An Overview of Side Channel Analysis Attacks

Thanh-Ha Le
University of Luxembourg
6, rue Richard
Coudenhove-Kalergi
L-1511 Luxembourg
thanhha.le@uni.lu

Cécile Canovas
CEA-LETI
17 rue des Martyrs
38054 Grenoble cedex 9,
France
cecile.canovas@cea.fr

Jessy Clédière
CEA-LETI
17 rue des Martyrs
38054 Grenoble cedex 9,
France
jessy.clediere@cea.fr

ABSTRACT

During the last ten years, power analysis attacks have been widely developed under many forms. They analyze the relation between the power consumption or electromagnetic radiation of a cryptographic device and the handled data during cryptographic operations. The goal of this paper is to give a global view of statistical attacks based on side channel analysis. These techniques are classified into two classes: attacks without reference device (e.g. Differential Power Analysis, Correlation Power Analysis) and attacks using a reference device (e.g. Template Attack, Stochastic Model Attack). In this paper, we present the attacks with an easy comprehensible way and focus on their implementation aspect. The pros and cons of each attack is highlighted in details with concrete electromagnetic signals. At least, our paper proposes also some solutions to enhance the existing attacks.

Categories and Subject Descriptors

E.3 [**Data**]: Data Encryption; B.8 [**Performance and Reliability**]: Miscellaneous

General Terms

Security

Keywords

Side Channel Attacks, DPA, DEMA, CPA, Template Attack, Stochastic Model.

1. INTRODUCTION

Since the first publication of Kocher *et al.* [1] titled "Differential Power Analysis", many power analysis attacks have been developed. Power analysis attacks exploit the dependence between the instantaneous power consumption of a cryptographic device and the data it processes and/or the operation it performs. This type of attacks is known as a powerful technique for revealing confidential data (e.g. a secret key of a cryptographic algorithm) because it is imperceptible to users and it does not require expensive equipments like the micro probing technique [2, 3].

The Simple Power Analysis (SPA) [1] is a visual inspection using only one (or very few) power consumption signals measured during cryptographic operations. The Differential Power Analysis (DPA) is a statistical test which examines a large number of power consumption signals to retrieve secret keys. The DPA is multiform itself. It can be performed by analyzing the intermediate values of one bit (mono-bit DPA) [1] or a set of several bits (multi-bit DPA) [4, 5, 6]. It can also be observed at one instant of time [1] (first-order DPA) or at some instants of time (higher-order DPA) [7, 8, 9]. In recent years, the Correlation Power Analysis (CPA) technique based on the correlation between the real power consumption of the device and a power consumption model has been widely studied [10, 11, 12, 13]. It is demonstrated that CPA can be written under a form of DPA divided by a normalization factor [6].

Electromagnetic radiation signals acquired by dedicated sensors were also successfully used to detect secret information [14, 15, 16]. One can combine power consumption and electromagnetic signals to perform multi-channel attacks [17]. For the sake of simplicity, hereafter, the terms SPA, DPA and CPA will be generalized to any side channel signals. Thus, speaking of power consumption models or analysis can be directly extended to electromagnetic radiation signals.

All DPA and CPA attacks are based on a power consumption model such as the Hamming weight model[1], the Hamming distance model[2], or the most recent "switching distance leakage model" [18]. However, in practice, these models do not always fit totally to the real power consumption of a device. The idea of using a reference device, which is identical (or very close) to the attacked one, to build a database stocking power consumption information dedicated to a type of device was initially proposed in [19]. This class of attacks was then developed under the name Template Attack [20]. It consists of two stages: a profiling stage and a key extraction stage. The profiling stage is performed on a large number of signals to learn details of the device implementation. In the key extraction stage, the secret key is obtained by analyzing very few signals. The template attack does not try to

[1]Hamming weight of a set of bit $\mathcal{B}$ is the number of bits 1 of $\mathcal{B}$

[2]Hamming distance of a set of bit $\mathcal{B}$ is the number of 0-1 and 1-0 transitions between $\mathcal{B}$ and a reference state $\mathcal{R}$

ASIACCS '08, March 18-20, Tokyo, Japan

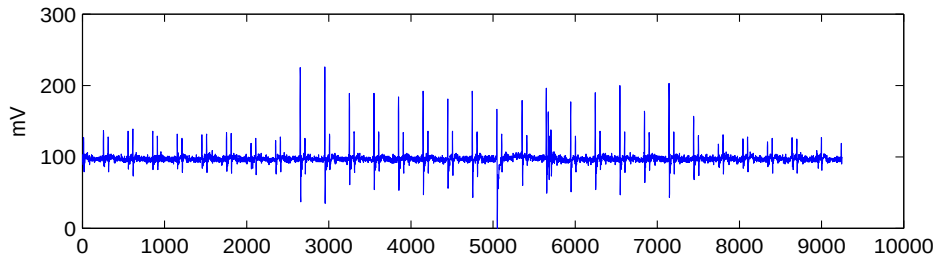

Figure 1: Electromagnetic signal

reduce noise but uses a multivariate-Gaussian noise model to extract information present in a single sample.

The stochastic model attack proposed in [21] can be considered as a combination of the attacks based on power consumption models (e.g. DPA, CPA) and the template attack. The power consumption of the device is estimated by a model with predefined functions. Moreover each predefined function will be balanced by a weight, which varies from a type of device to another. Hence, the stochastic model attack is also composed of two stages: the profiling stage to determine the weights of predefined functions and noise distribution; and the key extraction stage to detect the secret key.

