Template Attacks with a Power Model

— Illustration on the Side-Channel Cryptanalysis of an Unprotected DES Crypto-Processor —

Moulay Abdelaziz EL AABID Sylvain GUILLEY
Philippe HOOGVORST

Abstract

This article analyses some properties of the *template attack*. Examples come from attacks against an unprotected ASIC implementation of DES. The principal components analysis (PCA) is used to represent the templates in two dimensions. We give a physical interpretation of the templates PCA eigenvalues and eigenvectors.

We show that the S-boxes are *not* the target of template attacks. We point out that the efficiency of template attacks on unprotected implementations can be unleashed by using a power model. The most suitable power-model happens to be linked to the key schedule. This casts a new light on key schedule requirements for SCA resistance against a "template" attacker.

The results are tailored for DES, because this symmetric block cipher is emblematic and is still promised a long life. Its key schedule is also remarkably simple, with cryptanalytic weaknesses, that paradoxically turn out to be a strength against SCA.

1 Introduction

Up to now most cryptosystems have been designed only with the resistance against theorical cryptanalysis in mind. However passing from a high-level mathematical description to a netlist of gates and then to a silicon chip introduces hidden weaknesses and information leaks, which come from the very structure of the CMOS electronic systems. Side channel attacks (SCAs) are attacks which try to break cryptosystems by exploiting these informations leaked by hardware implementations.

Logical gates such as they are currently conceived dissipate energy in a way correlated with the sequence of data treated. The dissipation can be measured in order to obtain information about the secrets of the cryptoprocessor. We call *a trace* the power consumption measured during the operation, expressed as a discrete function of time. Practically it will be represented as a column vector of real number, the index of which is the time and the value is the instantaneous power consumption. If an attack needs more than a single trace all traces will have the same length n and begin at the same time within the cipherment operation.

The simple power analysis (SPA) exploits one *trace* to reconstruct the sequence of operations during the secret computaion and derive information about the secrets from this sequence. However countermeasures exist, which most of the time make the SPA unusable.

The differential power analysis (DPA) [11, 6] is a ciphertext-only attack which uses statistical analysis of the traces obtained during the encipherment of a large number of unknown plaintexts to confirm or infirm a guess about a manageable part of the key. However this attack is not optimal, as it fails to exploit all information available in each sample.

The **template attack** [3, 13, 5, 1, 2] makes a better use of all information present in each sample. It is thus conclusive even when the DPA fails to recover the secret key. It proceeds in two phases:

A training phase, which needs a clone of the attacked system, which contains no secret information and which is in the full control of the attacker. This phase consists in measuring the electrical activity of the clone sysem during a large number of encipherments are performed with random keys and cleartexts. Assuming that the attacked subkey can take K values, say $k \in [0, K[$, this phase ends with K models, $\mathcal{M}_0, \cdots, \mathcal{M}_{K-1}$, of the power dissipation.

An attack phase, which consists in measuring the electrical activity of the secret system during very few encipherments with the secret key and unknow clear- and ciphertexts. The attacker will then try to match the obtained measurements of power dissipation with one of the K models.

The rest of the paper is organized as follows. In section 2, the template attack in principal sub-spaces is sketched. In section 3, the ASIC that is attacked is thoroughly described. Its features will make it possible to explain our new attacks. Next, in section 4, we interpret the eigen-elements of a "classical" template attack, and we show that template attacks do not target sboxes, but the key schedule. In section 5, we explain the link between the physical dissipation and the template attack success. In particular, we introduce a new templates creation technique, based on a power model. Finally, the section 6 concludes the paper and presents further research perspectives related to template attacks.

2 Template Attack

During the *training phase* the attacker gathers a large number of traces, corresponding to random values of the plaintexts of the key. As the clone system is in the full control of the attacker, this number is limited only by time and available storage. The observed traces are then classified according to the value κ of the key by a function ϕ of the key space into the natural numbers. The function used in the litterature is the "identity" function from part of the key to its binary value. It will be designated as ϕ_0 in the sequel.

A trace is considered the realisation of a multivariate gaussian random variable in $\mathbb{R}^N$. For each set $\mathcal{S}_k, k \in [0, K[$ the attacker computes the average μ_k and the covariance matrix Σ_k. These are estimated by:

$$\mu_k = \frac{1}{|\mathcal{S}_k|} \sum_{t \in \mathcal{S}_k} t \qquad \text{and} \qquad \Sigma_k = \frac{1}{|\mathcal{S}_k| - 1} \sum_{t \in \mathcal{S}_k} (t - \mu_k)(t - \mu_k)^T . \tag{1}$$

The couple (μ_k, Σ_k) is called *the template associated with value k of the subkey.*

The difficult part of this attack is the size of the Σ_k matrices, namely $(N \times N)$, with $N \approx 10^4$. In our experiments, $N = 20,000$. To overcome this a few special indices can be chosen in $[0, N[$, called *interest points*, which contain most of the useful information. Various techniques are used to select these points: points whith large differences between the average traces [3], points with maximal variance [5], and, more recently, principal component analysis (*a.k.a* PCA [10].)

