Portability of templates

M. Abdelaziz Elaabid, Sylvain Guilley

Portability of Templates

M. Abdelaziz Elaabid[1]
Sylvain Guilley[2]

[1] Université de Paris 8, Équipe MTII, LAGA
2 rue de la liberté, 93526 Saint-Denis Cedex, France.
elaabid@TELECOM-ParisTech.fr

[2] Institut TELECOM / TELECOM ParisTech, CNRS LTCI (UMR 5141)
Département COMELEC, 46 rue Barrault,
75 634 PARIS Cedex 13, France.
sylvain.guilley@TELECOM-ParisTech.fr

Abstract

Template attacks consist of two stages: a profiling and a matching step. This way of attacking a circuit can be shown to be optimal when the profiling exactly describes the side-channel leakage of the circuit to be attacked. On the contrary, this article focuses on identifying the problems that arise when there is a discrepancy between the templates and the traces to match. Based on a real-world case-study, we show that two phenomena can hinder the success of template attacks when the precharacterized templates are outdated: the traces can be desynchronized and the amplitudes can be scaled differently. We observe that the consequence of these distortions can be as dramatic as ranking the correct key last, which is the worst degradation possible for a side-channel distinguisher; Since an attacker is usually interested in the first keys in the rankings. Then we suggest two ways to correct the templates mismatches: waveform realignment and acquisition campaigns normalization. After this processing, it appears that the template attacks almost do not loose any efficiency in terms of success rate and guessing entropy with respect to an attack with ideal templates.

Keywords: Template attack ; Resynchronization ; Side-channel attacks.

1 Introduction

Template attacks [10] are side-channel attacks based on a precharacterization phase. This phase is carried out off-line on an profiling circuit with known key, once for all the subsequent attacks. It aims at preparing templates that characterize the leakage of the circuit, in order to continue with the most powerful attacks, namely Bayesian attacks. These two phases are also referred to as "training" and "matching".

To our best knowledge, most state-of-the-art literature contains only proof-of-concept attacks, where the traces intended for the training and the matching phases are acquired consecutively

on the same circuit [10, 18, 3, 12, 4, 1, 21, 15, 8, 9]. This case is the most favorable to the attacker, since the templates are built in extremely similar conditions to that of the real attack. In more realistic cases, an attacker does not train and do the attack on the same circuit at the same moment. The effect of time can add new factors that may make the attack more difficult. In fact, the adversary should conduct traces acquisition (a difficult and error-prone experimental process) ideally in the same way for the training and for the matching. Every step involved in an acquisition, from the characteristics of the measurement resistor to the configuration of the oscilloscope, should be as similar as possible. To assess in which respect acquisition discrepancies can impede an adversary, we conduct some experiments that consist of changing conditions to study the induced effects. We identified two major problems: the curves desynchronization in time and in amplitude. We explore some strategies available to an adversary using template attacks to bypass them.

Recent works have formalized the quantity of information that is lost when templates are not portable. Notably, the notation of perceived information is introduced in [20] and applied on a protected circuit in [19]: it is equal to the information leakage an evaluator estimates using a model (templates obtained from another circuit) that differs from the actual leakage of the targeted circuit (on which matching is done). It is shown that this information is always less or equal to that of the hypothetical case where the model exactly describes the targeted circuit. Thus, the mismatch between the templates the targeted circuit underestimates the leakage. In this paper, we do not focus on information theoretic metrics, but rather on attack metrics. This distinction has been introduced in [22], to make a difference with the vulnerability assessment (leakage metrics) and the security assessment (attack metrics). Thus, we intend to practically conduct template attacks using the same device at different times. In addition, we target unprotected implementations. Thus, the differences in the templates due to process variation is expected to be much less dominant than in [19], where the study is conducted on a purportedly power-constant implementation. This motivates the focus of this paper on attack metrics.

The rest of the article is structured as follows. Our methodology, intentionally practice-oriented, is first explained in Sec. 2. Then, we recall in Sec. 3 how principal subspaces can be used to reduce the dimensionality of templates. Resynchronization techniques to temporally realign campaigns are discussed in Sec. 4. Their result is given in Sec. 5. The correction of vertical scaling mismatches is presented in Sec. 6. Eventually, Sec. 7 is the final evaluation of the portability of templates with horizontal and vertical correction. The conclusions and perspectives can be found in Sec. 8.

2 Methodology

This article aims at investigating the practical effects of possible experimental conditions mismatches between training and matching phases on the attacks' success rate and guessing entropy [22]. In this study, we focus on the effect of time in a template attack: the reproducibility of measurement setups is challenged. We choose to consider three sets of measurements acquired on an ASIC implementing DES:

1. **Campaign A**: 80,000 measurements obtained in year 2006 at nominal voltage (about 1.2 V) serve to build the templates,

2. **Campaign B**: 50,000 measurements obtained in year 2010 at nominal voltage (about 1.2 V) on the same ASIC are used for matching,

3. **Campaign C**: 50,000 measurements obtained in year 2010 at reduced voltage (about 1.0 V) on the same ASIC are also used for matching.

The goal of the campaign C is to provide a comparison of two campaigns (B and C) that were carried out close in time, but with slightly different experimental conditions. Here, the variation comes from the power supply. More precisely, the common features between the A and B/C campaigns are listed below:

- The same ASIC is tested.
- It is soldered on the same evaluation board (described in Appendix B of [13]).
- The same differential voltage probe (Agilent 1132A) is used to measure the voltage drop over a resistor placed between the ground of the evaluation board and the ground of the ASIC.
- The same oscilloscope (Agilent infiniium 54855A DSO) is used, with exactly the same setup file (refer to Tab. 1 that describes the settings of the several acquisitions).

The differences between the A and B/C campaigns are:

- The wiring between the evaluation board and the oscilloscope has been redone (thereby incurring maybe some delay variations, for instance on the trigger line); it is different between A and B, A and C, but equal for B and C. Indeed, the only modification done between campaigns B and C has consisted in turning the power supply button to reduce the voltage from 1.2 to 1.0 volts.
- The spying resistor has been changed between campaigns A and B/C.
- The ASIC has been aging, and has thus undergone hot-carrier-induced degradation [5] (an effect that is hard to quantify on a circuit that was not designed to be tested against aging).

