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 aims at describing a *template attack* against an unprotected ASIC implementation of DES. The goal of the article is to explain the working factors of template attacks. The principal components analysis (PCA) is used to represent the templates in two dimensions. Thanks to this tool, we provide, for the first time in the open literature, with the physical interpretation of the templates PCA eigenvalues and eigenvectors.

This enables us to show that substitution 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 shades 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. It features cryptanalytic weaknesses, that paradoxally turn out to be a strength against SCAs.

1 Introduction

Side channel attacks are very fast methods for break several cryptographic implementaions. They exploit the physical leakeges of material and use these informations in various manners. Ideed, the usual cryptosystems, do not take counts of problems bound to the material implementation. Such problems are reflected for example in the faults (failles) induced by the passage of the description of an algorithm of a high-level language towards its description in term of logical gates. Thus, logical gates such as them are currently conceived dissipate energy, in a way correlated with the sequence of data treated. We can thus measure this disspation in order to deduce certain information on the secrecy of the cryptoprocessor. Accordingly, Simple Power Analysis (SPA) exploits the leakage during the execution part of key schedule, and when the noise in scide channel is not very large, in the direction where information must be clear, to be able to find the key while interpreting the consumption during cryptographic calculations. When these conditions are not met, the SPA is not usable. It is then abandoned with the profit of others techniques such as DPA. Differential Power Analysis [9, 13, 10] uses statictical analysis on large samples which enable reduce the noise by calculating the averages, but this attack is not optimal, because it does not exploit all information present in each sample. The template attacks [3, 12, 5, 1, 2] can circumvent these defects, by exploiting all information present in each sample. It is conclusive when the SPA/DPA not works.

These attacks are the most powerful forms of side channel attacks. It make it possible to break iomplemetations whose security suppose that attacker cannot obtain many samples of the side channel. In order to carry out this attack, an attacker has need for a material programmable identical to the device target, wich will to obtein samples in the form of the traces when performs a K operation sequence $\{O_i\}_{i=1}^{i=K}$. We suppose that we collect T_k traces of a given operation O_k. Thus to model the sample of traces, the multivariate Gaussian density is used, i.e the distribution of a vector whose components are gaussian random variables. This Gaussian density is expressed by:

$$p(t) = \frac{1}{\sqrt{(2\pi)^N |R|}} exp(-\frac{1}{2}(t-\mu)^T R^{-1}(t-\mu)) \tag{1}$$

where for each vector of samples, the average μ and the covariance matrix are estimated by:

$$\mu = \frac{1}{T_k} \sum_{k=1}^{T_k} t_k \tag{2}$$

$$R = \frac{1}{T_{K-1}} \sum_{k=1}^{T_k} (t_k - \mu)(t_k - \mu)^T \tag{3}$$

Note that, the construction of template consists in estimate these two parameters for each operation O_k.

The principal difficulty of this attack is to manage the huge quatity of information or of data, which can be seen like enormous matrix X of dimension (n,p), containing the observations:

$$X = \begin{pmatrix} x_1^1 & \dots & x_1^p \\ \vdots & \vdots & \vdots \\ x_i^1 & x_i^j & x_i^p \\ \vdots & \vdots & \vdots \\ x_n^1 & \dots & x_n^j \end{pmatrix}.$$

Where x_i^j is the value of sample i for the variable j. The lines of this matrix correspond to the traces extracted the circuit.

Some techniques were found to regulate this problem, like the fact of seeking the points where we observe large differences between the average traces [3]. Anathor similar method was presented whish constit in making the sum of squares of the differences between the average traces and choose the points where maximum curve are reached on clock cycle[5].

The base of the use of the PCA [8], constists on calcul of eigenvectors (principal directions) of empirical covariance matrix given by:

$$S = \frac{1}{K-1} \sum_{k=1}^{K} (t_k - \bar{t})(t_k - \bar{t})^T \tag{4}$$

where $\{\mu_k\}_{k=1}^K$ are the empirical mean traces of each operation $\{O_k\}_{k=1}^K$ $m\bar{u} = \frac{1}{K}\sum_{k=1}^K \mu_k$ is the average of the mean traces. we can differently write this matrix of covariance by defining matrix T by:

$$T = \begin{pmatrix} \mu_1^T - \bar{\mu}^T \\ \mu_2^T - \bar{\mu}^T \\ \vdots \\ \mu_N^T - \bar{\mu}^T \end{pmatrix}$$

thus the simplified formula of the covariance matrix:

$$S = \frac{1}{K-1} T^T T \tag{5}$$

the eigenvectors of this matrix are precisely the columns of the matrix TU, where U are the matrix of eigenvectors of $\frac{1}{K-1}TT^T$. Finally, it does not remain any more which has normalise these vectors to obtein an orthonormal basis $(d_1, \dots, d_{K-1})$.

The rest of the paper is organized as follows. In section 2, the ASIC that is attacked is thoroughly described. Its features will make it possible to explain our new attacks. Next, in section 3, 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 4, 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. In section 5, we apply this technique to a realistic attack scenario, that happens to be a bitwise template attack. The security evaluation of this technique is conducted in section 6. Finally, the section 7 concludes the paper and presents further research perspectives related to template attacks.

2 Experimental Setup

We endeavour to demonstrate that the considerations developed later on in Sec. 4.1 are practical. In addition, we also wish to relate the attack results to a DES cryptoprocessor architecture.

For these reasons, we have actually realized 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. In the experiments reported in this paper, the microprocessor is responsible for generating a trigger signal and for launching encryptions.

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. For instance, it has been adopted by the ICAO for symmetric cryptographic operations in the international electronic passport.

In the SecMat system-on-chip, the DES co-processor has it own power supply, which allows us to capture the power consumption of the encryption operation 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 [14]. The RTL architecture is summarized below:

- One interface 64-bit register, called "IF", is responsible for the load/unload operators from/to the 8-bit memory.
- One datapath 64-bit register, called "LR", holds the round messages: IP(cleartext) initially and IP(ciphertext) at the end of the computation.
- One keypath 56-bit register, called "CD", holds the round keys: PC1(LS(key)) initially and PC1(LS^{28}(key)) = PC1(key) at the end of the computation.