The PCA attack constists in computing the eigenvectors (so-called "*principal directions*") of empirical covariance matrix of all traces together, computed by:

$$S = \frac{1}{K-1} \sum_{k=0}^{K-1} T T^T \quad \text{where} \quad \bar{\mu} = \frac{1}{K} \sum_{k=0}^{K-1} \mu_k \quad \text{and} \quad T = \begin{pmatrix} \mu_1^T - \bar{\mu}^T \\ \mu_2^T - \bar{\mu}^T \\ \vdots \\ \mu_N^T - \bar{\mu}^T \end{pmatrix} .$$

Though S is a $(N \times N)$ matrix, it is the sum of K matrices with rank 1. Its rank is thus at most K. It can be shown that the matrix $S' = \frac{1}{K-1} \sum_{k=0}^{K-1} T^T T$, which is a $(K \times K)$ matrix, has

the same non-zero eigenvalues λ_k as S, thus leading to compute the eigenvalues of a $(K \times K)$ matrix much smaller than a $(N \times N)$ one. It is customary — although certainly not optimal, as discussed later on in section 4 — to choose $K = 64$ for DES and $K = 256$ for AES.

The eigenvectors of S are the columns of the matrix TU, where U is the matrix of eigenvectors of S', ordered by decreasing values of the eigenvalues. The columns of TU are orthogonal thus one only has to normalize them to obtain an orthonormal basis of $\mathbb{R}^N$ consisting of eigenvectors of S.

The means traces and covariance matrices defined in (1) are then expressed in this basis by:

$$\nu_k = (TU)^T \mu_k \qquad \text{and} \qquad \Lambda_k = (TU)^T \Sigma_k (TU)\,.$$

Together with the (TU) matrix, they constitute the templates of the PCA.

The attack phase consists then in acquiring the trace τ of an encipherement performed by the target system using the secret key κ, projecting it into this latter basis and to match it against the patterns using Baye's rule:

$$\kappa = \text{argmax}_k \left(\frac{1}{\sqrt{(2\pi)^N |\Lambda_k|}} \exp\left(-\frac{1}{2} \cdot (TU(\tau - \mu_k))^T \Lambda_k^{-1} (TU(\tau - \mu_k)) \right) \right) .$$

3 Experimental Setup

We endeavour to demonstrate that the considerations developed later on in Sec. 5.1 are practical. In addition, we also wish to relate the attack results to a DES cryptoprocessor architecture.

For these reasons we have actually performed the templates attacks on a real ASIC, called SecMat [4, pp. 62–63]. SecMat is a 0.13 μm prototype academic circuit, designed for cryptographic attacks evaluation. It contains several co-processors, controlled by a microprocessor.

The encryptions are delegated to a DES co-processor. This secular encryption standard has been chosen because it is still used in many smartcard secured protocols. Moreover, DES will be around for a long time.

In the SecMat system-on-chip, the DES co-processor has it own power supply lines, which allows us to capture its power consumption alone, without any noisy activity from the rest of the system. This situation is in favor of the attack. However it allows us to analyze template attacks in an ideal situation and to have an unbiased insight into the side-channel analysis.

The DES co-processor has the multi-mode with triple encryption capability architecture described in [8]. The RTL architecture is summarized below:

- a 64-bit register ("IF") is used for the load/unload operators from/to the 8-bit memory.
- a 64-bit register ("LR") holds the round messages.
- a 56-bit register ("CD") holds the round keys.

The encryption is performed iteratively: one round is computed simultaneously for the message M and the key K in one clock cycle. The control is dictated by the following data-independent state machine:

Clock cycles $1 \mapsto 7$	: K_1 to K_7 loaded into IF, one byte at a time.
Clock cycle 8	: K_8 loaded into IF ; CD := IF.
Clock cycles $8 \mapsto 15$	: M_1 to M_7 loaded into IF, one byte at a time.
Clock cycle 16	: M_8 loaded into IF ; LR := IF.
Clock cycles $16 \mapsto 31$	: DES rounds 1 to 15.
Clock cycle 32	: DES round 16 ; IF := LR.
Clock cycles $32 \mapsto 40$	: (IF) is written into memory, one byte at a time.

For the comprehension of the rest of the article, it is relevant to insulate the key activity:

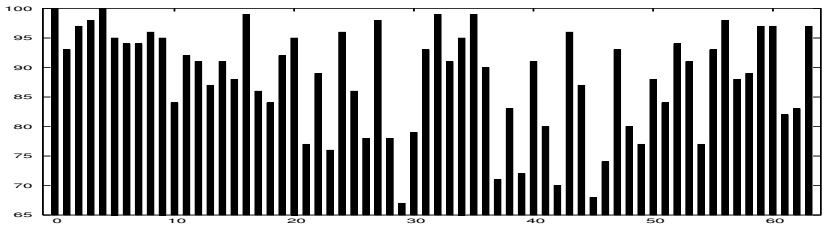

Figure 1: Results of template attacks with 32 principal components.

Clock cycles $1 \mapsto 8$	: The key is loaded in IF byte by byte.
Clock cycles $8 \mapsto 16$	: The key is progressively erased from IF by the incoming message.
Clock cycles $16 \mapsto 32$	: The key schedule (LS or LS^2 transfers in CD) is activated.
Clock cycles $32 \mapsto 40$	: The key is untouched during this phase, hence no activity.