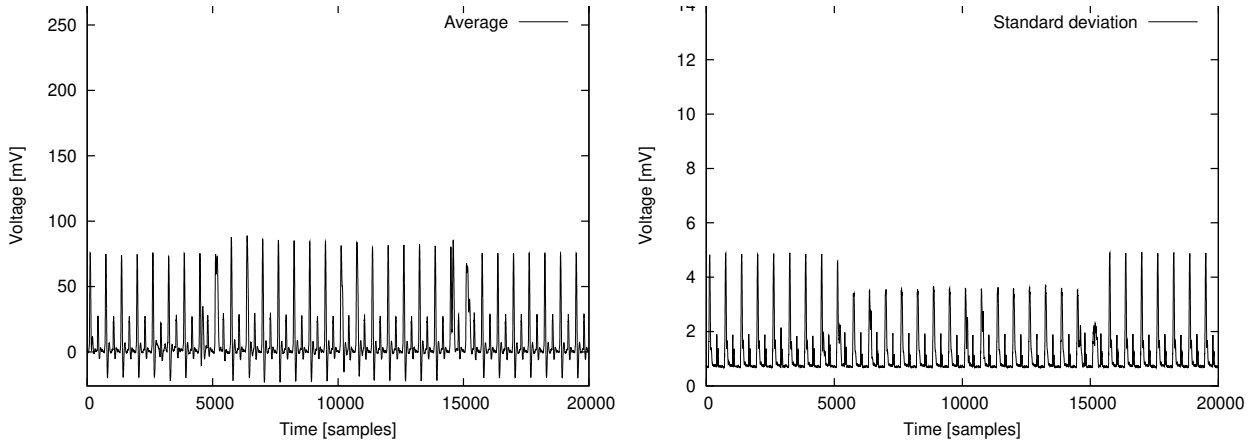
The first and second order statistics on the three campaigns are given in Fig. 1, 2 and 3.

It can be noticed that messages vary randomly, and that the encryption starts around time sample 5,000 and stops at about sample 15,000. The increase of power consumption during these sixteen clock cycles coincides with the sixteen rounds of DES encryption. As can be seen on the same figures (right part), the accompanying decrease of the standard deviation is consistent with the fact the control logic and lines become deterministic during the encryption, thereby subtracting a noise contribution from the encryption process. Indeed, in our architecture, the DES datapath is left enabled in the "idle" state, which makes it produce high and inconsistent activity outside encryption timing windows. From the variance curves, other interesting comments can be done:

- Regardless of the acquisition campaign (A, B or C), there is a constant noise level (slightly below 1 mV), that represents the intrinsic acquisition noise (incurred by quantization and thermal fluctuations).

Table 1: Setup of Agilent's files in the three campaigns studied in this article – this data has been extracted after the campaigns from the Agilent "`.bin`" files (whose format is described in [2] at pages 409-413).

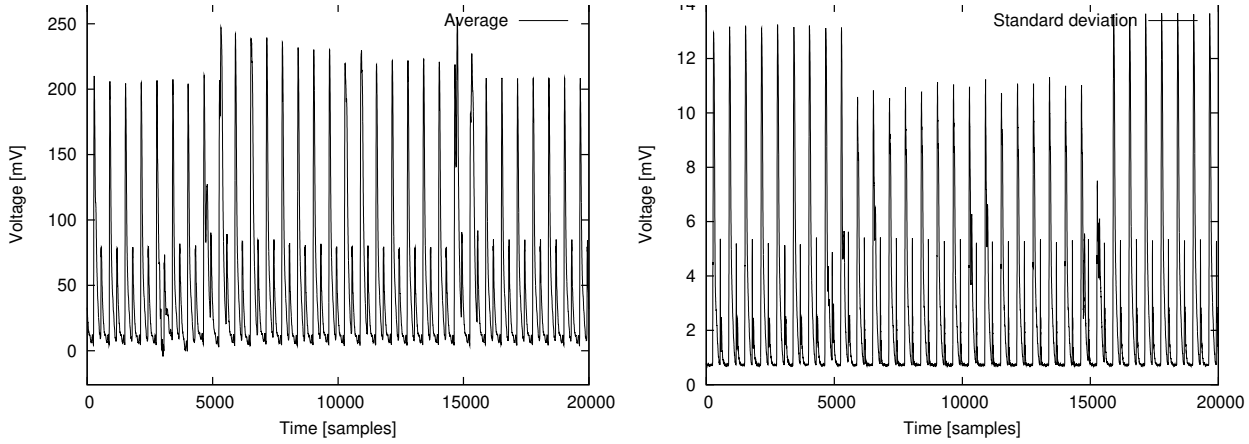
Campaign A	**Campaigns B & C**
### Binary Header Format	### Binary Header Format
# Version: 10	# Version: 10
# File Size: 80176	# File Size: 80176
# Nbr of Waveforms: 1	# Nbr of Waveforms: 1
### Waveform Header	### Waveform Header
# Header Size: 140	# Header Size: 140
# Waveform Type: HORIZONTAL_HISTOGRAM	# Waveform Type: HORIZONTAL_HISTOGRAM
# Nbr of Waveform Buffers: 1	# Nbr of Waveform Buffers: 1
# Nbr of Hits: 20003	# Nbr of Hits: 20003
# X Display Range: 7.4228448e-51	# X Display Range: 7.4228448e-51
# X Display Origin: 2.32682e-06	# X Display Origin: 2.32682e-06
# X Increment: 5e-11	# X Increment: 5e-11
# X Origin: 2.32676844e-06	# X Origin: 2.32676844e-06
# X Unit: 0	# X Unit: 0
# Y Unit: 0	# Y Unit: 0
# Date: 8 APR 2006	# Date: 27 MAY 2010
# Time: 11:31:01	# Time: 20:16:21
# Model: 54855A:	# Model: 54855A:
# Label: channel 3	# Label: channel 3
# Tag value: 0	# Tag value: 0
# Index: 0	# Index: 0
### Waveform Data Header:	### Waveform Data Header:
# Waveform Data Header Size: 12	# Waveform Data Header Size: 12
# Buffer Type: NORMAL_32	# Buffer Type: NORMAL_32
# Bytes Per Point: 4	# Bytes Per Point: 4
# Buffer size (in bytes): 80012	# Buffer size (in bytes): 80012



(a) Mean trace.

(b) Standard deviation.

Figure 1: Mean and standard deviation trace for the campaign A.



(a) Mean trace.

(b) Standard deviation.

Figure 2: Mean and standard deviation trace for the campaign B.

- The height of the variance peaks at the clock rising edges is increasing with that of the average peaks; This confirms that this variance actually models the algorithmic noise. As a matter of fact, it is indeed expected that the algorithmic noise is an increasing function of the direct power consumption.

The overall shape of these three campaigns looks quite different, especially in amplitude. Now, if we have a closer look at the synchronization of the campaigns between themselves, we observe that they are not in phase. The Fig. 4(a) typically emphasizes the timing mismatch between A, on the one hand, and B & C on the other hand. This figure is zoomed on the first round of encryption.