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

1. In clock cycles [1–8] . . . . . . The key is loaded in IF byte by byte. At clock cycle 8, the full key is written in parallel in CD, disposing parity bits.
2. In clock cycles [8–16] . . . . The message is loaded in IF byte by byte. At clock cycle 16, the full plaintext is written in parallel in LR.
3. In clock cycles [16–32] . . . The DES algorithm starts. At clock cycle 32, the ciphertext is copied in parallel in IF.
4. In clock cycles [32–40] . . . The ciphertext is serially written in memory, byte by byte.

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

1. In clock cycles [1–8] . . . . . . The key is loaded in IF byte by byte.
2. In clock cycles [8–16] . . . . The key is progressively erased from IF by the incoming random message.
3. In clock cycles [16–32] . . . The key schedule, consisting in LS or LS^2 transfers in CD, is activated.
4. In clock cycles [32–40] . . . The key is untouched during this phase, hence no activity.

An acquisition campaign consists in the recording of power traces for numerous couples {key, message}. As described in [6], 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.

```
End of Sylvain.
Begin of Aziz
```

=¿ Two traces acquisition campaigns:

Campaign #1: Resemble that of Archambeau [2]

Campaign #2: The key fully random

3 Templates do not Target Sboxes, but the Key Schedule

The contributions are:

- Template attacks, as described in the open literature, are stated as the most powerful attacks, since they require no *a priori* knowledge of the target
- They are currently believed to be a "power-model"-independent attack

By carring out all various templates attacks, we could obtain many results which us allowed to understand this attack, and to test it in various situations, in particular with various campaigns of acquisition traces. The first approch consted in remaking in the details the same preparation that already presented in literature. We builds the traces by varying only the six bits of the key at entries of the first sbox, and by supposing all the other bits with zero. This campain whom we name *Compaign*#1 enable us to break the key corresponding to a single trace taken on the target device with very good probabilities. The probabily approches very quickly to 100% by adding more traces correspondents to the same key.

entry first sbox	Key	break success of a one trace
0	0101010101010101	100,0%
1	0101800101010101	93,0%
2	0101010101018001	97,1%
3	0101800101018001	98,7%
4	0101010101010110	100,0%
5	0101800101010110	95,7%
6	0101010101018010	94,6%
7	0101800101018010	94,2%
8	0101010140010101	96,0%

9	0101800140010101	95,3%
10	0101010140018001	84,2%
11	0101800140018001	92,5%
12	0101010140010110	91,2%
13	0101800140010110	87,3%
14	0101010140018010	91,9%
15	0101800140018010	88,9%
16	0101010101012001	99,3%
17	0101800101012001	86,9%
18	010101010101a101	84,3%
19	010180010101a101	92,6%
20	0101010101012010	95,4%
21	0101800101012010	77,0%
22	010101010101a110	89,4%
23	010180010101a110	76,1%
24	0101010140012001	96,1%
25	0101800140012001	86,1%
26	010101014001a101	78,1%
27	010180014001a101	98,6%
28	0101010140012010	78,8%
29	0101800140012010	67,7%
30	010101014001a110	79,4%
31	010180014001a110	93,7%
32	0140010101010101	99,6%
33	0140800101010101	91,5%
34	0140010101018001	95,8%
35	0140800101018001	99,5%
36	0140010101010110	90,6%
37	0140800101010110	71,6%
38	0140010101018010	83,3%
39	0140800101018010	72,8%
40	0140010140010101	91,3%
41	0140800140010101	80,6%
42	0140010140018001	70,8%
43	0140800140018001	96,0%
44	0140010140010110	87,4%
45	0140800140010110	68,9%
46	0140010140018010	74,0%
47	0140800140018010	93,2%
48	0140010101012001	80,2%
49	0140800101012001	77,4%
50	014001010101a101	88,2%
51	014080010101a101	84,0%
52	0140010101012010	94,4%
53	0140800101012010	91,6%
54	014001010101a110	77,5%
55	014080010101a110	93,7%
56	0140010140012001	98,7%
57	0140800140012001	88,7%

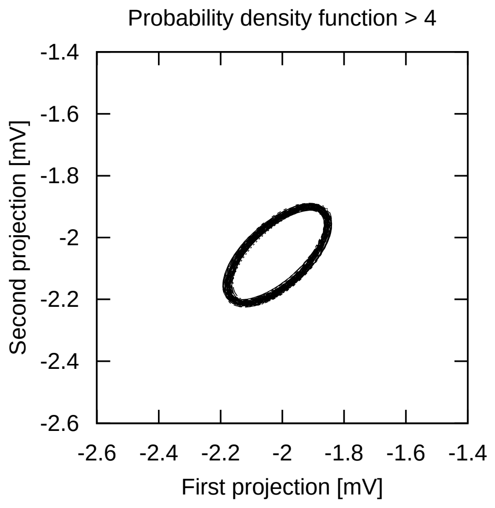


Figure 1: Probability density functions for the acquisition campaign #1.

58	014001014001a101	89,2%
59	014080014001a101	97,1%
60	0140010140012010	97,9%
61	0140800140012010	82,9%
62	014001014001a110	83,4%
63	014080014001a110	97,5%

Table 1: Results of template attacks.

We could also analyze the eigenvector corresponding to this compain. We have to notice that these eigenvectors show the effective moments of dissipation during the loading key and calculation. For better understanding the eigenvectors we made a new acquisition compain where in this time, we take 56 bits random keys. We note this compain $Compaign\#2$. This choise does not prove conclusive since it does not make it possible to break the target traces with good probabilities.

also, we analyze the distribution of the probability density for each of the 64 key. we notices that of compaign #1 are more distinct as those from compaign #2.

The reason of this difference between the two compain is that indeed we does not attack the entry of the sboxes but the part of key-scedule.