An acquisition campaign consists in the recording of power traces for many {key, message} couples. As described in [7], the acquisition apparatus is an INFINIIUM 54 855A oscilloscope from AGILENT. The probes' model is 1132A, featuring a bandwidth of 5 GHz. The E2669A differential connectivity kit was used. The power traces shown in this article were acquired with a solder-in connector. Every trace is averaged 64 times by the oscilloscope to filter out the environmental noise and to increase the vertical resolution from 8 to 12 bits.

4 Template Attacks Target the Key Schedule Only

Two traces acquisition campaigns were performed:

Campaign #1: the six bits of the key entering S-box 1 at first round were random, all other being zero.

Campaign #2: cleartext & key were fully random.

By carring out these attacks, we could obtain many results which us allowed to understand this attack. In the first step we remade in the details the same preparation as already presented in literature (*e.g.* in [2].) Campaign #1 enabled us to break the key corresponding to a single trace taken on the target device with very good probability. Fig. 1 shows the success rate of an attack using 32 components only with 1 trace from the attacked device. The "x" axis shows the actual key while the ordinates shows the probability of success of the attack in the range [65 %:100 %]. The probability approaches very quickly to 100 % by adding more traces correspondent to the same key.

Campaign #2 was then performed to get a better understanding of the eigenvectors. This time all 56 key bits were random. However we could not break the target traces with good probabilities.

To understand the reasons of this failure, we analyzed the distribution of the probability density for each of the 64 keys. We noticed that those of campaign #1 were a lot more distinct as those from campaign #2. Fig. 2 and Fig. 3 show the probability density functions (PDFs) associated with each campaign. For an explanation of the color code, refer to appendix A at page 14.

The eigenvector associated with the greatest eigenvalue obtained in campaign #1 is depicted on Fig. 4 while the same eigenvector obtained in campaign #2 is depicted on Fig. 5. Both figures show the times when energy is dissipated while the key is loaded and used. The fact that the energy dissipated at each round contributes to the eigenvector shows that our attack did not target the S-box 1 of round 1 only but something permanent during all the encipherment. As

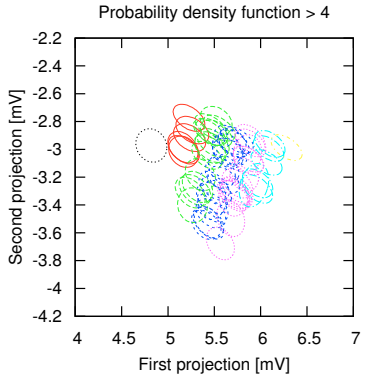

Figure 2: Probability density functions for the acquisition campaign #1.

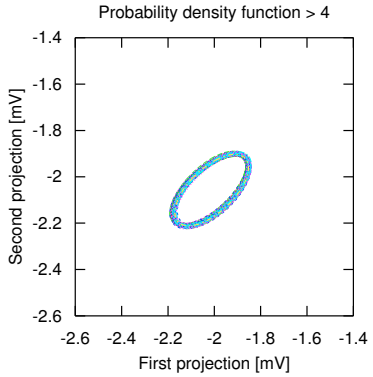

Figure 3: Probability density functions for the acquisition campaign #2.

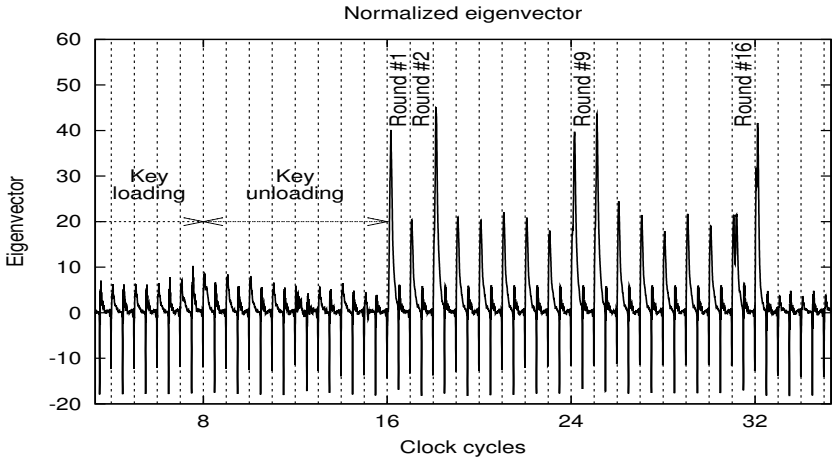

Figure 4: First eigenvector in campaign #1 with "identity"-based classification.