3 Template Attack with PCA Preprocessing

Template attacks require to characterize the leakage in an off-line step. This characterization gives all information needed to attack and recover the encryption key. In the usually assumed Gaussian model, the used data consists of averages and covariance of each set of traces categorized according to one model. During the attack phase, the adversary uses the maximum likelihood principle to rank the key hypotheses. Statistically, the more traces, the better the correct key emerges between the others. Ideally, the attacker uses these templates for a successful attack. However, the large size of covariance matrices makes the calculation infeasible in the case of long traces, since matrices are badly conditioned. In practice, points of interest (POIs) must be found to carry out all calculations. Many methods are presented in the literature. Among them we find:

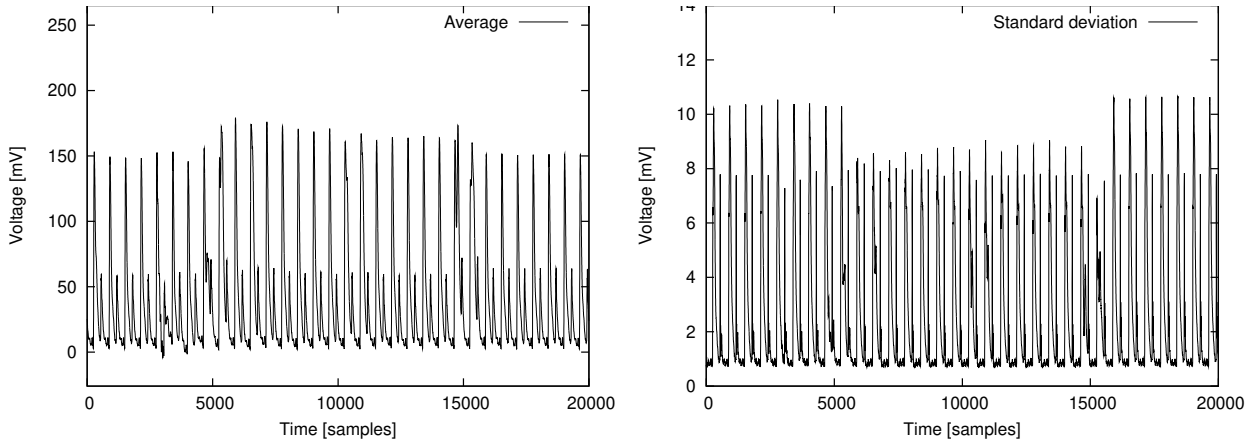
1. Manual selection, which requires expertise;
2. Sum Of Squared pairwise (T-)Differences (or sosd [11] / sost [12]);
3. Linear Discriminant Analysis (LDA [21])
4. Principal Components Analysis (PCA [4]).

To be accurate, the fourth method is a particular case of the so-called principal subspaces template attacks [21]. Indeed, the aim of PCA is to reduce the data to a lower-dimensional representation that summarize a large part of (if not all) the variability. In this article, we compare the templates in PCA subspaces, using only one direction for the projections, which will move from one multivariate analysis to a univariate analysis, while keeping the maximum information. For one or more traces acquired on the target circuit, the *attack phase* consists in guessing the secret key κ used for encryption using Bayes' rule. The attack is successful if and only if:

$$\kappa = \mathrm{argmax}_k \frac{1}{\sqrt{(2\pi)^n |\Lambda_k|}} \exp -\frac{1}{2} \cdot (\tau - \mu_k)^\top \Lambda_k^{-1} (\tau - \mu_k) \,.$$

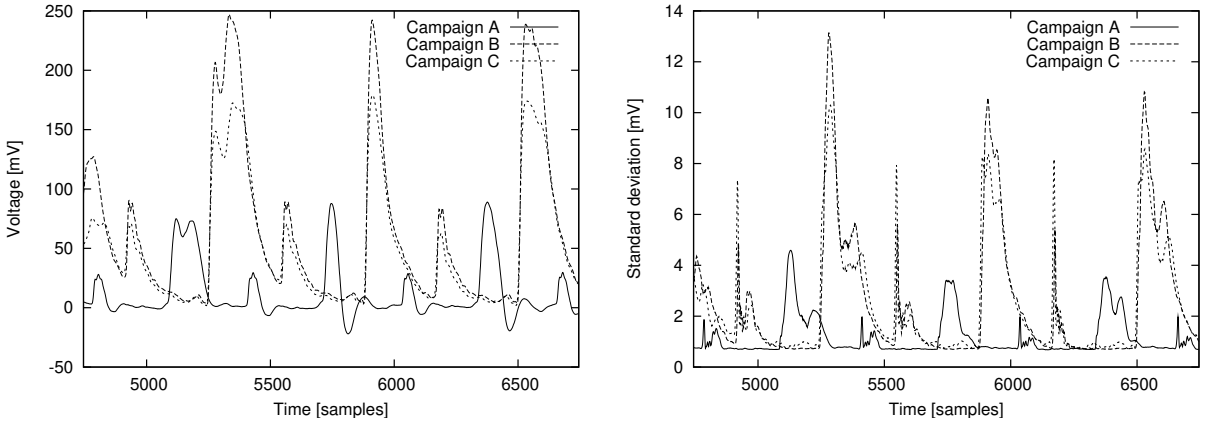
Here, τ is the attacked trace, the pairs (μ_k, Λ_k) are the templates that correspond to the supposed key k, and n is the number of retained directions.

Traces, averages and covariances are *a priori* projected into a new database given by the PCA to reduce dimensions. New directions are the eigenvectors of the covariance matrix constructed from averages representing each template. We choose to work with the first eigenvector as unique direction because it concentrates the maximum of variance: the table 2 shows the large difference between the first and the following eigenvalues for all the campaigns.



(a) Mean trace. (b) Standard deviation.

Figure 3: Mean and standard deviation trace for the campaign C.



(a) Averages. (b) Standard deviations.

Figure 4: Comparison of statistics properties of campaigns A, B and C.

Table 2: Eigenvalues of PCA on campaigns A, B and C.

PCA eigenvalues	**Campaign**		
	A	**B**	**C**
Eigenvalue 1	0.00062368	0.00482484	0.0033602
Eigenvalue 2	9.3984e-06	5.0050e-05	3.3966e-05
Eigenvalue 3	5.3011e-06	3.5375e-05	2.4543e-05
Eigenvalue 4	2.6232e-06	1.3643e-05	1.0195e-05

In this article, we focus on the Hamming distance model on the first round and for the first substitution box, given by $|(\mathrm{R}_0 \oplus (\mathrm{S}(\mathrm{R}_0 \oplus \mathrm{K}_1)) \oplus \mathrm{P}^{-1}(\mathrm{L}_0))[1:4]| \in \{0,1,2,3,4\}$ and detailed in Fig. 5. Consequently we obtain five templates in the profiling phase.

