ spans 10 decades: a reasoned choice for ϵ is not obvious. Therefore, in the sequel, ϵ is rather considered an empirical parameter.

2.5 Complexity of AOC, POC and T-POC

The computation of $(X \star Y)_i$ for a given i requires n multiplications. The naive algorithm to compute the n correlations $X \star Y$ corresponding to all the possible offsets (there are n of them) runs in $\mathcal{O}(n^2)$. Now, the DFT approach reduces this complexity down to $\mathcal{O}(n \log n)$.

Indeed, one DFT or one IDFT costs $\mathcal{O}(n \log n)$. We note that for all three formulas (Eqn. (4), (5) & (8)), the $\mathrm{DFT}(X)$ on the reference waveform X can be factored for the synchronization of all the other waveforms. For the AOC, the recurrent computations consist thus only in one component-wise multiplication (n operations), one DFT and one IDFT. Regarding the POC, one additional component-wise division (n operations) is required, which does not change the computation complexity. Eventually, the T-POC also runs in $\mathcal{O}(n \log n)$, but is however the slowest method. Nonetheless, we mention that the three resynchronizations algorithms run very efficiently in practice; the resynchronization using the DFT is not the limiting operation in side-channel analysis: the attack that follows the resynchronization is the real bottleneck.

For the experiences presented in the article, we have used FFTW3, that computes Fourier transforms efficiently for every $n \in \mathbb{N}^*$. This is important as the number of samples in typical campaigns is rather a power of 10 and not a power of 2. With this FFTW3 library, all the computations can be done in complex numbers, which has the advantage of simplicity. However, the speed factor and the memory footprint can be divided by two if we consider the input is real data. The operations involve an n-sample real-to-complex DFT, that turns an array of n real numbers into an array of $n/2 + 1$ complex numbers. Thus the products and the divisions in the frequency domain are conducted with complex arrays of size $n/2 + 1$. Then, n-(logical) sample complex-to-real inverse DFT transforms the $n/2+1$ complex array into an array of $2 \times (n/2+1)$ real numbers. The elements strictly above index $n - 1$ are "padding", and thus ignored for the maximum peak research.