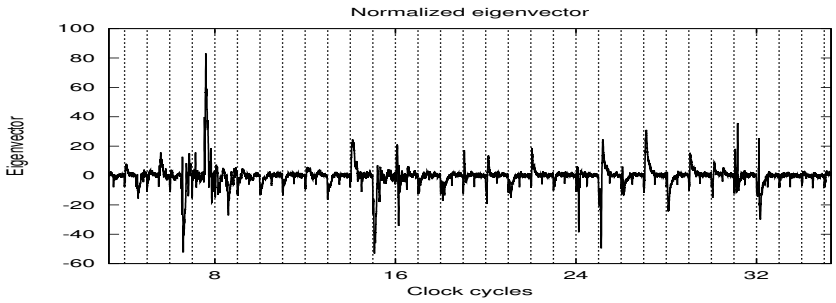

Figure 5: First eigenvector for the acquisition campaign #2 with "identity"-based classification.

the input messages were random, the data path did not contribute to the average consumption. Thus we concluded that the attacked part of the DES processor was its key schedule logic.

In PCA, the first eigenvector indicate the intensity of the side-channel leakage as a function of the time. In campaign #1, where the attack is successful, the eigenvector (Fig. 4) is perfectly correlated to the key activity. In campaign #2, the first eigenvector is decorrated with the key activity, which is a hint why the attack fails. The tools introduced in the next section 5 explain these facts.

5 Template Attacks with a Power Model

5.1 New Attacker Model

Knowing that substitution boxes are not the target of template attacks, the attacker may change her strategy. This leads us to introduce a new attacker model.

The attacker wants to retrieve some information on a secret. She insulates part of the secret, on which an exhaustive search is possible. Then she builds classes that represent the part of a secret to be extracted. However the way the classes are built is no longer a trivial "identity" mapping between the guessed value of the sub-key but a more complex function of this sub-key.

Actually, when neither the algorithm nor the device internals are known, the "identity"-based classification is the most natural as it reflects the fact that the attacker has to choose classes "in blind". This adversarial strategy is discussed in the following sub-section 5.2.

5.2 Template Attacks with a Hamming Weight Model

A direct application of the "identity" mapping from $\mathbb{F}_2^6$ to itself for the classification leads to unsuccessful attacks. The templates are examined to understand the underlying reason for this failure. Not surprising, it appears that the constructed templates are very close one from each other, as shown in Fig. 3. The $K = 2^6$ eigenvalues are plotted in Fig. 6. The number of "representative" eigenvalues, defined as the minimal number of eigenvalues that make up 85 % of the total variance, $K_{\text{representative}} \doteq \min \left\{ k \in]0, 2^6[\text{ s.t. } \sum_{l=0}^{k} \lambda_l \geq 0.85 \times \sum_{l=0}^{2^6-1} \lambda_l \right\}$, is 20.

At first glance, it may seem counter-intuitive that the attack described in section 4 works but fails when the key has its full entropy. As already mentioned in Sec. 1, we performed two templates constructions:
Campaign #1: the key is all zeroes, but the six attacked bits and
Campaign #2: the key is random.

In the campaign #1, the classification function is $\boxed{\phi_0 \doteq Id}$. But given the sparsity of the key, the transitions count is proportional to the Hamming weight of the key. This leads to

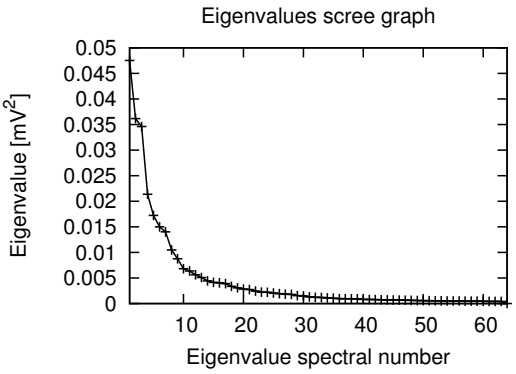

Figure 6: The 2^6 eigenvalues for the acquisition campaign #2 with ϕ_0.

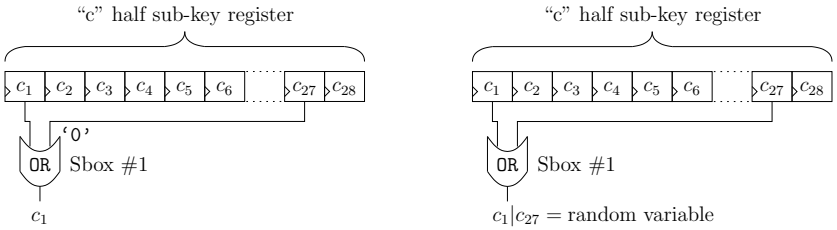

Figure 7: Consistent (left) and noisy (right) combinatorial logic driven by the "C" register.

the paradoxical situation in which the Hamming distance (w.r.t. to a constant 'zero' previous state) degenerates into a Hamming weight. This is illustrated in Tab. 2.

In addition, the dissipation spreads much further than the round key register CD. This is illustrated in the upper part of the Fig. 7, and will be discussed more thoroughly in the next sub-section 5.3.

5.3 Template Attacks with a Hamming Distance Model

In the previous section, an unknown power model was assumed. Nevertheless, in some situations, the power model can be inferred. A dissipation analysis of the cipher under attack will bring us one step further the state-of-the-art. Our goal is to increase the largest eigenvalue of the templates and the total variance. Our motivation is that eigenvalues reflect the distance between the template PDFs. Now, a template attack will succeed better if the PDFs are far one from each other.

In an *open-source* ASIC, such as SecMat, the power model is indeed well known. As first order, it is proportional to the number of commutations of the gates.