In the power analysis, the key detection is possible because of the dependency between the power consumption of devices and the intermediate values of the cryptographic algorithms. Therefore, if we want to prevent from power analysis attacks, this dependency should be broken. Countermeasures against power analysis are distinguished into two categories: hardware and software countermeasures. The goal of hardware countermeasures is to randomize the power consumption of the device. As a consequence, the dependency between the power consumption and the intermediate values is hidden. It can be done by adding a random noise or desynchronizing power consumption signals with a time jitter or a Random Process Interrupts [22, 23]. Software countermeasures, as for them, can be implemented at the algorithm level without changing the power consumption characteristics of the cryptographic device. They use the masking technique to randomize the intermediate values [24, 25].

In the context where many different power analysis methods have been proposed, a synthesis and a multifaceted investigation are both a need and a demand for further enhancements. To the best of our knowledge, there does not exist any paper which evaluates DPA, CPA, template attack and stochastic model attack side-by-side, in a same measurement condition and with an identical cryptographic device. Hence, our aim is to perform such an evaluation with real experimental signals to first explain and illustrate thoroughly each method, and then show out its advantages and inconveniences. In our experiment, we use a device in which countermeasures are not implemented. However, one may note that if hardware countermeasures are present, the sliding window technique can be used to reduce the effects of noise and of desynchronization as proposed in [22, 26, 27]. Regarding software countermeasures with the masking technique, the higher-order DPA can be applied [7, 8, 9].

The paper is structured as follows. In Section 2 we investigate the attacks which do not need any reference cryptographic device like DPA and CPA. The power analysis attacks using a reference device such as the template attack or the stochastic model attack are studied in Section 3. The comparison of two power analysis classes is shown in Section 4 followed by conclusions in the last section.

2. SIDE-CHANNEL ATTACKS WITHOUT REFERENCE DEVICE

In order to perform the attacks, we measure the electromagnetic emanations of a synthesized ASIC during a Data Encryption Standard (DES) operation. We want to detect the sub-key used in the first S-box of the first round of DES. The size of the sub-key is 6 bits, so there are 64 key assumptions. An electromagnetic signal as presented in Figure 1 is obtained when a plain text is encrypted.

In this section, we first present the classical mono-bit DPA. We synthesize then two ways to generalize the DPA. We consider the Partitioning Power Analysis (PPA) method [6] and propose two conditions that make it possible to enhance PPA. The correlation analysis CPA is studied in the last subsection with a detailed evaluation about the solution to reduce noise.

2.1 Mono-bit Differential Power Analysis

The original DPA is based on the fact that the power dissipation to manipulate a bit b to 1 is different from the power dissipation to manipulate it to 0. To test different key assumptions K_k ($k = 0, ..., 63$ in our case), DPA uses N texts C_i ($i = 1 \dots N$). Let us denote $W(C_i)$ as the power consumption (or electromagnetic) signal corresponding to the text C_i. In function of the intermediate values[3] of b (0 or 1) estimated by key assumption K_k, N signals $W(C_i)$ will be distributed in two groups:

$$G_{0,k} = \{W(C_i), i = 1 \dots N | H(C_i, b, K_k) = 0\}$$

$$G_{1,k} = \{W(C_i), i = 1 \dots N | H(C_i, b, K_k) = 1\}$$

where $H(C_i, b, K_k)$ is the Hamming weight (or the Hamming distance to a reference state) of b, corresponding to text C_i and estimated by key assumption K_k. Let us denote $N_{0,k}$ and $N_{1,k}$ as the number of elements of $G_{0,k}$ and $G_{1,k}$ respectively. The DPA signal corresponding to key assumption K_k is $\Delta_k(b)$:

$$\Delta_k(b) = \frac{\sum_{G_{1,k}} W(C_i)}{N_{1,k}} - \frac{\sum_{G_{0,k}} W(C_i)}{N_{0,k}} \qquad (1)$$

In theory, if the assumption K_k is correct, the $\Delta_k(b) \neq 0$ at the instant τ when b is handled. It is thus represented

[3]For example, the Hamming weight or the Hamming distance of one bit in the output of an S-box.

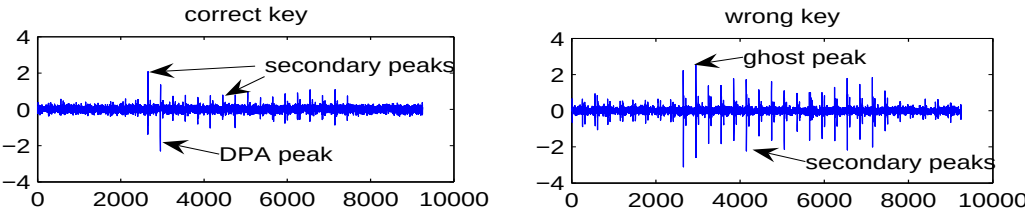

Figure 2: DPA signals of the correct key and a wrong key.