4 Horizontal Resynchronization: POC and AOC

Misalignment of traces is a problem customarily encountered in side-channel analysis. Several methods have been suggested to recover the proper synchronization between curves. Amongst them, we review amplitude-only and phase-only correlations (abridged AOC and POC [16]). Those methods consist in estimating the offset between two traces by maximizing their cross-correlation, or the cross-correlation of their phase. A survey of these methods can be found in [14].

In this section, we compare their efficiency. For this and only for the purpose of comparison, we assume to know the keys and we proceed to a training on the traces attacked using PCA. If the correct partitioning was known (which is true for the templates but not for the traces under attack), we could compare the relative ability of AOC and POC to recover the correct offset by applying them on the first eigenvector of PCA. The result, illustrated in Fig. 6(a), shows that AOC is definitely less noisy. However, without any prior information about the secret key of the campaign to attack, only the average of the campaigns or the traces one by one can be used to estimate the timing offset. The performance is respectively illustrated in Fig. 6(b) and (c). In this case, the POC seems more adequate: the maximal peak has a greater contrast for this method.

We thus use the POC to estimate the misalignment of the curves between campaigns A and B on the one hand and A and C on the other. We end up with a global resynchronization of the curves, depicted in Fig. 7(a) and 7(b). After this time shift, the first eigenvectors are also in phase, as depicted in Fig. 8.

The exact figures for resynchronization are given in Tab. 3; one sample represents 0.05 ns, because the sampling rate is 20 Gsample/s. This table shows that the *a priori* resynchronization on the power traces is slightly different from that using the first eigenvector of the PCA. Nonetheless, we rely in the following on these close values to bring campaigns B and C in synchronization

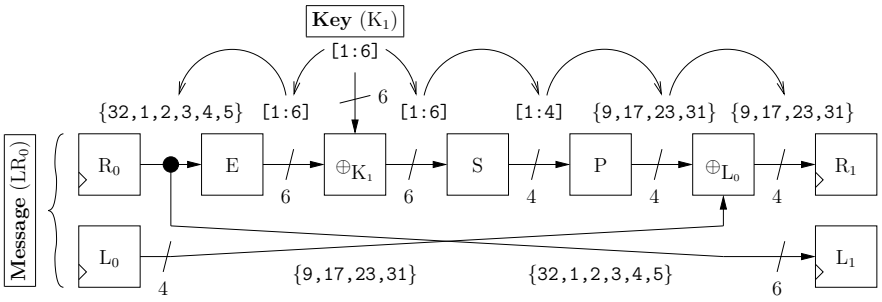


Figure 5: Datapath of DES involved in the attack of the first round.

with campaign A.

5 Successful Attack with Bad Timing Shifts

We tested templates attacks on a large window of offsets: $[0..1,000]$, which includes the offsets of about 170 found previously. Campaign A is used for training with whole set of 80,000 traces, and Campaign B for matching. Currently, we do not use the campaign C, which serves only to see the effect of voltage changing. In the following we focus only on the success rate of the resynchronized campaign B. The success rate on campaign C is similar.

The success rate and the guessing entropy are our comparison metrics. The first-order success rate is, by definition, the percentage of times that the key used during encryption is ranked first among the 64 key assumptions. In most estimates, we have experienced that an increasing number of matching traces generates an increasing success rate. For the sake of representativity, we fix the number of traces to 1,000 to keep enough traces for an accurate estimation of the metrics. The guessing entropy is also important because it illustrates the ranking among the encryption key assumptions. In that sense, this metric is less strict and thus more informative than the success rate about the attack trend.

The purpose of this first experiment is to check whether the success rate can reach 100% exactly at the right shift given by the synchronization characterized in Tab. 3. Basically, we want to know if an attacker has a margin of error in the resynchronization process. The first result is that the offsets offering a decent success rate or guessing entropy are not those actually given by the resynchronization. Indeed, figures 9(a) and 9(b) show that we have different peaks corresponding to different shifts.

On the other hand, using the strict guessing entropy definition [22, Eqn. (2)], we face a practical problem of calculation: the *ex aequo* keys. This problem is explained by the fact that more traces added involve that keys probabilities tend to limit values. Thus, it is possible that at some moments, the probability of some guesses keys becomes zero. This set of guesses keys may include the right key, in the case of traces badly processed. Thereby, the right key will have the same probability than other uncertain guessed keys, and therefore, the guessing entropy will be affected.