We compute the number of representative eigenvalues N as the minimal number of eigenvalues that make up 85 % of the total variance: $N \doteq \min \left\{ n \in]0, 2^6[\text{ such that } \sum_{i=0}^{n} \lambda_i \geq 0.85 \times \sum_{i=0}^{2^6-1} \lambda_i \right\}$.

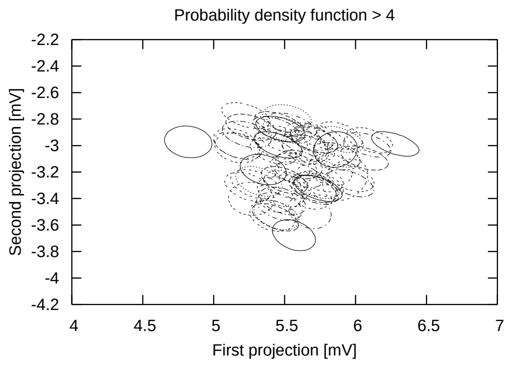


Figure 2: Probability density functions for the acquisition campaign #2.

4 Template Attacks with a Power Model

4.1 Attacker Model

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

The attacker is eager to retrieve some information on a secret. The attack methodology consists in insulating part of the secret, on which an exhaustive search is possible. Then the attacker builds classes that represent the part of a secret to be extracted.

The construction of template classes typically consists in the following steps:

1. collect traces with random messages and keys,
2. classify the traces into 2^6 "templates" (also referred to as "classes" or "populations" [8]). We suppose here that 6 bits of the key are to be retrieved,
3. estimate the mean vectors and the covariance matrices.

So far, in the open literature about template attacks, the second step is always realized trivially. More precisely, the following rationale is *implicitly* applied:

- if the 6 bits to guess equal 00000, then aggregate them into one template, say template #0;
- if the 6 bits to guess equal 00001, then aggregate them into one template, say template #1;
- ...
- if the 6 bits to guess equal 11111, then aggregate them into one template, say template #63;

This is equivalent to using the "identity" mapping (called ϕ_0) from $\mathbb{F}_2^6$ to itself between, where:

- **the input are**: the key bits to guess and
- **the output are**: the template index.

Actually, when neither the algorithm nor the device internals are known, this classification is the most natural. It consists in choosing classes "in blind". This attack strategy is discussed in the following sub-section 4.2.

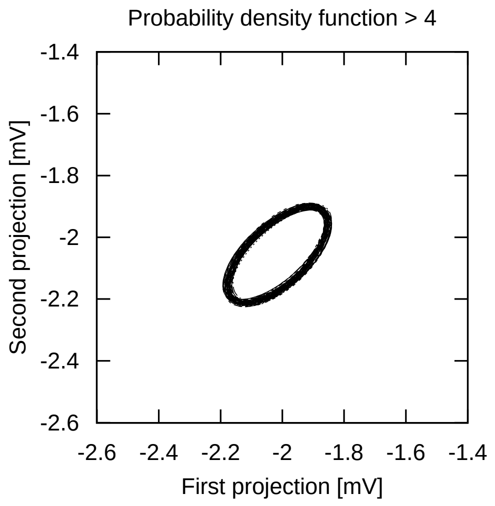


Figure 3: Probability density functions for the acquisition campaign #2 with ϕ_0.

4.2 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 2^6 eigenvalues are plotted in Fig. 4. The number of "representative" eigenvalues is 20. The first eigenvector is shown in Fig. 5. It is clear that this vector is totally decorrelated from the encryption process, as described in section 2.

At first glance, it may seem counter-intuitive that the attack described in section 3 work but fail when the key has its full entropy.

As already mentioned in Sec. 1, we performed two templates constructions:

1. the key is plain zeros, but for the six attacked bits (XXX `07_19` campaign),
2. the key is random (XXX `08_18` campaign).

In the first case, the classification function is $\phi_0 \doteq Id$. But given the sparsity of the key, the transitions count is proportional to the Hamming weight of the key. This leads to the paradoxical situation, where the Hamming distance (w.r.t. to a constant 'zero' previous state) degenerates into a Hamming weight.

This is illustrated in Tab. 2.

For example, in template 63, the CD register initially contains, in hexadecimal, the value `44124080000000`. Consequently,

- At round 0, the binary content of register CD is:
`0100 0100 0001 0010 0100 0000 1000 0000 0000 0000 0000 0000 0000 0000`,
- At round 1, the binary content of register CD is:
`1000 1000 0010 0100 1000 0001 0000 0000 0000 0000 0000 0000 0000 0000`.

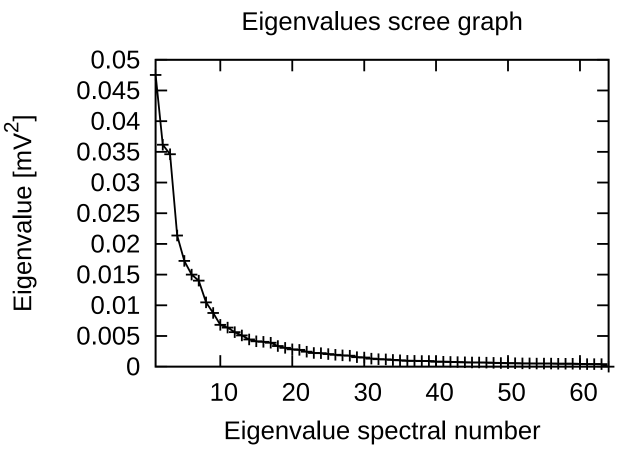


Figure 4: The 2^6 eigenvalues for the acquisition campaign #2 with ϕ_0.