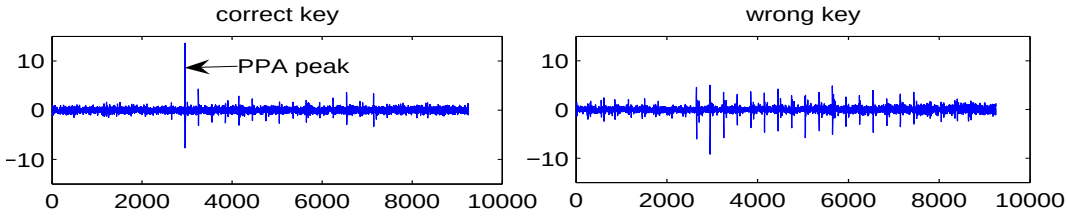

Figure 3: PPA signals of the correct key and a wrong key ($d = 4$)

by a peak in the DPA signal at the instant τ (DPA peak). For incorrect key assumptions, $\Delta_k(b)$ tends to 0 and no significant peak appears. However, in practice, due to the bit distribution conditions and some output correlations of key assumptions, we observe other peaks at τ in DPA signals corresponding to wrong key assumptions (ghost peaks), or secondary peaks which appear at an instant other than τ in a DPA signal corresponding to any key assumption (see Figure 2).

2.2 Multi-bit Differential Power Analysis

The first way to generalize the mono-bit DPA is to extend the notion of groups as proposed the method Partitioning Power Analysis (PPA) [6]. Instead of considering one bit b, a set of d bits $\mathcal{B} = b_1 b_2 ... b_d$[4] is examined. The group partition will be then based on the estimated Hamming weight $H(C_i, \mathcal{B}, K_k)$ of $\mathcal{B}$, corresponding to text C_i and key assumption K_k (or the Hamming distance compared to a reference state $\mathcal{R}$). Hence, the number of groups is $(d+1)$.

$$G_{j,k} = \{W(C_i), i = 1 \dots N | H(C_i, \mathcal{B}, K_k) = j\}$$

for $j = 0, ..., d$. The PPA signal corresponding to key assumption K_k is given by:

$$\Sigma_k(\mathcal{B}) = \sum_{j=0}^{d} a_{j,k} \frac{\sum_{G_{j,k}} W(C_i)}{N_{j,k}} \qquad (2)$$

where $N_{j,k}$ is the number of elements of group $G_{j,k}$ and $a_{j,k}$ is the weight corresponding to $G_{j,k}$.

The weights $a_{j,k}$ can be dependent or independent of key assumption K_k. In latter case, we have $a_{j,k} = a_j$ for any key K_k. From Figure 2 and Figure 3, we observe that the ghost peaks and secondary peaks raise highly in the mono-bit DPA signals. They are reduced in the case of PPA signals resulting from $d = 4$ bits. This shows the advantage of multi-bit concept against the mono-bit one.

The performance of PPA depends on the choice of weights a_j but this dependency has not been shown in [6]. Therefore, in this paper, we want to determine the conditions related to weights a_j for an efficient key detection. Assume that the relation between the real power consumption W of the device and the Hamming weight (or the Hamming distance) H is represented by a model F ($W = F(H)$). Two conditions on weights a_j are (see more details in Appendices 1 & 2):

1. The PPA signal $\Sigma_k^w(\mathcal{B})$ corresponding to a wrong key assumption is equal to zero:

$$(\sum_{j=0}^{d} a_j)(\sum_{k=0}^{d} \frac{C_d^k}{2^d} F(k)) = 0 \Leftrightarrow \sum_{j=0}^{d} a_j = 0 \qquad (3)$$

2. The signal-to-noise ratio (SNR) of the PPA signal corresponding to the correct key is maximized.

$$\max_{\{a_j\}} SNR = \frac{|\sum_{j=0}^{d} a_j F(j)|}{\sqrt{\sum_{j=0}^{d} \sigma . a_j^2 / N_{j,k}}} \qquad (4)$$
$$\Leftrightarrow \quad a_2 = 0, -2a_0 = -a_1 = a_3 = 2a_4$$
in case of F linear and $d = 4$.

where σ is the standard deviation of noise.

The first condition $\sum_{j=0}^{d} a_j = 0$ is interesting because it does not depend on the power consumption model F. It means that if no information about the device is available, it is still possible to detect the secret key by only choosing a_j satisfying this condition. The second condition shows a relation between weights a_j and power consumption model F. For each model F, one can make the PPA attack more powerful by selecting suitable weights a_j.

In order to evaluate the attack performance, we use two indexes presented in [6]. The first index i_1 is defined as the ratio between the PPA peak corresponding to the correct key (expected peak) and the highest PPA peak resulting from wrong keys (ghost peaks). These peaks are observed at the same time location τ when the data are handled. If this index is greater than 1, the expected peak is higher than any ghost peak and the key detection is reliable. The second

[4]For example, 4 bits in the output of an S-box.