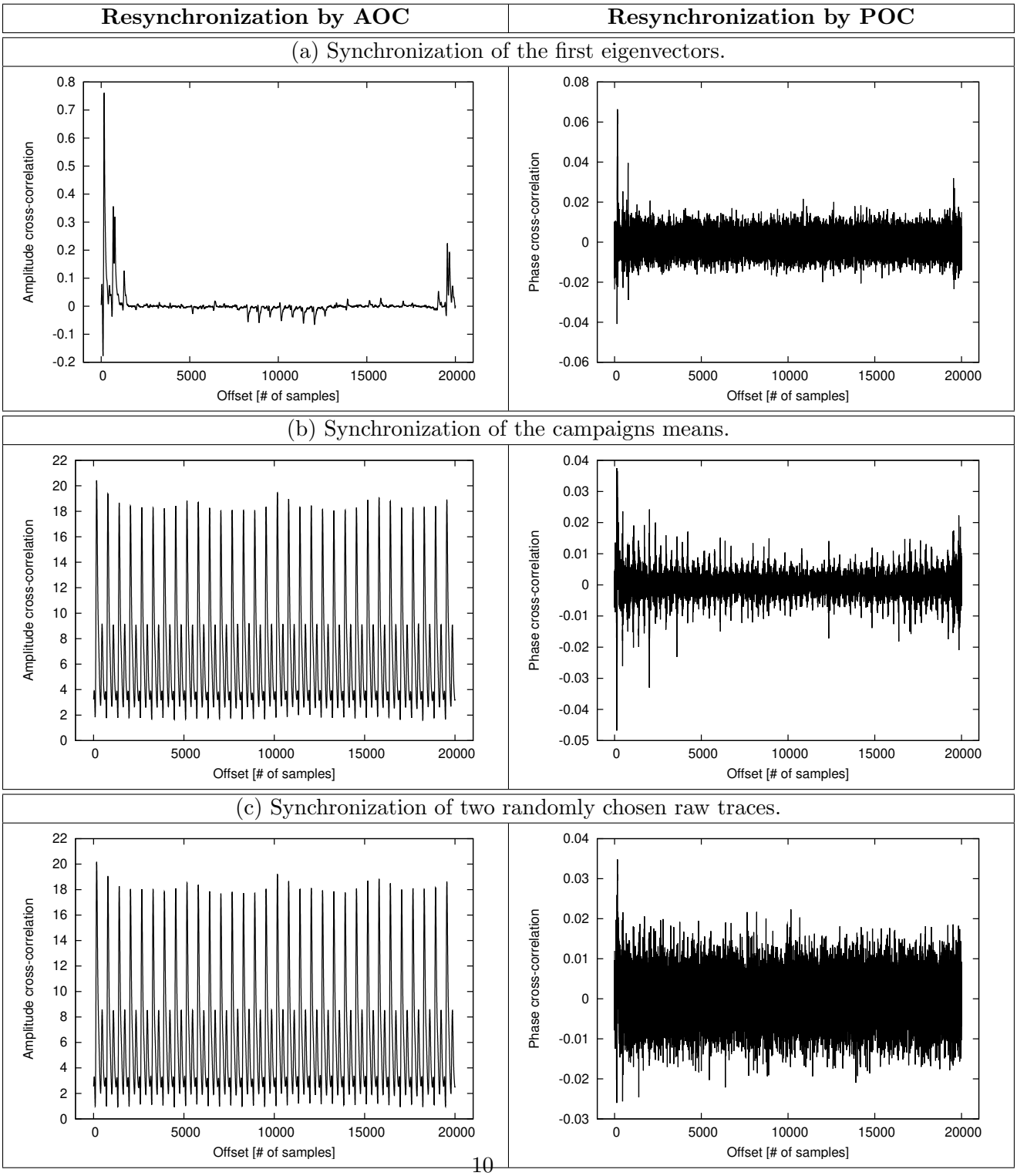
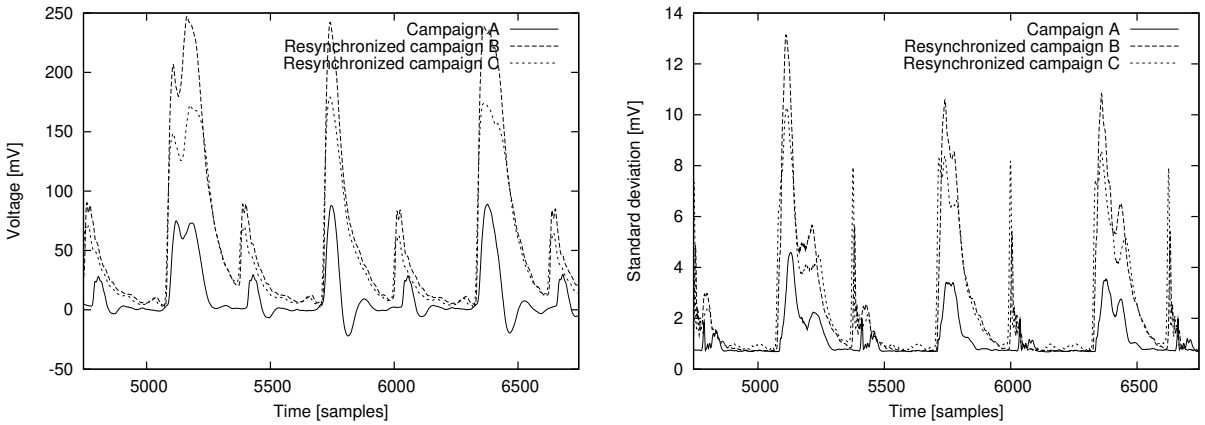


Figure 6: Results of resynchronization between (a) eigenvectors, (b) campaigns means and (c) two samples traces, for AOC and POC.



(a) Averages. (b) Standard deviations.

Figure 7: Comparison of statistical properties of resynchronized campaigns A, B and C.

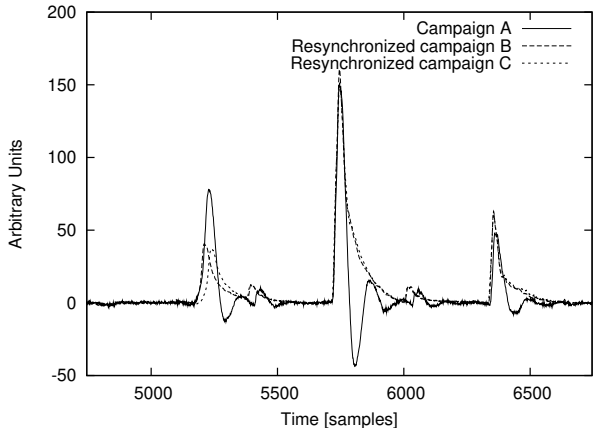


Figure 8: Comparison of resynchronized first eigenvectors of campaigns A, B and C passed through PCA.

Table 3: Optimal time offsets found by POC on campaigns A, B and C, given in sample count units.

Reference for the synchronization	**Campaign**		
	A	**B**	**C**
1st eigenvector (*See Fig. 6(a)*)	0	166	166
Raw traces (*See Fig. 6(c)*)	0	170	172

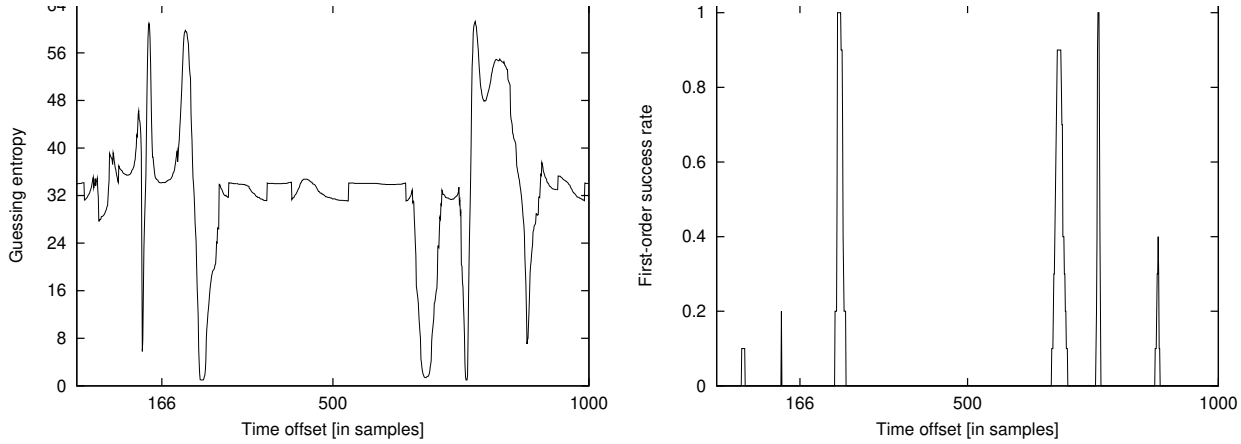
With this in mind, we refine the concept of guessing entropy by adding the notions of *pessimistic* and *optimistic* ranking. The ranking is pessimistic (resp. optimistic) if we consider the worst (resp. best) ranking for *ex aequo* keys. In our figures, the guessing entropy we represent is the average between the pessimistic and the optimistic guessing entropies. Thus, for instance, when the template attack finds the correct key is not the actual one with probability 1 (which can happen due to the finite resolution of the floating point numbers handled by personal computers), then the pessimistic ranking is 64 whereas the optimistic one is 2. Therefore, we opt for a "tradeoff" guessing entropy of $(64 + 2)/2 = 33$.