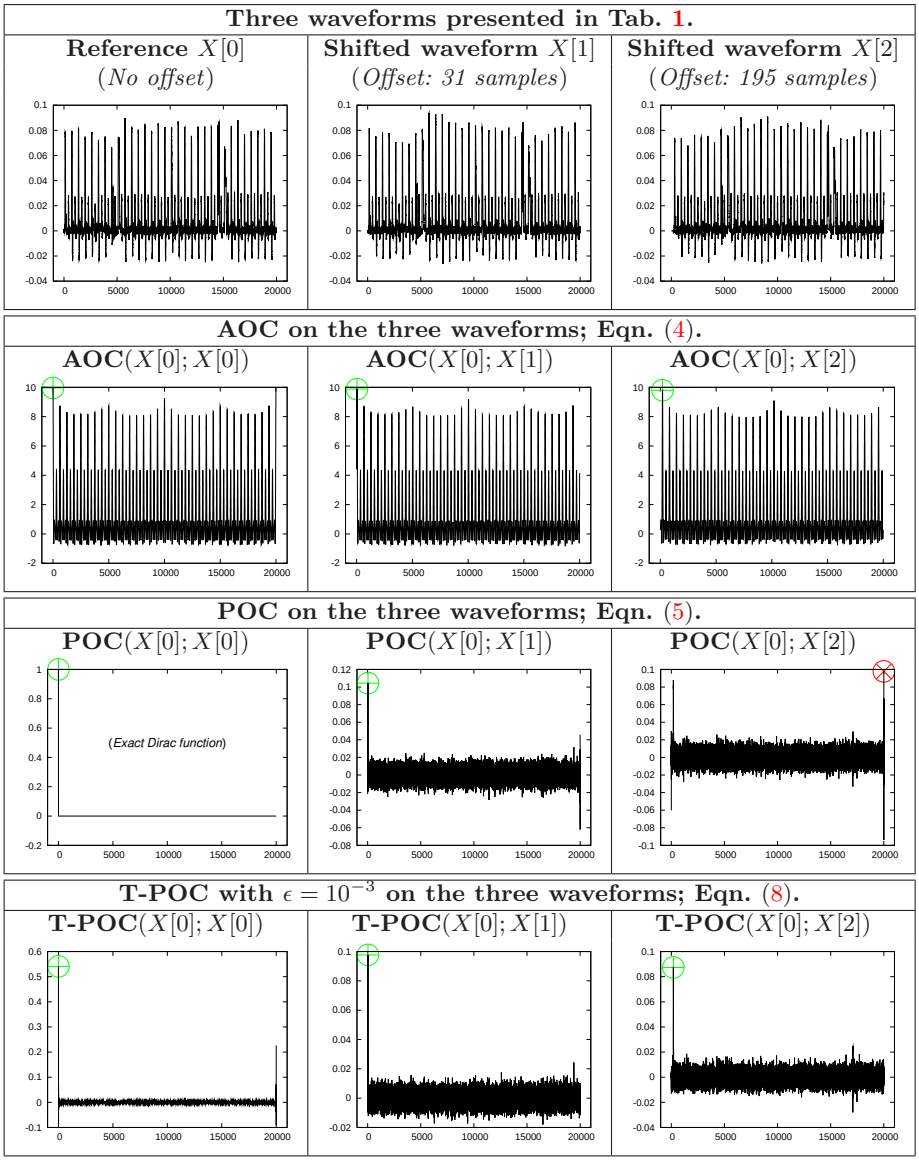

Fig. 2. Three waveforms (*topmost*) and empirical test for the resynchronization, with, from 2nd line to the 4th, respectively AOC, POC and T-POC with $\epsilon = 10^{-3}$. In these campaigns, the number of samples is $n = 20,000$. The colored circle indicates the maximum of the resynchronization algorithm. When it is green (⊕), the resynchronization is successful, whereas when it is red (⊗), it is not.

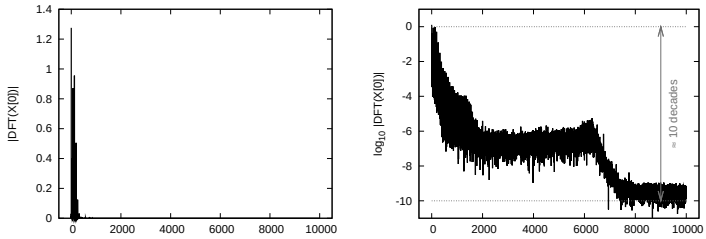

Fig. 3. Spectral power of $X[0]$, in regular scale (*left*) and in log-scale (*right*).

3 Evaluation of Resynchronization Algorithms

3.1 Formal Framework

To evaluate the several resynchronization methods fairly, we need a formal framework based on a figure of merit. Basically, such a framework assumes the knowledge of a correct synchronization and needs to assess the performance of resynchronisation algorithms based on a relative notion of resynchronization correction. The general approach is similar to the formal framework introduced in [18] in the sibling field of side-channel attacks, that introduced a "success rate" or a "guessing entropy". These metrics are fully relevant in the context of key recovery attacks, insofar as only the exact solution for the key is informative for the attacker. Indeed, an approximation on the key is useless in cryptanalysis, since all keys are equiprobable. In addition, the other way round, the key ranked second by a side-channel attack is typically decorrelated from the correct key.

The situation is different for the synchronization problem. Indeed, an approximate resynchronization (*i.e.* with an error of only one or few samples) is nearly as good as an exactly correct resynchronization, because very often the side-channel leakage remains consistent over some samples. This is all the more true as data is acquired at a large sampling rate. In the examples of the Fig. 2, a correlation power analysis (CPA [1]) leads to peaks that are about 50 samples large. This width, illustrated in Fig 4, is caused by an impedance mismatch between the side-channel sensor and the spied circuit. Thus a resynchronization algorithm still performs well if it predicts an offset a few tens of samples away from the correct offset. This means that the resynchronization cannot be solely evaluated by its success or failure rates. Indeed, we need a qualitative appreciation.

Obviously, it is better to synchronize by reducing the offset than to still make it worse. We introduce a factor of merit for the resynchronization accuracy: it is equal to the average distance to the correct resynchronization value.