The energy dissipated by a portion of the hardware is thus measured by a Hamming distance.

However the best way to distinguish the templates is to build them in such a way that their dissipations differ a lot from each other. As, at first order, the dissipation is correlated to the Hamming distance, the best mapping is the Hamming distance between two successive values of the key in CD. Thus two classification functions are of real interest:

$\boxed{\phi_1 : k \mapsto k \oplus \mathrm{LS}(k)}$ and

$\boxed{\phi_2 : k \mapsto k \oplus \mathrm{LS}^2(k)}$, where LS denotes the DES "circular left shift" operation.

Given our architecture, we use ϕ_1. The templates classification is expected to be as good as the one were the uninteresting key bits are stuck at zero. However, the driven combinatorial

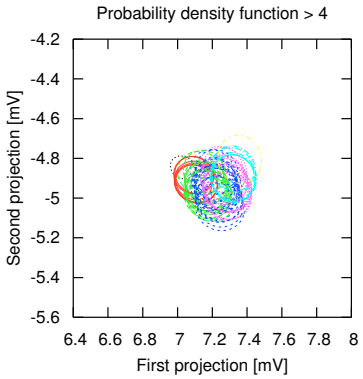

Figure 8: Probability density functions for the acquisition campaign #2 with ϕ_1.

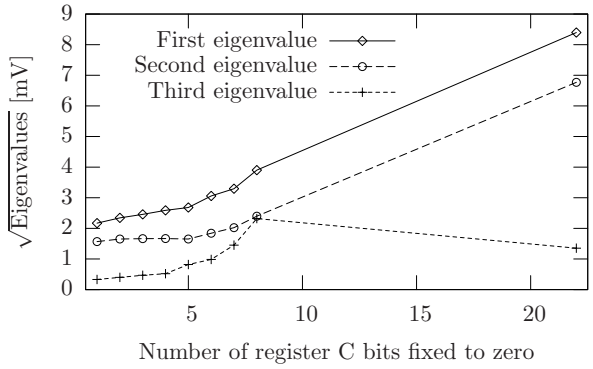

Figure 9: Three main eigenvalues when 1, 2, ..., 8 bits are forced to zero in the key.

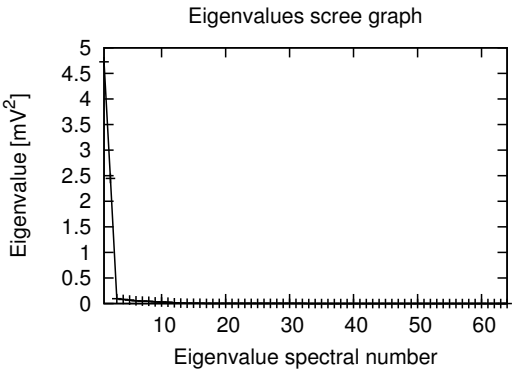

Figure 10: The 2^6 eigenvalues for the acquisition campaign #2 with ϕ_1.

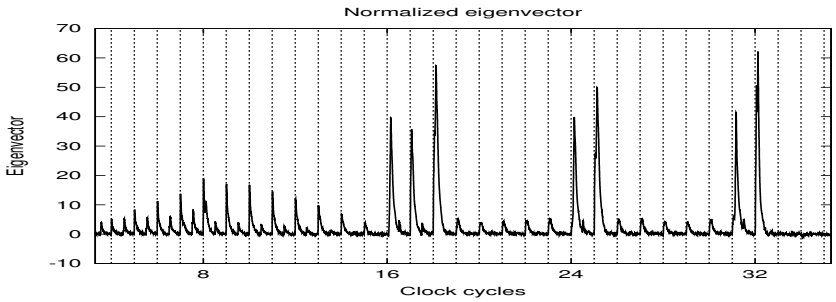

Figure 11: First eigenvector for the acquisition campaign #2 with ϕ_1.

logic (substitution boxes, *etc.*) is not activated coherently. This phenomenon, customarily called "algorithmic noise", is depicted in the lower part of the Fig. 7. In this figure, only one gate of the S-Box is represented. However, many of them participate to the coherent dissipation with CD. Our estimation is that:

- $\sqrt{70.50} = 8.4$ mV of active logic is collected in campaign #1, whereas
- only $\sqrt{4.71} = 2.2$ mV of useful key activity is extracted in campaign #2.

The square root of the eigenvalues is indeed an extensive value, proportional to the quantity of logic that consumes power. From the previous figures, we conclude that the S-Box activity is $(8.4 - 2.2)/2.2 = 2.9$ times higher that the CD register alone. The propagation in the S-Box logic (at *every* round, not only the *first* one) is thus very deep and accounts for the template attack on campaign #1 success.

This hypothesis is corroborated by the fact that the eigenvalues increase when one traces with certain key bits (1 or more amongst the set {9, 1, 58, 50, 42, 34, 26, 18}) are chosen to build the templates. The result is shown in Fig. 9. A complete explanation for the 'trend' would require more measures, because when 8 bits are forced to zero, there remains about only 11 traces per template in campaign #2!