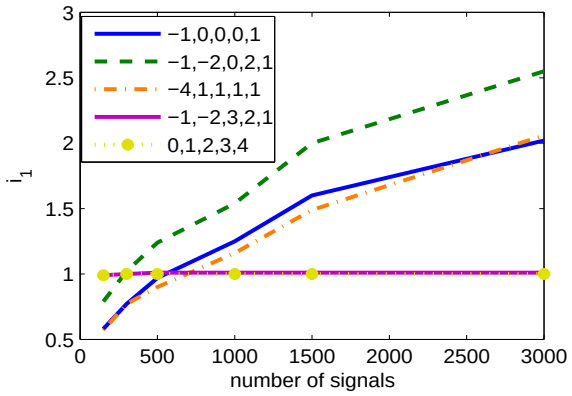

Figure 4: PPA: variation of i_1 in function of the number of signal

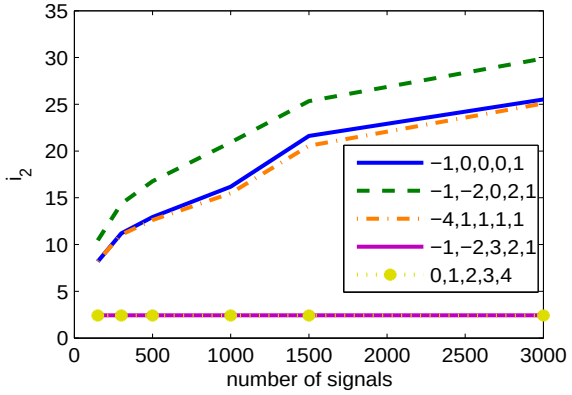

Figure 5: PPA: variation of i_2 in function of the number of signals

index, denoted as i_2, is the signal-to-noise ratio of the PPA signal corresponding to the correct key.

To validate two previous conditions, we test the PPA attack with different sets of weights:

- sets of weights a_j whose sum is not equal to 0: {-1, -2, 3, 2, 1} and {0, 1, 2, 3, 4},
- sets of weights a_j whose sum is equal to 0: {-4, 1, 1, 1, 1} and {-1, 0, 0, 0, 1},
- the optimal set of weights a_j of a linear model F: {-1, -2, 0, 2, 1}.

The Figures 4 and 5 represent the variation of i_1 and i_2 corresponding to the five sets of weights. They show that if the sum of weights is different from 0 ({-1, -2, 3, 2, 1} and {0, 1, 2, 3, 4}), index i_1 is always equal to 1. It means that the PPA cannot distinguish the difference between the signal of the correct key and the one of a wrong key. In the contrary, if the sum of weights is equal to 0 ({-4, 1, 1, 1, 1} and {-1, 0, 0, 0, 1}), we can detect the correct key from about 800 signals with a good signal-to-noise ratio. The optimal set {-1, -2, 0, 2, 1} gives the best performance and the correct key can be detected with only about 300 signals.

The second way to perform the multi-bit DPA of the set of d bits $\mathcal{B}$ is to calculate the mono-bit DPA signals $\Delta_k(b_n)$ for each bit b_n of $\mathcal{B}$ $(n = 0, ..., d)$ and then take the sum of these signals as proposed in [5]. The multi-bit DPA signals is $\Sigma_k(\mathcal{B}) = \sum_{n=0}^{d} \Delta_k(b_n)$. Two methods to improve this multi-bit DPA were proposed in [28]. These solutions consider that all the targeted bits do not give the same contribution to the power consumption. Therefore, each mono-bit DPA signal $\Delta_k(b_n)$ is balanced by a weight α_n.

$$\Sigma_k(\mathcal{B}) = \sum_{n=0}^{d} \alpha_n \Delta_k(b_n) \tag{5}$$

The first method proposed in [28] consists of finding out the optimal ratio among the weights α_n of a specific device. It is thus an attack based on a reference device that we will discuss in details in the next section. The second one is a statistical analysis which does not need a reference device as well as the suitable set of weights. It tests a large number of sets of weights $\{\alpha_n\}$. For each set, the key assumption that gives the highest DPA peak is considered as the correct key. Therefore, each key assumption has a frequency of being designated as the correct key. The key assumption which gives the highest occurrence frequency is considered as the correct key. Although a large number of sets $\{\alpha_n\}$ are tested, the mono-bit DPA signals $\Delta_k(b_n)$ are computed only one time for each key assumption K_k. Therefore, the computation time of this method is not so long.

2.3 Correlation Power Analysis

The CPA exploits the correlation between the power consumption W of a device and its power consumption model F [12, 11, 13]. The most common model is given under a linear form such as the Hamming weight model and the Hamming distance model. The correlation factor between W and F is proportional to the correlation factor between W and H. The correlation factor of CPA is given by the following formula [13]:

$$\hat{\rho}_{WH,k}(\mathcal{B}) = \frac{E(W.H_k) - E(W).E(H_k)}{\sigma_W \sigma_{H_k}} \tag{6}$$

where $E(W)$, $E(H_k)$, $E(WH_k)$ represent the expectations of W, H_k (the values of H estimated by the key K_k) and $W.H_k$; σ_W and σ_{H_k} are the variances of W and H_k. The Figure 6 illustrates CPA signals corresponding to the correct key and a wrong key assumption.

According to [6], the correlation between W and H can be rewritten under a PPA form divided by a normalization factor:

$$\hat{\rho}_{WH,k}(R) = \frac{\sum_{j=0}^{d} \left(\alpha_{j,k} \frac{\sum_{G_{j,k}} W(C_i)}{N_{j,k}} \right)}{\sigma_W \sigma_{H_k}} \tag{7}$$

It is demonstrated in [6] that the normalization factor induces a high noise level in CPA signals. A solution was proposed to reduce this normalization effect by adding to $\sigma_W(t)$ a positive constant ε:

$$\hat{\rho}_{WH,k}(\mathcal{B}) = \frac{\Sigma_k(\mathcal{B})}{(\sigma_W + \varepsilon)\sigma_{H_k}} \tag{8}$$

The enhancement of CPA depends thus on the choice of ε. In order to illustrate this dependency, we use two indexes i_1