This notion can be formalized. We denote by A an algorithm that rates each possible offset. In this study, A is either AOC, POC or T-POC (defined in Eqn. (4), (5) or (8)). Given two synchronized waveforms X and Y, and a maximal offset K, we set up an experiment called "**ResynchError**", in which

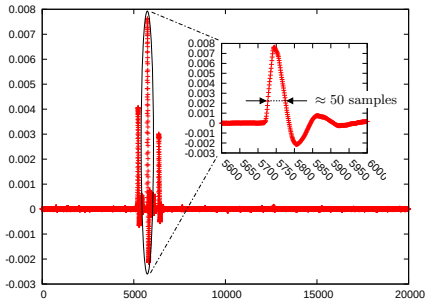

Fig. 4. Correlation power analysis (CPA) on the first round of the DES [20]. The approximate width of the peak indicates the tolerated inaccuracy of the resynchronization algorithms.

Y is artificially shifted in time by a uniformly distributed random quantity in $[\![0, K[\![$. The experiment returns the distance between the actual offset and the best rated by A. This procedure can be expressed as:

$$\begin{array}{l}\textbf{Experiment } \textbf{ResynchError}_{\mathsf{A}}(X;Y;K)\\ \left[\begin{array}{l} k' \xleftarrow{R} [\![0,K[\![;\\ \text{Return } \left|k' - \text{argmax}_{k\in\mathbb{Z}_n}\, \mathsf{A}\left(X;\text{SRL}_{k'}\left(Y\right)\right)_k\right|; \end{array}\right.\end{array}$$

where $\text{SRL}_{k'}$ operates as $\text{ROR}_{k'}$, with the sole difference it inputs k' zeros on the left end instead of reinjecting the k' samples flushed outside from the right end. A synchronized acquisition campaign C is a collection of $Q \in [\![2, +\infty[\![$ waveforms. Every waveform $C[q]$, $1 \leq q < Q$ is synchronized. The quality of the resynchronization algorithm A for waveforms randomly misaligned by offsets uniformly distributed in $[\![0, K[\![$ is assessed by:

$$\textbf{AvgResynchError}_{\mathsf{A}}(C;K) \doteq \frac{1}{Q-1}\sum_{q=1}^{Q-1} \textbf{ResynchError}_{\mathsf{A}}(C[0];C[q];K)\ . \tag{9}$$

Resynchronization algorithm A is said better than A' if

$$\textbf{AvgResynchError}_{\mathsf{A}}(C;K) \quad \leq \quad \textbf{AvgResynchError}_{\mathsf{A}'}(C;K)\ .$$

We will see in the next Sec. 3.2 that this notion does depend on the campaign C and on the maximal offset K.

We recall that the POC can be used to resynchronize with a resolution inferior than the sampling rate. Incidentally, such a method could also be applied to AOC. However, we have not tested this option, because, as will be shown in Sec. 3.2, the distinction between the resynchronization algorithm can already be clearly seen at a resolution equal to the clock period. Furthermore, modern oscilloscopes digitize waveforms at a very fast sample rate, thereby reducing the interest of fractional sample resynchronization.

3.2 Benchmarking of Representative Waveforms

We validate the average resynchronization error introduced in Eqn. (9) on five representative campaigns, corresponding to three setups. One setup is an experimental evaluation environment. On the three setups, the same DES algorithm (*i.e.* synthesized from the same VHDL source code) is run. The first setup is that of the DPA contest first edition [20], where the DES is an ASIC and where the acquisitions are averaged 64 times by the oscilloscope. The second one is carried out on an ASIC but with unaveraged acquisitions. Eventually, the third setup is identical to the second one, except that the device under analysis is an FPGA, and not an ASIC. More details are provided in Tab. 2. We have selected very different setups on purpose to gather various representative side-channel types.

Table 2. The three setups studied.

Setup	**Samples/clk**	**Fclk [MHz]**	**Nature**	**Device**
#1	625	32.000	Power	ASIC (0.13 μm technology, 1.2 Volt)
#2	150	33.333	Power	ASIC (0.13 μm technology, 1.2 Volt)
#3	120	8.333	EM	FPGA (0.13 μm technology, 1.5 Volt)

The second and third setups are also used to implement side-channel resistant versions of DES. On the second setup, one campaign is done on a hiding countermeasure [13, Chp. 7]. On the third setup, one campaign is done on a masking countermeasure [13, Chp. 9]. In the sequel, we represent the five studied campaigns as per Fig. 5, that gives one raw trace for each campaign.