Figs 9(a) and 9(b) show a similarity between success rate and guessing entropy. At this level, we can deduce that an attacker may recover the key with a high probability, even if she does not synchronize the training and the matching campaigns with the correct offset. Also, the adversary will notice that, for some shifts, the key is ranked last (*i.e.* at position 64 out of 64) in terms of guessing entropy. This phenomenon is repeated at different times. This is due to the fact that the number of time offsets is greater than the clock cycle.

We conjecture these errors are caused by the difference in amplitude between the two campaigns. Actually, the templates built from campaign A do not have the same scale as traces from campaigns B or C; Hence, they match poorly. As a matter of fact, despite our wish to make the acquisition of campaigns B and C close to that of campaign A, we faced amplitudes mismatches. To validate our hypothesis, we take traces from the same campaign A, and split them into halves: one for training and the other for matching. We shift each target trace in a window of 2,000 samples and examine the success rate and the guessing entropy, represented in figures 10(a) and 10(b). We note that the success rate grows to 100% at near zero offset, and remains stuck at zero otherwise. As we can see in Fig. 4(a), the amplitude is significantly different between the averages. Thus, without resynchronization techniques, it would be difficult to recover the key from traces of campaign B or C using the templates built from campaign A.

6 Vertical Homothety

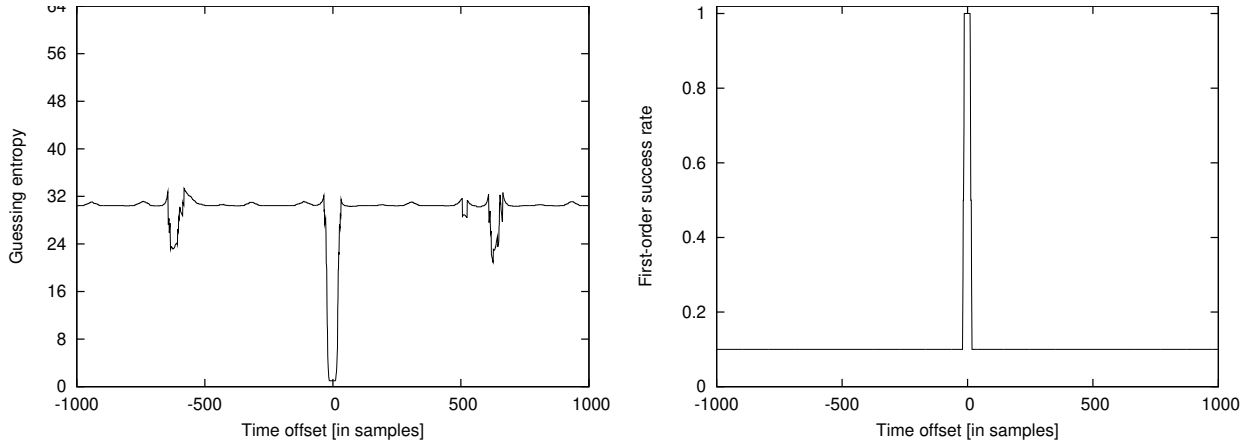
Template attacks are only poorly resilient to homotheties (multiplication by a scalar), therefore the vertical variations can definitely hinder the attack. The effect of an homothety in the voltage is sketched in Fig. 11. This figure describes a case with only two templates, where the attacked



(a) Guessing entropy and different shifts.

(b) Success rate and different shifts.

Figure 9: Looking for the best possible shift for campaign B versus A. We recall from Tab. 3 that the offset found by POC is 166.



(a) Guessing entropy, A *vs* A.

(b) Success rate for the same campaign A.

Figure 10: Success rate and guessing entropy in perfect conditions (campaign A *vs* A). The correct offset is 0.

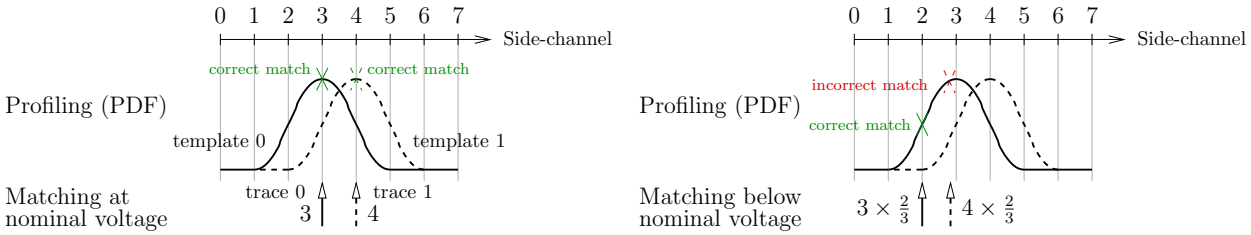


Figure 11: The matching phase can reveal incorrect hypotheses if the observations have gone through an homothety. In this illustration, there are two templates (shown in plain or dashed lines) and the homothety factor is 2/3.