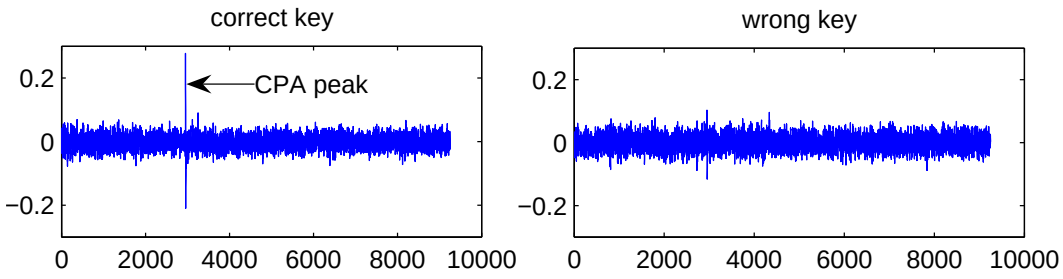

Figure 6: CPA signals of the correct key and a wrong key ($d = 4$)

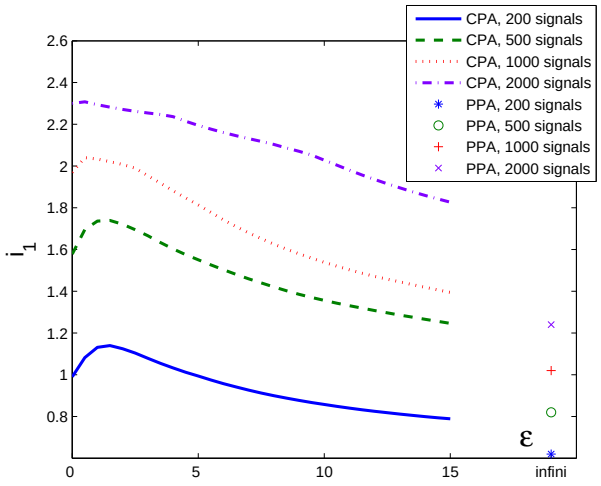

Figure 7: CPA: variation of i_1 in function of ε

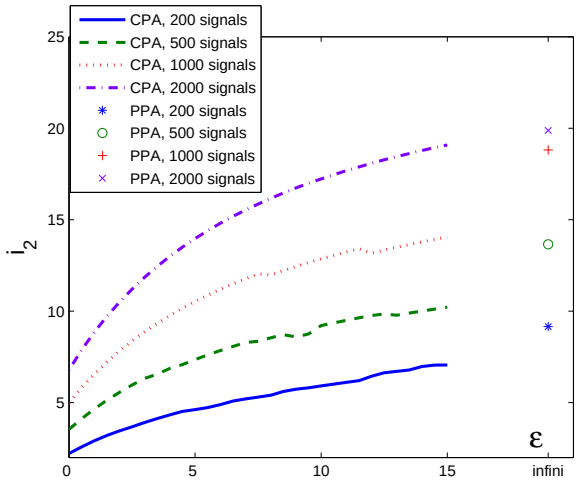

Figure 8: CPA: variation of i_2 in function of ε

and i_2 previously presented. The value of ε varies from 0 to 15 mV. When ε tends to ∞, the normalization factor tends to a constant and the CPA will be equivalent to the PPA. The variation of i_1 and i_2 in function of ε corresponding to the CPA method is shown in Figure 7 and Figure 8. We observe that the fact of adding a positive constant ε allows CPA to improve indexes i_1 and i_2. However, if the value ε is too high, the correlation factor is modified and by consequence index i_1 reduces. Hence the positive ε should be selected in such a way to compromise the performance of i_1 and i_2.

In conclusion, the CPA attack can be considered as a multi-bit DPA attack. The epsilon-adding method allows CPA to reduce the noise level of signals. The choice of ε depend on the signal form, the noise level and the number of signals used in CPA. Our evaluation on value ε presented in this subsection shows the influence of ε on the CPA performance and gives a useful indication on how to choose it.

3. SIDE CHANNEL ATTACKS WITH A REFERENCE DEVICE

In this section, we focus on the template and the stochastic attacks. Instead of considering the signal as a random variable that depends on time, we consider it as L random variables at different *points of interest* $t_1 \dots t_L$. Thus the signal study is bi-dimensional according to time and input messages with covariance matrices that represent the noise. Moreover these methods use the noise estimation instead of the correlation to a consumption model. For each method, a summary is presented in the first time followed by experimental results obtained from our electromagnetic signals.

3.1 Template Attack

3.1.1 *Survey of the template attack*

As the DPA and CPA attacks are based on power consumption models, their efficiency is strongly depends on the considered one. If the power consumption is wrongly modelled, the key detection is impossible. In general, these attacks need a large number of signals to detect the secret key due to the presence of noise. This condition is sometimes difficult to satisfy in practice. Facing up to the "shortcoming" of DPA and CPA attacks, a new class of power attacks was initially proposed in [19] and then developed under the well-known name, the Template Attack [20]. This class of attacks contains two stages: the profiling stage to learn about the device and the key extraction stage to detect the secret key.