The average resynchronization error is represented in Fig. 6 for those five campaigns, based on $Q = 1,000$ artificial shifts. It gives, for the AOC and the POC (with 4 values of ϵ) the mean absolute error of resynchronization **AvgResynchError** as a function of the synchronization error window K.

The figure 6 reveals very different behaviors of resynchronization performance $K \mapsto \mathbf{AvgResynchError}(C; K)$. Notably, the setups #1 and #3 fail to have their unprotected designs properly resynchronized for some algorithms.

In Setup1_Ref, large errors occur for the POC and the T-POC with the smallest correction value $\epsilon = 10^{-6}$. These errors increase almost linearly with the desynchronization amplitude. More precisely, there is an improvement when the desynchronization maximal value is not a multiple of half the clock period. This observation shows that the computational flaw identified in the POC in Sec. 2.4 is the main limitation to the resynchronization on this campaign.

Interestingly enough, the campaign Setup3_Ref features an opposite behaviour. The AOC and the T-POC with a large $\epsilon = 10^3$ coefficient both fail. We notice that when ϵ becomes larger and larger, then T-POC tends towards AOC, since the denominator in Eqn. (8) becomes negligible. The reason for the AOC to fail can be accounted by the nature of the setup: the measurements are noisy,

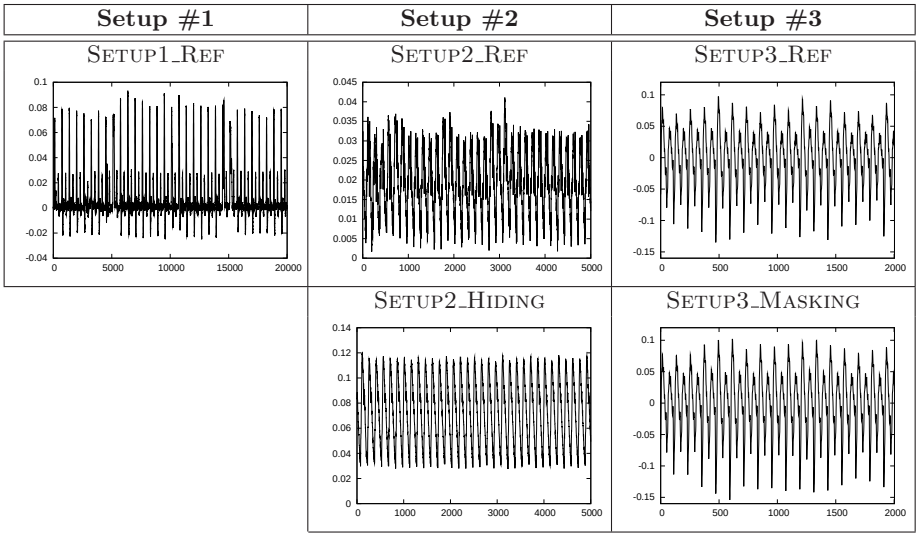

Fig. 5. Raw traces examples for the five investigated campaigns.

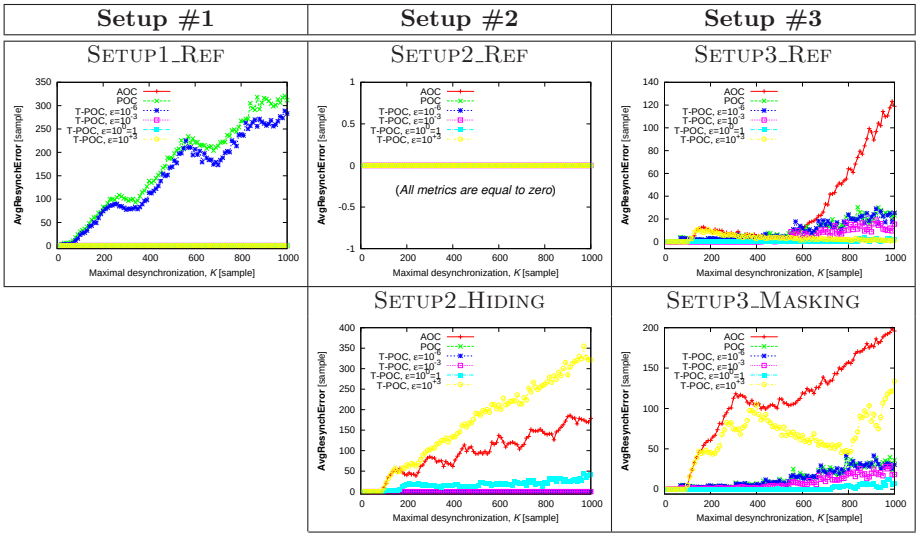

Fig. 6. Average resynchronization performance for the five campaigns presented in Tab. 5.