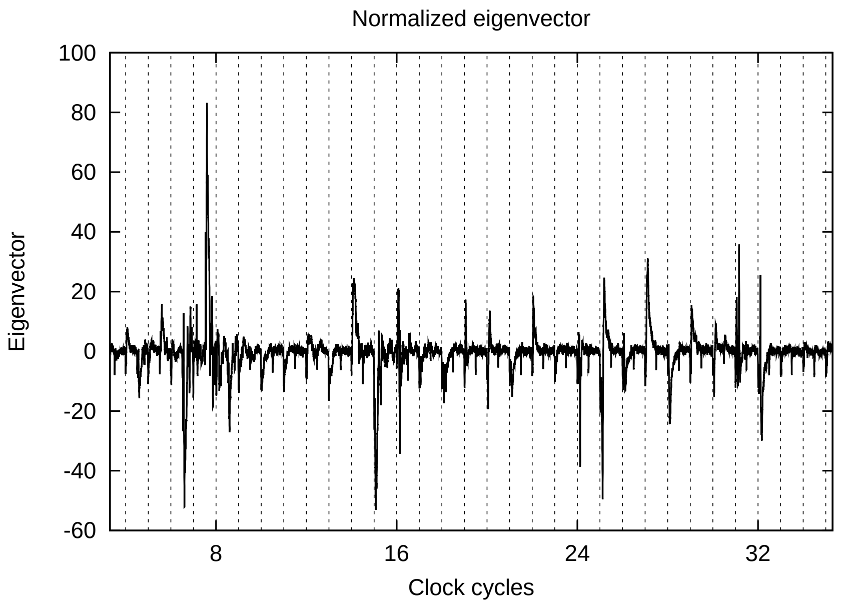


Figure 5: First eigenvector for the acquisition campaign #2 with ϕ_0.

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	010101014001a101	40104000000000	3
27	010180014001a101	44104000000000	4
28	0101010140012010	00104080000000	3
29	0101800140012010	04104080000000	4
30	010101014001a110	40104080000000	4
31	010180014001a110	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

We can observe that the only transitions are caused by `01` pairs of bits being left shifted to `10`. Hence a power signature proportional to the Hamming weight of the key.

In addition, the dissipation spreads much further than the round key register CD. This is illustrated in the upper part of the Fig. 6, and will be discussed more thoroughly in the next sub-section 4.3.

4.3 Template Attacks with a Hamming Distance Model

In the previous section, a 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.

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, given the fact that: the best way to distinguish the templates is to build them in such a way their dissipation differ a lot from one template to another. As at first order, the dissipation is correlated to the Hamming distance, the best mapping is the Hamming distance of the key between two successive values.

In a pipelined implementation of DES, the key schedule consists in left shifts, of multiplicity 1 or 2. Thus two classification functions are of interest:

1. $\phi_1 : c \mapsto c \oplus \mathrm{LS}(c)$ and
2. $\phi_2 : c \mapsto c \oplus \mathrm{LS}^2(c)$,

where LS denotes the DES "circular left shift" operation.

A real classifier that properly maps unknown key bits to their associated dissipation in the key schedule must be used. 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 logic (substitution boxes, *etc.*) is not activated coherently. This is depicted in the lower part of the Fig. 6. In this figure, only one gate of the sbox 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 sbox activity is $(8.4 - 2.2)/2.2 = 2.9$ times higher that the CD register alone. The propagation in the sbox logic is thus very deep and accounts for the template attack on campaign #1 success.

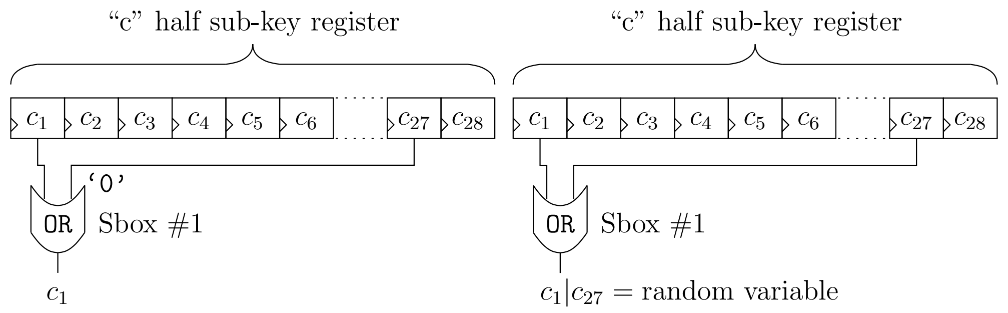
This hypothesis is corroborated by the fact that the eigenvalues increase when one traces with certain key bits (1 or more amongst the set $\{9, 1, 58, 50, 42, 34, 26, 18\}$) are chosen to build the templates. The result is shown in Fig. 7.

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!

In this section, we assume that the device is known: therefore, classes can be built according to a power model (Hamming distance).

The eigenvalues become significant. We thus have a better distinguisher. The eigenvectors indicate the importance of the leakage, as a function of the time. There are actually two possible eigenvectors, namely $+V$ and $-V$, that satisfy simultaneously:

1. $\overline{S}V = \lambda V$ and
2. $V^T V = 1$.



"c" half sub-key register
"c" half sub-key register
c_1 c_2 c_3 c_4 c_5 c_6 c_{27} c_{28}
c_1 c_2 c_3 c_4 c_5 c_6 c_{27} c_{28}
'0'
OR
Sbox #1
OR
Sbox #1
c_1
$c_1|c_{27}$ = random variable

Figure 6: Consistent (up) and noisy (down) combinatorial logic driven by the "C" register.

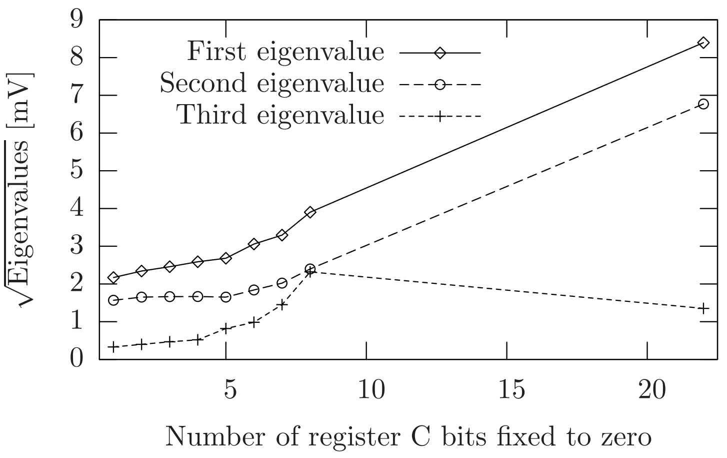


Figure 7: Three main eigenvalues for XXX `08_18` acquisitions where 1, 2, . . . , 8 bits are forced to zero in the key, and for the XXX `07_19` acquisition campaign.