In the ***profiling stage***, a large number of signals is used to build a database dedicated to a type of device. The database contains different *templates* which are important elements of the template attack. In the original work [20], a template $\mathcal{T}(C_i, K_k)$ is defined for each pair of text C_i and key assumption K_k. A template is composed of *mean signal* $m_{i,k}$ computed from $N_{i,k}$ power consumption signals and *covari-*

ance matrix $C_{i,k}$ which represents the noise probability distribution. Using this definition, the number of template is very large. For example, in the case of DES, there are 2^6 possibilities for C_i and 2^6 possibilities for K_k. Hence there are in total $2^{12} = 4096$ templates. One may note that the attack based on templates $\mathcal{T}(C_i, K_k)$ is not practical. Firstly, the number of templates is too large and by consequence, the time to build the database and to find the key is very long. Moreover, one has to build the templates for all key assumptions. The latter condition is not always attained in practice since the encryption key can only be changed if the device is in the test mode.

In order to make the template attack more practical, one can employ other types of templates. Template based on bit values like the output of an S-box used in [29] is a solution. The output contains 4 bits, there are thus $2^4 = 16$ templates, denoted by $\mathcal{T}(V)$ $(V = 0, ..., 15)$. Each template contains also mean signal m_V and covariance matrix C_V. We can also use a function like the Hamming weight (distance) to build templates. We have 5 templates corresponding to 5 possible Hamming weight values of an S-box output. The templates is denoted by $\mathcal{T}(H)$ $(H = 0, ..., 4)$. As usual, a template $\mathcal{T}(H)$ consists of mean signal m_H and covariance matrix C_H.

Attacks using the templates $\mathcal{T}(V)$ or $\mathcal{T}(H)$ are easier to implement because the number of templates is only 16 or 5 respectively and we only use one key (which is already implemented in the device) to build all templates. However, these solutions do not make it possible to determine directly the secret key by analyzing only one signal. It is due to the fact that there does not exist a one-to-one mapping relation between the templates and the key assumptions. More precisely, if the templates $\mathcal{T}(H)$ are considered, 64 key assumptions are mapped to only 5 templates. Therefore, several signals must be employed in the second stage to find out the secret key. Note also that the computing and processing time of the covariance matrices is related to the signal size L. Hence, one should not take all samples present in measured signals to compute the covariance matrix. Only the points of interest, where the leakage is significant, are considered. Several solutions to select these points were proposed in [30, 21].

The task of the ***key extraction stage*** is to determine the correct key using the database built in the profiling stage. Multivariate noise statistics are applied to extract the maximum of information from a single signal (or very few signals). Noise, represented by $(s - m)$, is supposed to be Gaussian. The probability that signal s of length L corresponds to the template of mean signal m and covariance matrix C is[5] [20]:

$$p(s; (m, C)) = \frac{\exp(-\frac{1}{2}(s-m)^T C^{-1}(s-m))}{\sqrt{(2\pi)^L det(C)}} \qquad (9)$$

Such detection method is based on the maximum likelihood metric (ML). The best template of s is the one giving the highest probability $p(s; (m, C))$.

Some authors [21] simplify the computation of the detector $p(s; (m, C))$ by setting the covariance matrix to the identity matrix (do not consider the covariances between the points). In this case, Eq. (9) becomes:

$$\begin{aligned} p(s; m) &= \frac{\exp(-\frac{1}{2}(s-m)^T(s-m))}{\sqrt{(2\pi)^L}} \\ \ln(p(s; m)) &= -\frac{1}{2}(s-m)^T(s-m) - \frac{L}{2}\ln(2\pi) \quad (10) \end{aligned}$$

From equation (10), maximizing $p(s; m)$ becomes minimizing the Euclidean distance between two vectors s and m. In this case, the detection method is referred to the minimal distance metric (MD), which uses only the mean signals.

In order to reveal the secret key, in the key extraction stage, we need to analyze a set of q signals s_i $(i = 1 \dots q)$ associated with texts C_i. Assume that templates based on the Hamming distance $\mathcal{T}_H$ $(H = 0 \dots 4)$ are used. For each signal s_i, if the template $\mathcal{T}_H$ is estimated, the probability $p(s_i; (m_H, C_H))$ will be added up to all key assumptions corresponding to $\mathcal{T}_H$. At the end, when all q signals s_i are processed, the key assumption which has the maximal sum will be considered as the correct key.

3.1.2 Performance evaluation

In our experiment, we build the templates based on the Hamming distance. There are thus five templates. The attack using bit value based templates, which are not presented here, can be performed in a similar way. We select L points around the instant τ where data are handled as points of interest. 3000 electromagnetic signals are used to build the data base. Both maximum likelihood and minimal distance metrics are tested in the key extraction stage to compare the efficiency of these metrics.

The Figures 9 and 10 represent the mean signals m_H $(H = 0 \dots 4)$ corresponding to five templates. A test signal s corresponding to the template 1 is presented by the dotted curve. We observe that the distance between the test signal and the corresponding mean signal m_1 is much larger than the distances among mean signals $m_0, ..., m_4$ themselves. We performed the noise variation computation and the result shows that the noise standard deviation of electromagnetic signals is about $7mV$ compared to the maximal distance between two consecutive mean signals at the top of peaks $v = 1.7mV$ (see Figure 10). It means that if only mean signals are used in the key extraction stage like the MD metric, the detection may not be exact due to a high noise level.

We study the performance of the ML and MD metrics in the *template estimation* when only one signal is used in the key extraction stage. The Figure 11 represents the distribution of test signals into 5 templates (one signal per test). The first histogram corresponds to the correct distribution of signals into templates; the second histogram corresponds to the distribution of templates estimated by the MD metric; the third histogram corresponds to the distribution of templates estimated by the ML metric. As noise is high compared to the distance between mean signals, the test signal s can be easily out of the zone Z with MD metric (see Figure 10) and the estimated template is either 0 or 4 (the template 4 instead of the template 1 in this case). Therefore, the second histogram has high values for the template 0 and the template 4. Otherwise, with ML metric, we observe that the distribution is very similar to the first one. The templates are well estimated by the ML metric.