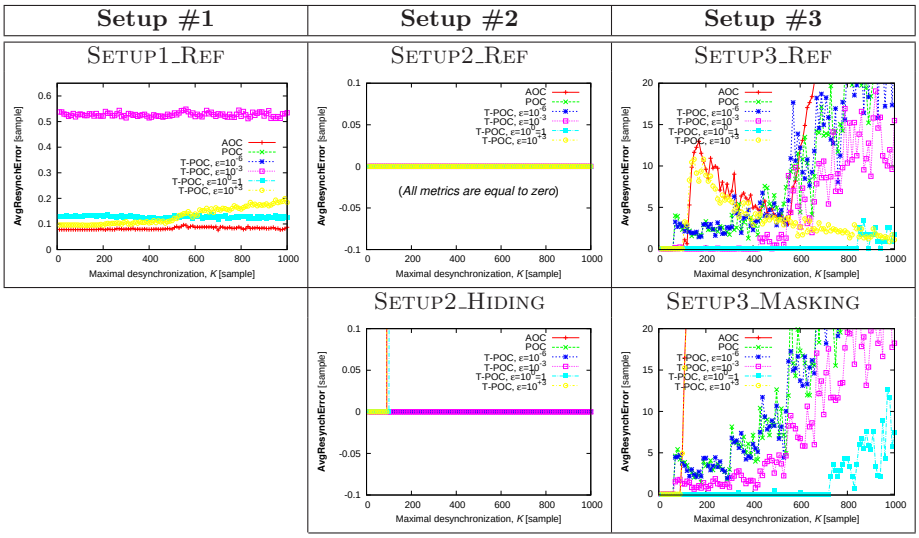

Fig. 7. Average resynchronization performance for the five campaigns; vertical zoom on Fig. 6, that focuses on errors that do not diverge with K.

which makes the identification of the correct phase by the analysis of the waveforms amplitude very error-prone. The cause of high noise in the measurement setup #3 is threefold:

1. Unaveraged measurements have a greater quantification noise than traces that have been averaged;
2. FPGAs activate a lot of logic per single logical event in the netlist, which increases the algorithmic noise [1];
3. EM measurements are notoriously more noisy than power measurements.

Nonetheless, this noise is independently and identically distributed (*iid*) over the samples. Therefore, the phase information, which is collective over one clock period, is less affected. In particular, because of the high level of noise, the DFT transform of the setup #3 waveforms is rich in frequencies, and therefore varies less than that of Fig. 3. Therefore the POC flaw does not manifest. We observe in this campaign that the pure POC neither succeeds in resynchronizing well the curves, but that T-POC with $\epsilon = 1$ is almost successful 100% of the time. Thus, for this campaign, the best resynchronization algorithm is a tradeoff between amplitude- and phase-correlation.

The campaign Setup2_Ref is perfectly resynchronized with all the studied algorithms. The explanation clearly stands out by looking at the sample waveform provided for this campaign in Fig. 5. Every waveform has both a very clear shape (which favors amplitude-related matching techniques) and an elaborate spectrum (both clock-level and higher frequencies are already visible on the time-domain trace, which is beneficial for phase-related matching techniques).

It is interesting to zoom on the resynchronization performance for the campaigns carried out on unprotected circuits. These graphs are provided with in Fig. 7. AOC definitely best realigns the campaign Setup1_Ref. This is due to the extremely accurate acquisition in amplitude; notably, the averaging of the waveforms helps make resynchronization with vertical values reliable. The characteristic shape for these waveforms, associated with their high resolution, makes each of them very recognizable. The resemblance (intra-waveform similarity) outperforms the difference between the waveforms (acquired with different plaintexts). The opposite conclusion can be drawn for the noisy Setup3_Ref campaign: even if we manage to identify some points of larger amplitude than others in each individual waveform, the noise makes each waveform dissimilar in amplitude. As the phase is noisy too, the T-POC is the best tool to extract the synchronization between waveforms of campaign Setup3_Ref.