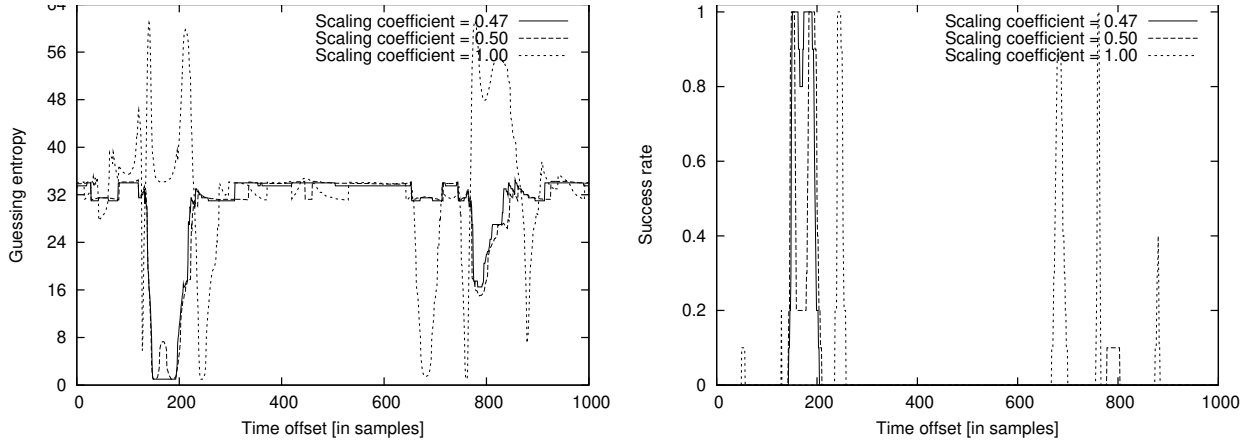
trace has a voltage less than the nominal value. To overcome this problem, we perform a vertical homothety on the matching traces to bring their averages as close as possible to that of the templates. We investigate which scaling factor yields the best results. We begin by multiplying each trace by 0.5, and by calculating the success rate and the guessing entropy for different time shifts. The results, illustrated in Fig. 12(a) and 12(b), show that the shifts for a successful attack are entirely different from those obtained without considering the amplitude. Also we get rid off the surprising sharp peaks of Fig. 9(b) and we observe that the success rate is growing to 100% at the offsets predicted in Sec. 4. We test with another factor (namely 0.47) obtained by approaching the first peak on the trace attacked with the first peak on the general mean training traces. This new factor further improves the guessing entropy and the success rate, especially near the right time shift.

We could work separately on each point and calculate a scaling coefficient per point. However, to automate this procedure, we suggest to center and normalize all traces: those for profiling and also those for matching. This normalization harmonizes the acquisition campaigns and thereby reduces the scaling deviation between them. This method is customarily used in side-channel analysis (see for instance [17, §4.1]): it is further investigated in Sec. 7.

7 Estimation of the Portability

Using the vertical scaling (described in Sec. 6) and the adequate horizontal resynchronization, we compare the efficiency of the templates attacks. The metric is the success rate and the guessing entropy.

We clearly see in Fig. 13(a) and 13(b) that when traces are taken on a setup different from the ones of the templates, the success rate is lower than in the ideal case, where both the matching and training campaigns originate from consecutive measurements. We insist, of course, that the traces for each experiment are well separated: there is indeed no intersection between traces for matching and those of training. Although the attacker could find the key, she nevertheless requires more traces to do so. Actually, the attacker able to globally scale the matching campaign (by a factor of 0.47) does not have the same effectiveness as an attacker who better con-



(a) Guessing entropy *vs* different shifts.

(b) Success rate *vs* different shifts.

Figure 12: Comparison of attacks for different vertical scalings as a function of the resynchronization offset. The training is done on campaign A and the matching on campaign B.

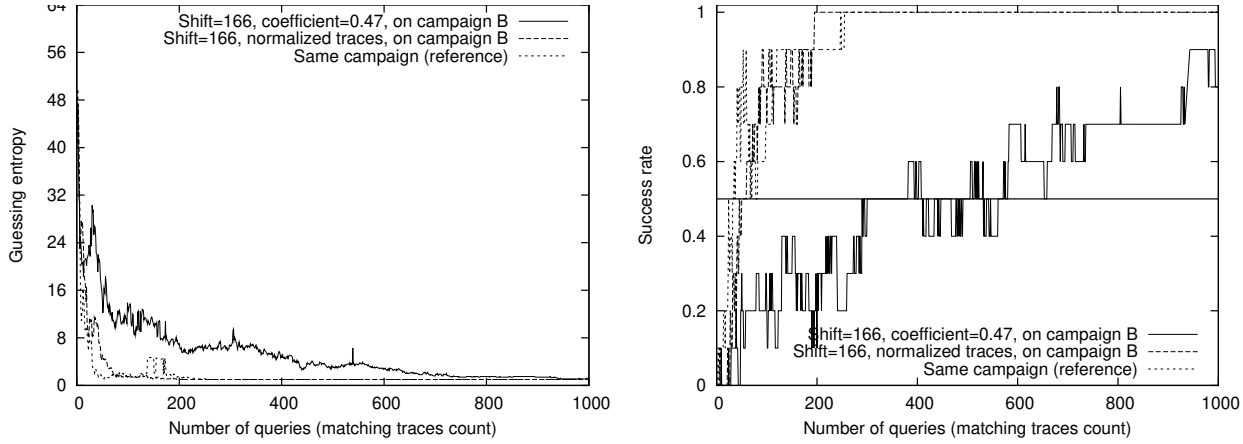
trols the acquisition of traces. The loss can be estimated in terms of number of traces to reach 50% success rate: as can be seen in Fig. 13(b), without precaution, the attack requires about 10 times more queries.

Nonetheless, if both campaigns for templates and for matching are normalized (each trace is replaced by its difference with the average trace of the campaign, and this subtraction is itself divided by the overall campaign standard deviation), we observe (also in Fig. 13(a) and 13(b)) that the metrics are almost as good as for the template attacks on the reference campaign (half of A for training *versus* the other half for matching). Thus, the normalization of campaigns in conjunction with timing resynchronization is a preprocessing that allows for a successful portability of templates from campaign A to B. The the same work can be done on the campaign C. The success rates will be similar to those observed with B. It can therefore be claimed that, according to our experiments, template attacks (with the indicated preprocessing) can indeed be almost 100% efficient even if discrepancies exist in timing or vertical scaling.

It is also interesting to compare template attacks with univariate attacks (typically DPA [6] and CPA [7]). In Fig. 14, it can be seen that attacking a campaign B with raw (hence unadapted) precharacterization from campaign A yield results worse than DPA or CPA, but that template attacks conducted with resynchronization and normalization perform better than those attacks.

8 Conclusion and Perspectives

Despite the favors granted to a template attack adversary, such as unlimited training on a clone device, she can come across difficulties if the traces are neither to scale nor synchronized. Based on real experiments, we indeed note that the vertical amplitude can change between the



(a) Comparing guessing entropy.

(b) Comparing success rate.

Figure 13: Effectiveness of an attacker using preprocessing techniques suitable for template attacks, using campaign A for the templates.