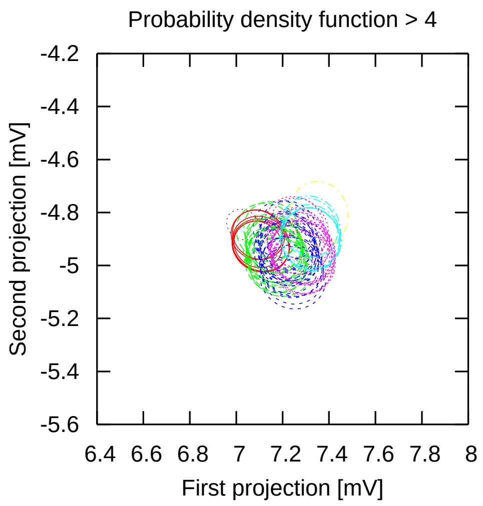


Figure 8: Probability density functions for the acquisition campaign #2 with ϕ_1.

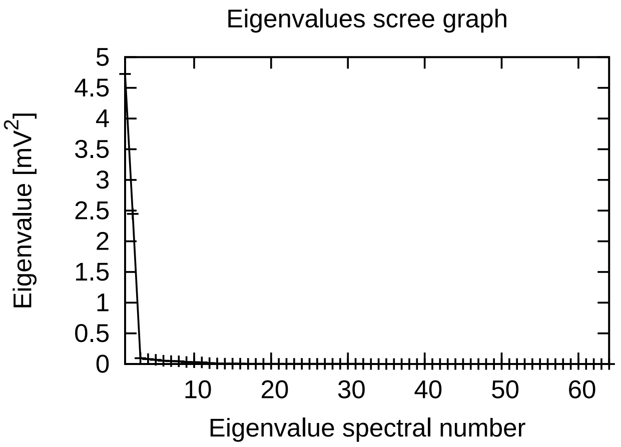


Figure 9: The 2^6 eigenvalues for the acquisition campaign #2 with ϕ_1.

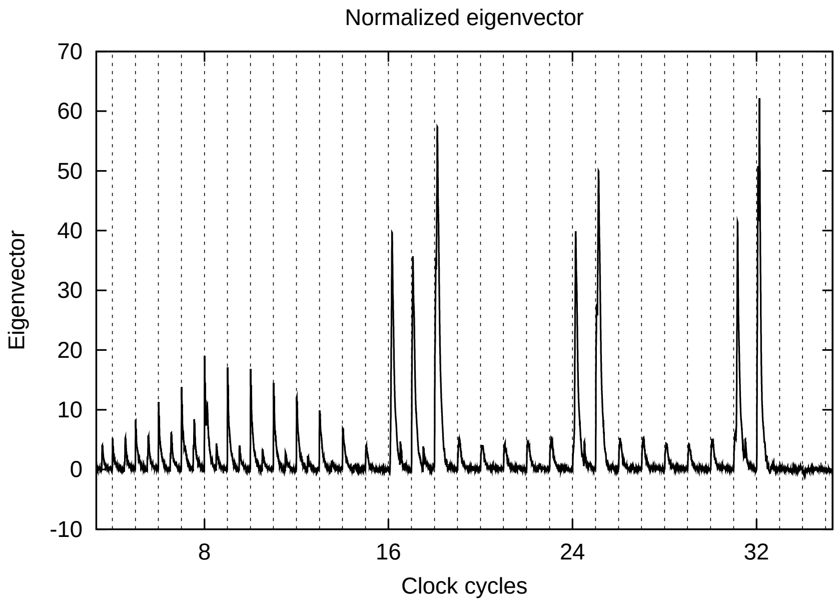


Figure 10: First eigenvector for the acquisition campaign #2 with ϕ_1.

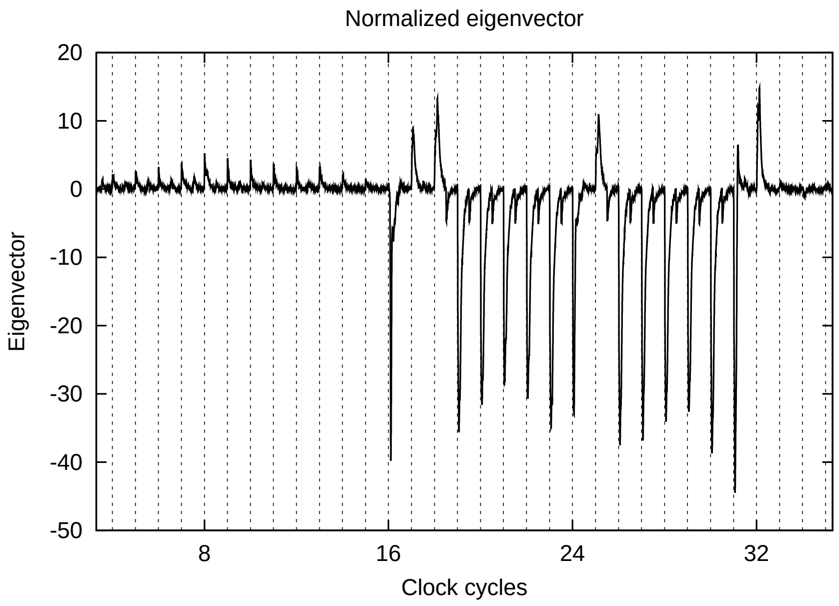


Figure 11: Second eigenvector for the acquisition campaign #2 with ϕ_1.