We evaluate the *template estimation* in relation to the number of points of interest L. We define $p^{\mathcal{T}}_{MD}$ $(p^{\mathcal{T}}_{ML})$ the

[5] Mean signal m can be $m_{i,k}, m_V, m_H$ and covariance matrix C can be $C_{i,k}, C_V, C_H$

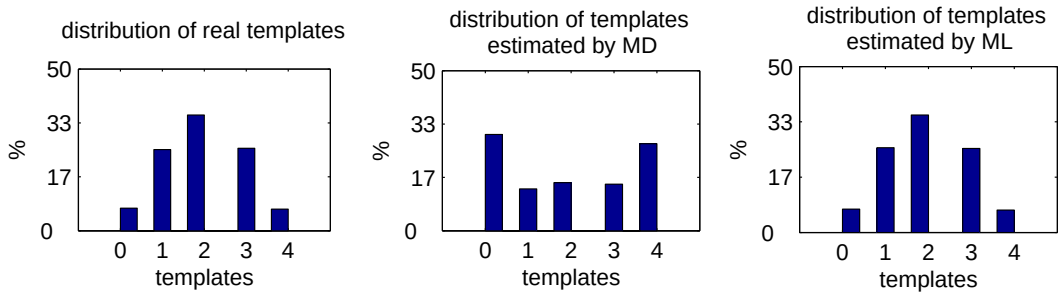

Figure 11: Different distributions of templates. Signal size $L = 100$

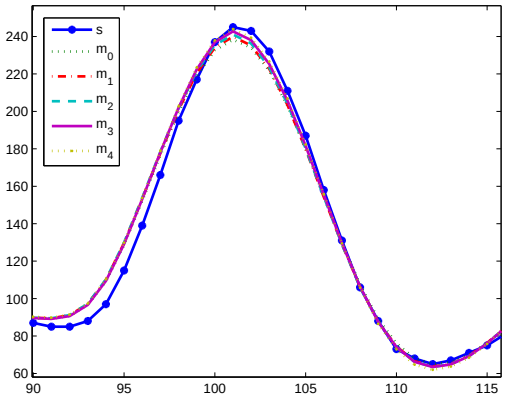

Figure 9: Mean signal of 5 templates with 50 points around τ and a test signal used in the key extraction stage

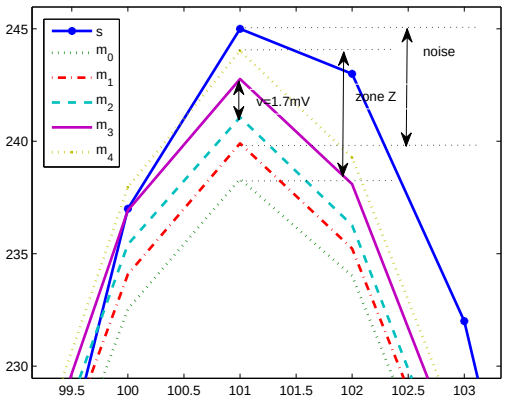

Figure 10: Zoom of Figure 9

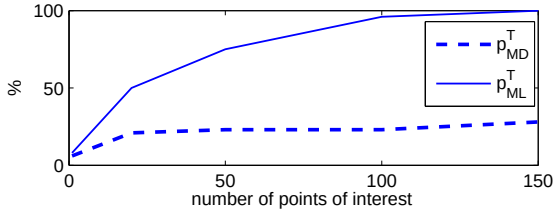

Figure 12: Variation of $p_{MD}^{\mathcal{T}}$ and $p_{ML}^{\mathcal{T}}$ in function of number of points of interest L

Table 1: p_{MD} and p_{ML} of the template attack.

Num. of signals	4	6	8	10	20	40
$p_{MD}(\%)$	7	8	9	10	13	21
p_{ML} (%)	92	96	99	**100**	**100**	**100**

empirical probability (occurrence percentage) that a signal is estimated to the correct template by the MD (ML) metric. A method is efficient if its template estimation frequency is high. The Figure 12 represents the evolution of $p_{MD}^{\mathcal{T}}$ and $p_{ML}^{\mathcal{T}}$ in function of the number of points of interest L. The $p_{ML}^{\mathcal{T}}$ tends to 100% with about 150 points of interest and the $p_{MD}^{\mathcal{T}}$ is always low (about 20%). The result shows clearly the advantage of the ML metric, which uses information of noise from covariance matrices, compared to the MD metric, which uses only mean signals.

As stated previously, the secret key detection needs several signals in the key extraction stage. The *key detection* performance is now examined. We define p_{MD} (p_{ML}) the occurrence percentage that the correct key is correctly detected with the MD (ML) metric. The Table 1 gives different values of p_{MD} and p_{ML} according to the number of signals used in the second stage. The number of points of interest L is set at 100 in this case. It shows that by using the maximum likelihood metric, the secret key can be correctly detected with only 10 signals. The maximum distance metric is undoubtedly less efficient.

According to the previous experimental results, we demonstrate that the template attack can be easily implemented using templates based on the Hamming weight (distance) function. Useful information contained in noise is well exploited by using the maximum likelihood metric. The attack

is powerful since the secret key can be detected with only 10 signals in the key extraction stage.