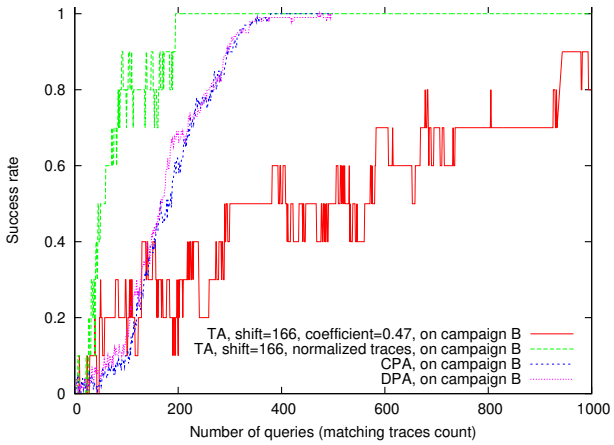


Figure 14: Comparison in terms of success rate of template attacks (raw or with suggested prechar-acterization) and traditional univariate attacks (DPA and CPA).

acquisitions on the clone and the measurements on the targeted circuit.A desynchronization in time might occur, which induces errors especially in the choice of points of interest. We investigate this kind of situation, in the case where an adversary uses PCA to reduce the dimensionality of the side-channel traces. For our case-study, we have made acquisitions at very different dates, and we conclude that despite all our efforts to maintain the same conditions, the traces appearance is not the same.

In this situation, we recommend that the adversary adds a treatment phase between training and the real attacks. We demonstrate how to resynchronize traces in amplitude and time to be able to recover the key. Realignment in time is straightforward; but it is difficult to find a consensual coefficient to change the amplitude of each trace. Thus we introduce the normalization of the traces (both for the templates "training" and for the "matching" traces). This pretreatment appears efficient: it allows to keep a success rate equivalent to an attack that is made on the same acquisition campaign.

References

[1] Aabid, M.A.E., Guilley, S., Hoogvorst, P.: Template Attacks with a Power Model. Cryptology ePrint Archive, Report 2007/443 (2007). http://eprint.iacr.org/2007/443/

[2] Agilent Technologies: Agilent InfiniiVision 5000/6000/7000 Series Oscilloscopes – User's Guide. http://cp.literature.agilent.com/litweb/pdf/54695-97022.pdf

[3] Agrawal, D., Rao, J.R., Rohatgi, P., Schramm, K.: Templates as Master Keys. In: CHES, vol. 3659, pp. 15–29. Springer (2005). Edinburgh, UK

[4] Archambeau, C., Peeters, É., Standaert, F.X., Quisquater, J.J.: Template Attacks in Principal Subspaces. In: CHES, *LNCS*, vol. 4249, pp. 1–14. Springer (2006). Yokohama, Japan

[5] Baba, A.H., Mitra, S.: Testing for Transistor Aging. In: VTS, pp. 215–220 (2009)

[6] Bevan, R., Knudsen, E.: Ways to Enhance Differential Power Analysis. In: ICISC, *Lecture Notes in Computer Science*, vol. 2587, pp. 327–342. Springer (2002). Seoul, Korea

[7] Brier, É., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: CHES, *LNCS*, vol. 3156, pp. 16–29. Springer (2004). Cambridge, MA, USA

[8] Bär, M.: Verbesserung von Template Attacken auf Chipkarten (Improvements of Template Attacks against Smart Cards). Master's thesis, Hochschule Konstanz (2008). (Diploma thesis, in German)

[9] Bär, M., Drexler, H., Pulkus, J.: Improved Template Attacks. In: COSADE, pp. 81–89 (2010). Darmstadt, Germany. http://cosade2010.cased.de/files/proceedings/cosade2010_paper_14.pdf

[10] Chari, S., Rao, J.R., Rohatgi, P.: Template Attacks. In: CHES, *LNCS*, vol. 2523, pp. 13–28. Springer (2002). San Francisco Bay (Redwood City), USA

[11] Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual Information Analysis. In: E. Oswald, P. Rohatgi (eds.) Cryptographic Hardware and Embedded Systems

– CHES 2008, *LNCS*, vol. 5154, pp. 426–442. Springer (2008)

[12] Gierlichs, B., Lemke-Rust, K., Paar, C.: Templates vs. Stochastic Methods. In: CHES, *LNCS*, vol. 4249, pp. 15–29. Springer (2006). Yokohama, Japan

[13] Guilley, S., Hoogvorst, P., Pacalet, R., Schmidt, J.: Improving Side-Channel Attacks by Exploiting Substitution Boxes Properties. In: Presse Universitaire de Rouen et du Havre (ed.) BFCA, pp. 1–25 (2007). May 02–04, Paris, France, `http://www.liafa.jussieu.fr/bfca/books/BFCA07.pdf`

[14] Guilley, S., Khalfallah, K., Lomne, V., Danger, J.L.: Formal Framework for the Evaluation of Waveform Resynchronization Algorithms. In: LNCS (ed.) WISTP: Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing, *LNCS*, vol. 6633, pp. 100–115. Springer (2011). Heraklion, Greece. DOI: 10.1007/978-3-642-21040-2_7

[15] Hanley, N., Tunstall, M., Marnane, W.P.: Unknown Plaintext Template Attacks. In: WISA, *Lecture Notes in Computer Science*, vol. 5932, pp. 148–162. Springer (2009). Busan, Korea

[16] Homma, N., Nagashima, S., Imai, Y., Aoki, T., Satoh, A.: High-Resolution Side-Channel Attack Using Phase-Based Waveform Matching. In: CHES, *LNCS*, vol. 4249, pp. 187–200. Springer (2006). Yokohama, Japan

[17] Prouff, E., Rivain, M., Bevan, R.: Statistical Analysis of Second Order Differential Power Analysis. IEEE Trans. Computers **58**(6), 799–811 (2009)

[18] Rechberger, C., Oswald, E.: Practical Template Attacks. In: WISA, *LNCS*, vol. 3325, pp. 443–457. Springer (2004). Jeju Island, Korea

[19] Renauld, M., Kamel, D., Standaert, F.X., Flandre, D.: Information Theoretic and Security Analysis of a 65-Nanometer DDSLL AES S-Box. In: B. Preneel, T. Takagi (eds.) CHES, *Lecture Notes in Computer Science*, vol. 6917, pp. 223–239. Springer (2011)

[20] Renauld, M., Standaert, F.X., Veyrat-Charvillon, N., Kamel, D., Flandre, D.: A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In: EUROCRYPT, *LNCS*, vol. 6632, pp. 109–128. Springer (2011). Tallinn, Estonia

[21] Standaert, F.X., Archambeau, C.: Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In: CHES, *Lecture Notes in Computer Science*, vol. 5154, pp. 411–425. Springer (2008). Washington, D.C., USA

[22] Standaert, F.X., Malkin, T., Yung, M.: A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In: EUROCRYPT, *LNCS*, vol. 5479, pp. 443–461. Springer (2009). Cologne, Germany