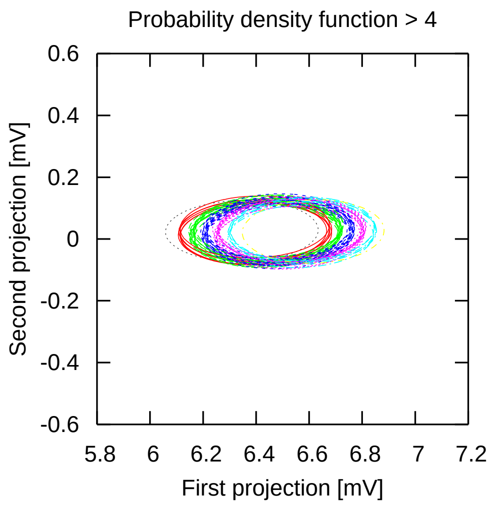


Figure 12: Probability density functions for the acquisition campaign #2 with ϕ_2.

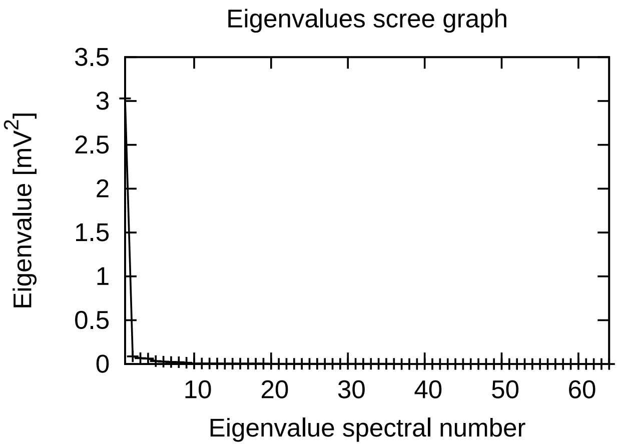


Figure 13: The 2^6 eigenvalues for the acquisition campaign #2 with ϕ_2.

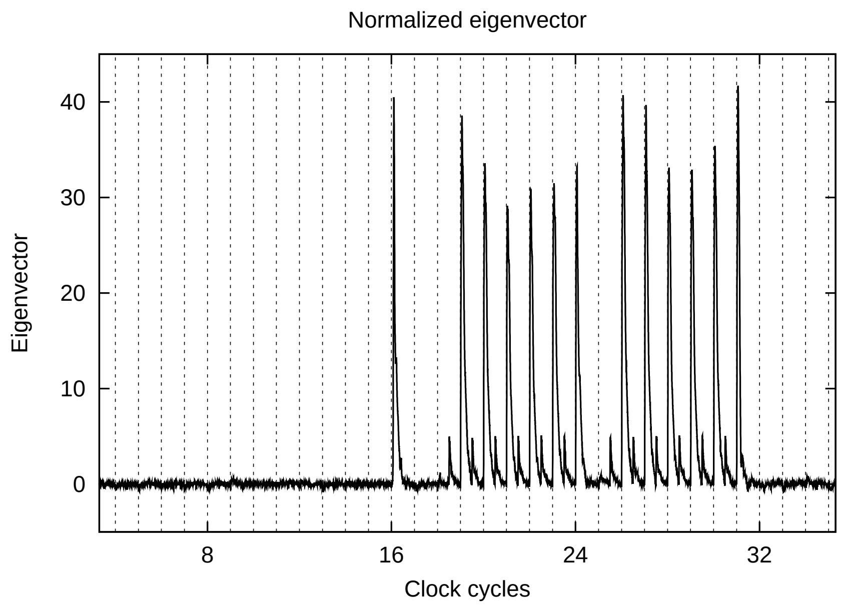


Figure 14: First eigenvector for the acquisition campaign #2 with ϕ_2.

(XXX replace both equations by that of Sec. 1)

Indeed, if V is a solution, all αV, for $\alpha \in \mathbb{R}$, satisfies the eigenvalue equation. The second relation imposes $(\alpha V)^T(\alpha V) = \alpha^2 = 1$, hence $\alpha = \pm 1$. We say that the power model is physical when the eigenvector is always of 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.

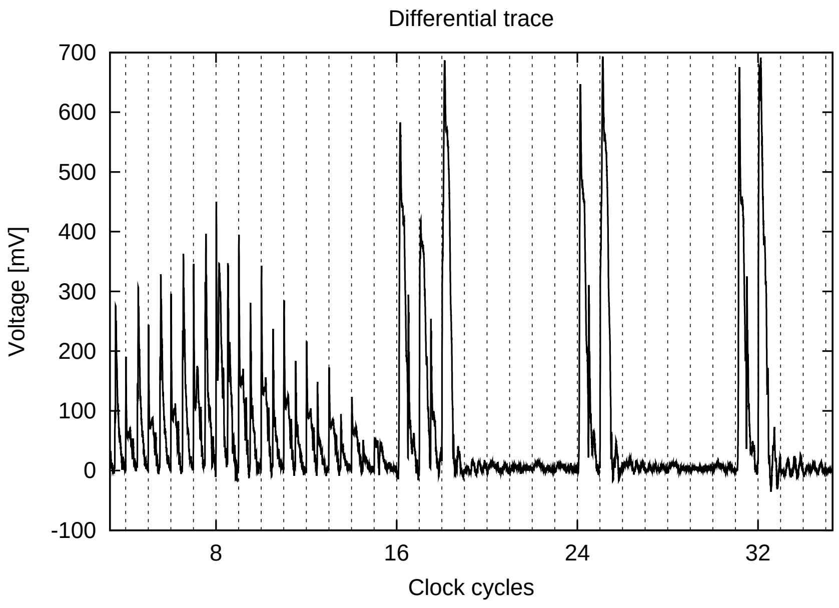
Notice that the template built with ϕ_1 as the classification function has two representative eigenvalues. The first eigenvector (Fig. 10) accounts for the LS activity in the key schedule. The second eigenvector (Fig. 11) actually grabs the dissipation from the double bit shifts LS^2. This operation 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. 11) is similar to the first eigenvector for ϕ_2 (modulo an arbitrary sign, see Fig. 14).

The template built with ϕ_2 (Fig. 12) is clearly unidirectional, which is in line with the existence of only one representative eigenvalue (Fig. 13).

In Fig. 15, the eigenvectors are compared to the differential traces obtained by the weighting [7] 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.