The eigenvalues become significant. We thus have a better distinguisher. The eigenvectors indicate the importance of the leakage, as a function of the time. Given an eigenvalue λ and an associated eigenvector V, for any $\alpha \neq 0$, αV is also an eigenvector associated with λ.

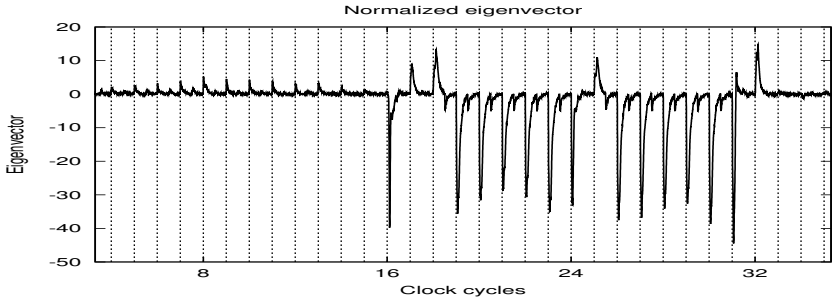

Figure 12: Second eigenvector for the acquisition campaign #2 with ϕ_1.

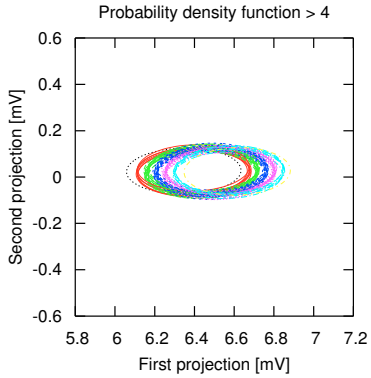

Figure 13: Probability density functions for the acquisition campaign #2 with ϕ_2.

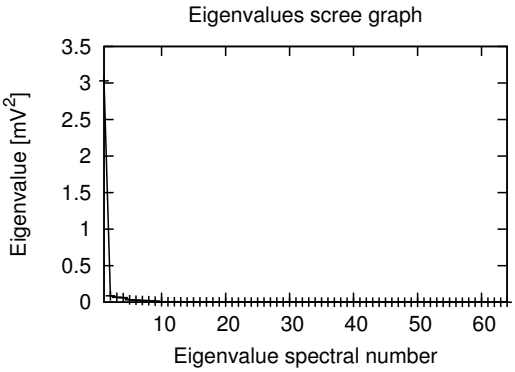

Figure 14: The 2^6 eigenvalues for the acquisition campaign #2 with ϕ_2.

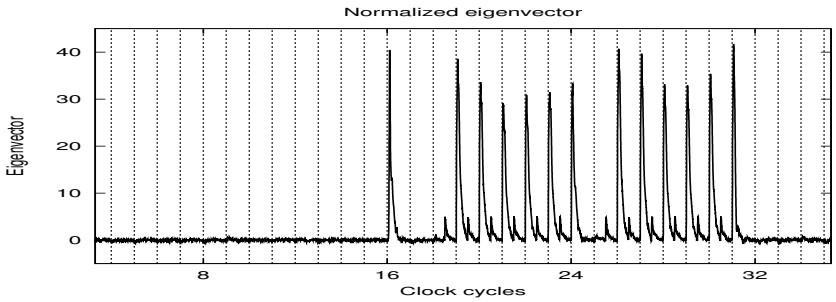

Figure 15: First eigenvector for the acquisition campaign #2 with ϕ_2.

The normalization imposes $(\alpha V)^T(\alpha V) = \alpha^2 = 1$, hence $\alpha = \pm 1$. We say that the power model is physical when all coordinates of the eigenvector have the same sign (either positive or negative). The eigenvectors thus indicates the proportion of the leakage as a function of time. We observe a perfect correlation between the algorithm execution and the eigenvectors. This is the first interpretation in the open literature of the PCA.

Notice that the template built with ϕ_1 as the classification function has two representative eigenvalues. The first eigenvector (Fig. 11) accounts for the LS activity in the key schedule. The second eigenvector (Fig. 12) actually grabs the dissipation from the double bit shifts LS^2. This operation is indeed realized by first shifting by one bit (there is a multiplexer dedicated to this operation) and second by shifting by another bit (a dedicated multiplexor also cares for this operation). These two operations, occurring very close one from each other in time, actually behave as two LS shifts, and thus dissipate $\phi_1 \circ \phi_1 = \phi_1^2$. Given that:

$$\begin{aligned} \phi_1^2(c) &= c \oplus \mathrm{LS}(c) \oplus \mathrm{LS}(c \oplus \mathrm{LS}(c)) \\ &= c \oplus \mathrm{LS}(c) \oplus \mathrm{LS}(c) \oplus \mathrm{LS}^2(c) \qquad \text{// By linearity of LS} \\ &= c \oplus \mathrm{LS}^2(c) \;=\; \phi_2(c)\,, \end{aligned}$$

it is not surprising to find that the second eigenvector in PCA with the classification function ϕ_1 (Fig. 12) is similar to the first eigenvector for ϕ_2 (modulo an arbitrary sign, see Fig. 15).

The template built with ϕ_2 (Fig. 13) is clearly unidirectional, which is in line with the existence of only one representative eigenvalue (Fig. 14).