Let us now study the two campaigns on protected implementations, namely Setup2_Hiding and Setup3_Masking. It is straightforward to see in the corresponding graphs of Fig. 6 that the AOC is the worst resynchronization algorithm. Two compelling arguments can explain this. On power-constant circuits, the goal of the countermeasure is to balance the side-channel leakage, by having its amplitude as constant as possible. Thus, it is expected that resynchronization based on amplitude-matching fail. However, it has been noted in [19,10] that small (much beneath the clock period) discrepancies in evaluation dates could exist. This phenomenon is referred to as "early propagation effect" in the specialized literature. The success of the resynchronization using the phase information of the waveforms might be a confirmation of this effect. On masked circuits, the shapes of the waveforms are forced to look random in a view to mitigate first-order side-channel attacks. It is therefore no surprise if AOC is ineffective in average. Nonetheless, it is noteworthy that the phase of the signals carry information about the algorithm scheduling. We conjecture that despite the additional amount of noise carried out by the masking countermeasure, the sequence of operations (registers evaluation, then maybe the addressing of a RAM, or the activity that comes from the control block, *etc.*) might be a characteristic signature of the DES operations.

All in one, the Fig. 7 shows that campaigns acquired from a protected circuit are more difficult to synchronize than those acquired from unprotected circuits implemented on the same setups. Nevertheless, our general noting is that the two prominent countermeasures (hiding and masking) aim at dissimulating the information in amplitude, but that unexpectedly the phase is still useful to achieve a correct waveforms realignment. Those conclusions are in line with many papers focusing on DFT attacks [5,9,14]. They all conclude that the side-channel waveforms exhibit extremely distinguishable features once turned into the frequency domain. We note that the best value for ϵ happens to be small ($\epsilon \leq 1$) for Setup2_Hiding: all the information lays in the waveforms phase. The optimal ϵ for Setup3_Masking is exactly the same as for Setup3_Ref. Indeed, the masking simply increases the algorithmic noise level, but does not fundamentally affect the acquisition.

4 Conclusions and Perspectives

Side-channel measurements can be desynchronized for various reasons, especially in the common case of acquisitions where a reliable trigger signal is not available. This study introduces a formal practice-oriented evaluation framework for resynchronization algorithms. In this article, we compare several approaches to realign the waveforms. We conclude that, in the absence of countermeasures, if the acquired signal is of excellent vertical quality, then the amplitude should be used to resynchronise the signals. Otherwise, in the case of noisy measurements, the phase-based correlations are better techniques. We notice that under some circumstances, the genuine version of the phase-only correlation (POC) is not efficient, and we introduce the threshold POC (*aka* T-POC). If a countermeasure is employed, then, undoubtedly, the T-POC (including T-POC with $\epsilon = 0$, *i.e.* the original POC) is the best realignment algorithm. The reason is that state-of-the-art side-channel countermeasures aim at impeding amplitude-level waveforms variation, but neglect to protect the information carried by the phase. Therefore, using POC or T-POC algorithms, we show how to successfully resynchronize protected waveforms.

Several questions remain however open. For instance, what is the optimal threshold value ϵ involved in T-POC? Also, we wonder if a mixed resynchronization techniques (for instance based on wavelets, that feature a compromise between time and frequency) could bridge the gap between amplitude-only and phase-only correlation algorithms.