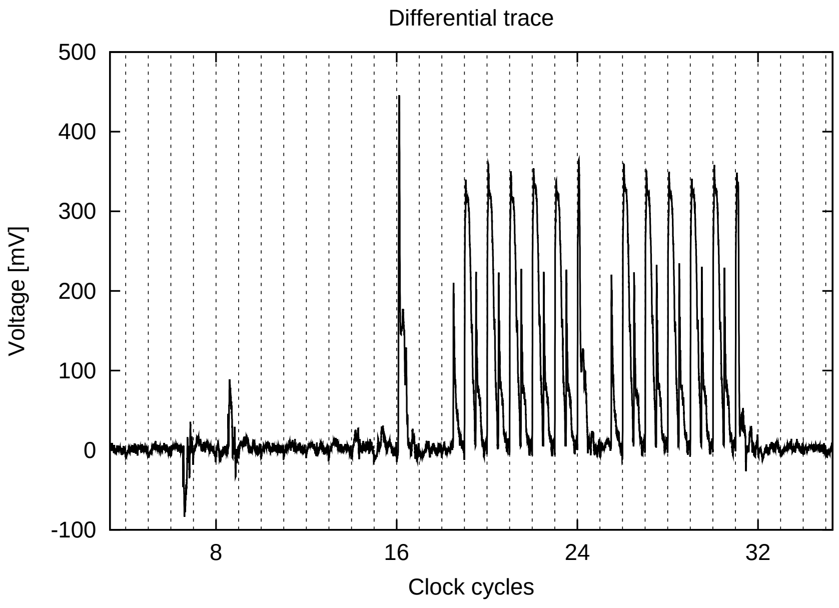


Figure 15: Differential traces obtained by traces weighting with ϕ_1 (upper) and ϕ_2 (lower).

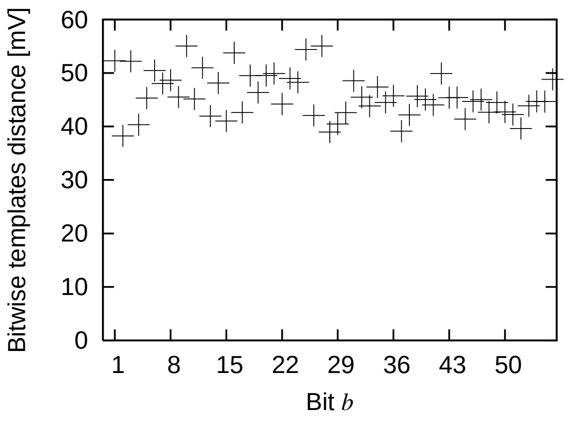


Figure 16: Distance $|\mu_0 - \mu_1|$ for the 56 $\phi_1[b]$ DES bitwise templates. (XXX update the notations XXX)

5 Bitwise Template Attacks with a Power Model

The key schedule of DES is very particular, in two respects:

1. all the round key bits are indiscernible and
2. the round keys remain correlated over the whole key schedule.

The first observation unfortunately causes the attack to be more difficult. At first order, all the bits having the same dissipation, because the key schedule handles them collectively. As a consequence, they cannot be distinguished one from each other. This means that templates are degenerated into constant Hamming distance sub-classes. In Figures 3, 8 and 12, the following color code is used to disambiguate the classes built with the classification functions ϕ_i, for $i \in \{0, 1, 2\}$:

Black	for the $\binom{6}{0} = 1$ template corresponding to $\lvert\phi_i(c)\rvert = 0$.
Red	for the $\binom{6}{1} = 6$ templates corresponding to $\lvert\phi_i(c)\rvert = 1$.
Green	for the $\binom{6}{2} = 15$ templates corresponding to $\lvert\phi_i(c)\rvert = 2$.
Blue	for the $\binom{6}{3} = 20$ templates corresponding to $\lvert\phi_i(c)\rvert = 3$.
Magenta	for the $\binom{6}{4} = 15$ templates corresponding to $\lvert\phi_i(c)\rvert = 4$.
Cyan	for the $\binom{6}{5} = 6$ templates corresponding to $\lvert\phi_i(c)\rvert = 5$.
Yellow	for the $\binom{6}{6} = 1$ template corresponding to $\lvert\phi_i(c)\rvert = 6$.

This motivates for a so-called "bitwise" template attack.

We have computed all the 56 templates for the classification functions $\phi_1[b]$, for $b \in [1, 56]$. The estimated distance $|\mu_0 - \mu_1|$ is plotted in Fig. 16. It indeed appears that all the bits are roughly indiscernible: the distance between the two templates when guessing $c_b \oplus c_{b+1}$ (modulo 28, as in LS) is the same for every bit.

The second observation accounts for the fact that eigenvectors are non-null over the whole encryption.

Even more, the key loading also disclosed information. The fact is that the same classification function happens to extract both the key loading and the singe bit shifts in the key schedule. This helps the attack, because side-channel leakage is collected over a larger period of time.

The bitwise templates $\phi_1[b]$, for $b \in [1, 8]$ are plotted in Fig. 17.

The contents of IF when the key is loaded is: key[1–8] at clock period 1, key[9–16] at clock period 2, $\cdots$, key[57–64] at clock period 8.

Expressed in CD register contents, the successive IF bytes are: $\mathrm{c}_0[8]$, $\mathrm{c}_0[16]$, $\cdots$ at clock period 1, $\mathrm{c}_0[7]$, $\mathrm{c}_0[15]$, $\cdots$ at clock period 2, $\cdots$, $\mathrm{c}_0[1]$, $\mathrm{c}_0[9]$, $\cdots$ at clock period 8.

As a result, $\phi_1[1] = c_1 \oplus c_2$ signs late (from clock period 8), whereas $\phi_1[2] = c_2 \oplus c_3$ signs one clock cycle earlier. And finally, $\phi_1[8] = c_8 \oplus c_9$ does not sign in the key loading stage, because those two bits c_8 and c_9 live in different positions of the IF eight-bit register.