In Fig. 16, the eigenvectors are compared to the differential traces obtained by the weighting [9] of the campaign #2 acquisition. The figures are very similar, which reinforces the interpretation of the first eigenvector as the principal indicator of the leakages instants.

The table 1 summarizes the eigenvalues obtained in campaign #2 with classification functions ϕ_0, ϕ_1 and ϕ_2. It is clear that the use of the power models ϕ_1 or ϕ_2 increase by several orders of magnitude the variances in the principal subspaces.

6 Conclusion

We have shown that, unlike often believed, the template attacks do not attack substitution boxes. It is sometimes suggested to implement the datapath with normal logic [12], but to increase the effort on substitution boxes. The alleged reason is twofold:

- Substitution boxes are the most dissipative elements of the datapath
- Because of the S-Boxes are non-linear, which helps distinguishing correct key guesses from incorrect guesses.

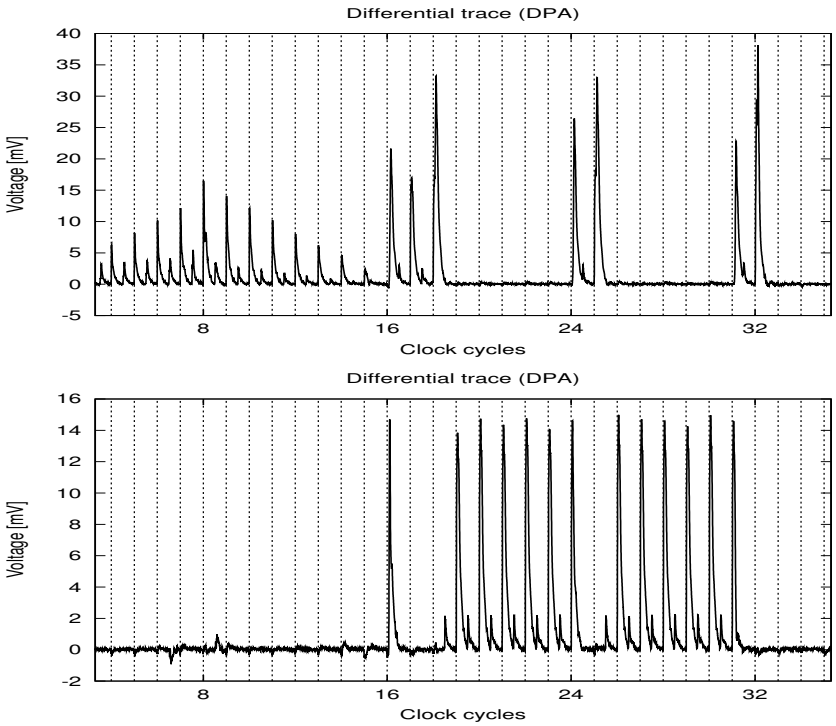

Figure 16: Differential traces obtained by traces weighting with ϕ_1 (upper) and ϕ_2 (lower).

Table 1: Comparison between eigenvalues in campaign #2 for three classification functions.

First eigenvalue		
ϕ_0	ϕ_1	ϕ_2
0.05 mV^2	4.73 mV^2 ($\times$99 w.r.t ϕ_0)	3.03 mV^2 ($\times$64 w.r.t ϕ_0)
Sum of the 2^6 eigenvalues		
ϕ_0	ϕ_1	ϕ_2
0.30 mV^2	7.77 mV^2 ($\times$26 w.r.t ϕ_0)	3.46 mV^2 ($\times$12 w.r.t ϕ_0)

The DPA concentrates on the first or the last round, because it consists in the derivation of a power model for every message and for a key guess. Instead, in template attacks, the message is irrelevant, since the templates are based on averages of the non-fixed algorithm parameters.

The template attacks, in their strongest forms (as described in this article) actually exploit the key schedule. We introduce "template attacks with a power model". They are shown to increase by several orders of magnitude the eigenvalues in the PCA against an ASIC implementation of DES. Future researches will consist in actually exploiting these new templates.

We have described here that the knowledge of the key schedule of an unprotected implementation can help infer the physically relevant power model. However, the automatic retrieval of the adequate power model is still an open issue. Additionally, a technique to combine various power models would be a powerful tool to further improve SCAs.

References

[1] Dakshi Agrawal, Josyula R. Rao, Pankaj Rohatgi, and Kai Schramm. Templates as Master Keys. In *CHES'05*, pages 15–29, 2005.

[2] Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template Attacks in Principal Subspaces. In *CHES'06*, pages 1–14, 2006.

[3] S. Chari, J.R. Rao, and P. Rohatgi. Template Attacks. In *CHES*, volume 2523 of *Lecture Notes in Computer Science*, August 2002. ISBN: 3-540-00409-2.

[4] "Circuits Multi-Projets" (alias CMP, < cmp@imag.fr >) Annual Report 2005. (Online PDF versions: web / local).

[5] Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. Stochastic Methods. In *CHES*, pages 15–29, 2006.

[6] S. Guilley, Ph. Hoogvorst, and R. Pacalet. Differential Power Analysis Model and some Results. In *Proceedings of WCC/CARDIS'04*, pages pp 127–142, August 2004. Toulouse, France.