References

1. Éric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *CHES*, volume 3156 of *LNCS*, pages 16–29. Springer, August 11–13 2004. Cambridge, MA, USA.
2. Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In *CHES*, LNCS, pages 252–263, London, UK, August 2000. Springer-Verlag.
3. Jean-Sébastien Coron and Ilya Kizhvatov. Analysis and Improvement of the Random Delay Countermeasure of CHES 2009. In *CHES*, volume 6225 of *LNCS*, pages 95–109. Springer, August 17-20 2010. Santa Barbara, CA, USA.
4. Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, February 2005.
5. Catherine H. Gebotys, Simon Ho, and Chin Chi Tiu. EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA. In *CHES*, volume 3659 of *LNCS*, pages 250–264. Springer, August 29 – September 1 2005. Edinburgh, UK.
6. Naofumi Homma, Sei Nagashima, Yuichi Imai, Takafumi Aoki, and Akashi Satoh. High-Resolution Side-Channel Attack Using Phase-Based Waveform Matching. In *CHES*, volume 4249 of *LNCS*, pages 187–200. Springer, October 10-13 2006. Yokohama, Japan.
7. Naofumi Homma, Sei Nagashima, Takeshi Sugawara, Takafumi Aoki, and Akashi Satoh. A High-Resolution Phase-Based Waveform Matching and Its Application to Side-Channel Attacks. *IEICE Transactions*, 91-A(1):193–202, 2008.

8. Mohaned Kafi, Sylvain Guilley, Sandra Marcello, and David Naccache. Deconvolving Protected Signals. In *ARES/CISIS*, pages 687–694, Fukuoka, Kyūshū, Japan, March, 16th – 19th 2009. IEEE Computer Society Press.
9. Timo Kasper, David Oswald, and Christof Paar. EM Side-Channel Attacks on Commercial Contactless Smartcards Using Low-Cost Equipment. In *WISA*, volume 5932 of *LNCS*, pages 79–93. Springer, August 25-27 2009. Busan, Korea.
10. Konrad J. Kulikowski, Mark G. Karpovsky, and Alexander Taubin. Power Attacks on Secure Hardware Based on Early Propagation of Data. In *IOLTS*, pages 131–138. IEEE Computer Society, 2006. Como, Italy.
11. Thanh-Ha Le, Jessy Clédière, Cécile Canovas, Bruno Robisson, Christine Servière, and Jean-Louis Lacoume. A Proposition for Correlation Power Analysis Enhancement. In *CHES*, volume 4249 of *LNCS*, pages 174–186. Springer, 2006. Yokohama, Japan.
12. Stefan Mangard. Hardware Countermeasures against DPA – A Statistical Analysis of Their Effectiveness. In *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004. San Francisco, CA, USA.
13. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, December 2006. ISBN 0-387-30857-1, http://www.dpabook.org/.
14. Edgar Mateos and Catherine H. Gebotys. A new correlation frequency analysis of the side channel. In *Proceedings of the 5th Workshop on Embedded Systems Security*, WESS '10, pages 4:1–4:8, New York, NY, USA, 2010. ACM.
15. Sei Nagashima, Naofumi Homma, Yuichi Imai, Takafumi Aoki, and Akashi Satoh. DPA Using Phase-Based Waveform Matching against Random-Delay Countermeasure. In *ISCAS*, pages 1807–1810. IEEE Computer Society, May 27-20 2007. New Orleans, Louisiana, USA. DOI: 10.1109/ISCAS.2007.378024.
16. Denis Réal, Cécile Canovas, Jessy Clédière, M'hamed Drissi, and Frédéric Valette. Defeating classical Hardware Countermeasures: a new processing for Side Channel Analysis. In *DATE*, pages 1274–1279. IEEE Computer Society, March 10-14 2008. Munich, Germany.
17. Matthieu Rivain, Emmanuel Prouff, and Julien Doget. Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers. In *CHES*, volume 5747 of *LNCS*, pages 171–188. Springer, September 6-9 2009. Lausanne, Switzerland.
18. François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, volume 5479 of *LNCS*, pages 443–461. Springer, April 26-30 2009. Cologne, Germany.
19. Daisuke Suzuki and Minoru Saeki. Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style. In *CHES*, volume 4249 of *LNCS*, pages 255–269. Springer, October 10-13 2006. Yokohama, Japan.
20. TELECOM ParisTech SEN research group. DPA Contest (1st edition), 2008–2009. http://www.DPAcontest.org/.
21. Michael Tunstall and Olivier Benoît. Efficient Use of Random Delays in Embedded Software. In *WISTP*, volume 4462 of *Lecture Notes in Computer Science*, pages 27–38. Springer, May 9-11 2007. Heraklion, Crete, Greece.
22. Jasper G. J. van Woudenberg, Marc F. Witteman, and Bram Bakker. Improving Differential Power Analysis by Elastic Alignment. http://www.riscure.com/fileadmin/images/Docs/elastic_paper.pdf.