6 Attack on DES Key Schedule

Not that easy. Sylvain: 31 août, 2. Figures. Observation:

1. Success rate is low
2. Success rate does not increase with the number of matching traces.

Also notice that the key neighboring bits interact with the secret that is being retrieved. ==¿ Attacks failure in general ==¿ But... There are particular keys in DES that are weak to the bitwise template attacks. They are characterized by the following property: $c_{i-2} = ci - 1$ and $c_{i+1} = ci + 2$. Thus, one key out of four is weak. For these keys, it is expected that the attack success rate increases with the number of matching traces. For others, increasing the number of matching traces do noes increase the success rate. ==¿ TODO There is a systematic bias

Explanation: The templates are not Gaussian. There are actually a mixture of gaussians...

7 Conclusion

Conclusions:

Unlike often believed, the template attacks do not attack substitution boxes. It is sometimes suggested to implement the datapath with normal logic [11], 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 sboxes are non-linear, which helps distinguishing correct key guesses from incorrect guesses.

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.

In the particular case of DES, the key schedule is original, since fully linear. It leads to the degeneracy of some key bits guess, which can be fixed by a so-called bitwise attack. In addition, the neighbor bits interfere in a systematic way, which makes some bits more difficult to guess.

Perspectives:

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

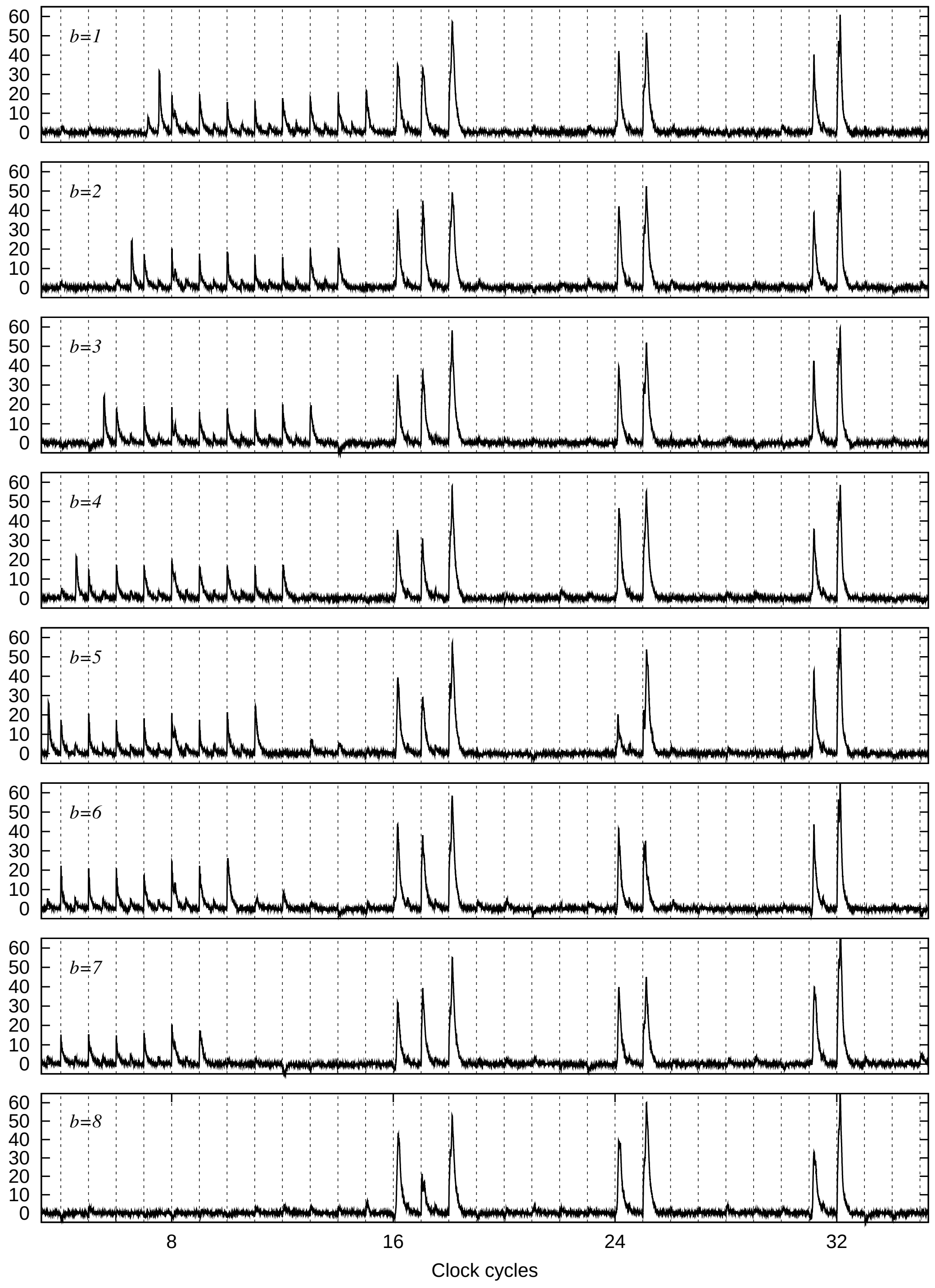


Figure 17: First (and unique) eigenvector for the bitwise template attack $\phi_1[b]$.

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.

We also note that DES key schedule is immune to template attacks with a power model. Non-linear key schedule (AES, Khazad, MIST) are expected to be much more easily breakable with templates.

References

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

[2] Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template attacks in principal subspaces. In *CHES*, 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] Sylvain Guilley. *Geometrical Counter-Measures against Side-Channel Attacks*. PhD thesis, ENST / CNRS LTCI, January 2007. `http://pastel.paristech.org/2562/`.

[7] 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.

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

[9] 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. Springer-Verlag, (PDF).

[10] 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.

[11] Thomas Popp, Mario Kirschbaum, Thomas Zefferer, and Stefan Mangard. Evaluation of the Masked Logic Style MDPL on a Prototype Chip. In *CHES'07*, September 2007.

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

[13] S. Guilley and 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.

[14] Sylvain Guilley and 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.