[7] Sylvain Guilley. *Geometrical Counter-Measures against Side-Channel Attacks*. PhD thesis, ENST / CNRS LTCI, January 2007. `http://pastel.paristech.org/2562/`.

[8] Sylvain Guilley, Philippe Hoogvorst, and Renaud Pacalet. A Fast Pipelined Multi-Mode DES Architecture Operating in IP Representation. *Integration, The VLSI Journal*, 40:479–489, July 2007. DOI: 10.1016/j.vlsi.2006.06.004.

[9] Sylvain Guilley, Philippe Hoogvorst, Renaud Pacalet, and Johannes Schmidt. Improving Side-Channel Attacks by Exploiting Substitution Boxes Properties. pages 1–25. BFCA'07 – `http://www.liafa.jussieu.fr/bfca/`, May 02–04 2007, Paris, France.

[10] Ian T. Jolliffe. *Principal Component Analysis*. 2002. ISBN: 0387954422.

[11] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis: Leaking Secrets. In *Proceedings of CRYPTO'99*, volume 1666 of *LNCS*, pages pp 388–397. Springer, 1999.

[12] Thomas Popp, Mario Kirschbaum, Thomas Zefferer, and Stefan Mangard. Evaluation of the Masked Logic Style MDPL on a Prototype Chip. In *CHES'07*, volume 4727 of *LNCS*, pages 81–94. Springer, September 2007.

[13] Christian Rechberger and Elisabeth Oswald. Practical Template Attacks. In *Lecture Notes in Computer Science*, volume 3325 of *LNCS*, pages 443–457. Springer, august 2004.

A Appendix: Campaign #1 Description

The table 2 describes the acquisition campaign #1. The 64 keys used in this campaign share the property that, at the first round, the input of all the S-Boxes of DES, but the first, have the same input: 0x00. The initial content of the register CD is given, because, as argued in section 4, the template attack success is linked to the activity of CD, not of the S-Box. The Hamming weight of CD, that is constant throughout the DES key schedule, is computed in the last column. It helps see that the 2^6 template PDFs (see Fig. 2) regroup in sub-classes of constant Hamming weight. Actually in the very particular case of the campaign #1, it is a degenerated Hamming distance.

Table 2: Value of the DES key K and of the first round CD register.

Template index	64-bit key K	Register CD initial content	$\lvert CD \rvert$
0	0101010101010101	00000000000000	0
1	0101800101010101	04000000000000	1
2	0101010101018001	40000000000000	1
3	0101800101018001	44000000000000	2
4	0101010101010110	00000080000000	1
5	0101800101010110	04000080000000	2
6	0101010101018010	40000080000000	2
7	0101800101018010	44000080000000	3
8	0101010140010101	00100000000000	1
9	0101800140010101	04100000000000	2
10	0101010140018001	40100000000000	2
11	0101800140018001	44100000000000	3
12	0101010140010110	00100080000000	2
13	0101800140010110	04100080000000	3
14	0101010140018010	40100080000000	3
15	0101800140018010	44100080000000	4
16	0101010101012001	00004000000000	1
17	0101800101012001	04004000000000	2
18	010101010101a101	40004000000000	2
19	010180010101a101	44004000000000	3
20	0101010101012010	00004080000000	2
21	0101800101012010	04004080000000	3
22	010101010101a110	40004080000000	3
23	010180010101a110	44004080000000	4
24	0101010140012001	00104000000000	2
25	0101800140012001	04104000000000	3
26	0101010140001a101	40104000000000	3
27	0101800140001a101	44104000000000	4
28	0101010140012010	00104080000000	3
29	0101800140012010	04104080000000	4
30	0101010140001a110	40104080000000	4
31	0101800140001a110	44104080000000	5
32	0140010101010101	00020000000000	1
33	0140800101010101	04020000000000	2
34	0140010101018001	40020000000000	2
35	0140800101018001	44020000000000	3
36	0140010101010110	00020080000000	2
37	0140800101010110	04020080000000	3
38	0140010101018010	40020080000000	3
39	0140800101018010	44020080000000	4
40	0140010140010101	00120000000000	2
41	0140800140010101	04120000000000	3
42	0140010140018001	40120000000000	3
43	0140800140018001	44120000000000	4
44	0140010140010110	00120080000000	3
45	0140800140010110	04120080000000	4
46	0140010140018010	40120080000000	4
47	0140800140018010	44120080000000	5
48	0140010101012001	00024000000000	2
49	0140800101012001	04024000000000	3
50	014001010101a101	40024000000000	3
51	014080010101a101	44024000000000	4
52	0140010101012010	00024080000000	3
53	0140800101012010	04024080000000	4
54	014001010101a110	40024080000000	4
55	014080010101a110	44024080000000	5
56	0140010140012001	00124000000000	3
57	0140800140012001	04124000000000	4
58	014001014001a101	40124000000000	4
59	014080014001a101	44124000000000	5
60	0140010140012010	00124080000000	4
61	0140800140012010	04124080000000	5
62	014001014001a110	40124080000000	5
63	014080014001a110	44124080000000	6