3.2 Stochastic Model Attack

3.2.1 Implementation

The stochastic model attack [21] is based on two stages as the template attack. In the ***profiling stage***, instead of using mean signal m computed from measured signals, the stochastic model attack estimates the power consumption by predefined functions. In the case of DES, four predefined functions are defined as the values of four bits b_n $(n = 1 \dots 4)$ in the output $\mathcal{B}$ of an S-box. The stochastic power consumption model is then formulated as:

$$P(\mathcal{B}) = \beta_0 + \sum_{n=1}^{4} \beta_n b_n \tag{11}$$

The weight β_0 represents the non-data dependent signal part, it is thus fixed for every value of $\mathcal{B}$. The weights β_n $(n = 1 \dots 4)$ are the bit-wise data dependent signal portions. According to [21], vector β containing weights β_n $(n = 0, .., 4)$ is computed as: $\beta(t) = (A^T A)^{-1} A^T s(t)$. The vector $s(t)$ corresponds to the values at the instant t of N power consumption signals used in the profiling stage and $A = \{a_{i,n}\}$ is matrix of size $N \times 5$. All elements of the first column of matrix A, which correspond to the fixed weight β_0, are set to 1. Element $a_{i,n}, n \neq 0$ corresponds to the value of bit b_n $(n = 1 \dots 4)$ when text C_i are encrypted by the correct key. A covariance matrix which represents the noise distribution is also computed in the profiling stage for each value of $\mathcal{B}$. There are thus 16 covariance matrices.

In the ***key extraction stage***, the weights β_n $(n = 0 \dots 4)$ calculated in the profiling stage are used to estimate the power consumption corresponding to each value of $\mathcal{B}$, instead of the mean for template attack. It is the main difference between the template attack and the stochastic attack. Both MD and ML metrics can be employed to estimate the noise and detect the secret key as in the template attack.[6]

3.2.2 Experimental results

Figure 13 represents the variation of $\beta_n(t)$ $(n = 0, .., 4)$ in time. We observe that the non-data dependent signal part (β_0) is very large compared to the bit-wise data dependent signal portions $(\beta_1, ..., \beta_4)$. The weights $\beta_1, ..., \beta_4$ are zoomed in Figure 14. These weights vary in time and they are not identical. It means that a Hamming distance model would not really precise in this case since this model considers that all bits are equivalent. If the weights $\beta_1, ..., \beta_4$ are identical, the stochastic model attack is exactly the template attack based on the bit values $\mathcal{T}(V)$.

In the key extraction stage, for each test signal, there are 64 key assumptions which map to only 16 possibilities of $\mathcal{B}$[7]. Hence, there does not exist a one-by-one mapping relation between the stochastic power consumption model and key assumptions. Like the template attack based on a model or intermediate values, the stochastic attack can not detect directly the secret key with only one signal in the key extraction stage. One should use several power consumption signals to estimate the secret key.

[6]In case of MD metric, the covariance matrices are the identity.

[7]There are 4 bits b_n, each bit has 2 possibilities 0 and 1. So there are in total $2^4 = 16$ possibilities of $\mathcal{B}$

Table 2: p_{MD} and p_{ML} of the stochastic model attack.

Num of signals	4	6	8	10	20	40
$p_{MD}(\%)$	5	5	6	7	10	27
p_{ML} (%)	**100**	**100**	**100**	**100**	**100**	**100**

We use the same notation p_{ML} and p_{MD} to represent the occurrence percentage that the secret key is correctly detected with the ML and MD metrics. The points of interest are similarly chosen as in the template attack with $L = 100$ points. The variation of p_{ML} and p_{MD} of the stochastic attack in function of number of signals used in the key extraction stage is given in the Table 2. Once again, the attack based on the ML metric is much more powerful than the one based on the MD metric. It is explained by the fact that the ML metric exploits useful information contained in noise. The stochastic model attack with ML metric needs only about 4 signals in the key extraction stage to find out the secret key. This result is better than the one of the template attack since the stochastic model template takes into account the non-equivalence among the bits by using the stochastic weights β_n in the power consumption model.

The multi-bit DPA presented at the end of the subsection 2.1 uses also the weights α_n to compensate the imbalance of bits. These weights can be determined by using a reference device. However this multi-bit DPA attack and the stochastic model attack are not identical. In the case of the multi-bit DPA, the weights α_n are applied to mono-bit DPA signals to obtain a multi-bit DPA signal, which allows a more efficient key detection. Meanwhile, the weights β_n of the stochastic model attack are applied to the predefined functions to model precisely the power consumption of the device.

In conclusion, the stochastic attack can be considered as a variation of the template attack in which a template consists of a stochastic power consumption model given by predefined functions and a covariance matrix.

4. DISCUSSION

The Sections 2 and 3 give a general view about two power analysis classes: attacks without a reference device and attacks with a reference device. It is not an easy task to quantitatively compare the performance of these attacks because they are not performed in the same conditions. The first class, which consists of DPA and CPA attacks, does not need any reference device. Their performance is basically evaluated by the number of signals needed to detect the secret key. This number depends on the quality of signals such as the noise level or the misalignment level.

In our experiment, the mono-bit DPA attack needs about 2000 signals, the 4-bit DPA needs about 300 signals, the template attack and the stochastic model attack need only 10 signals to detect the correct key with the ML metric. However, in order to obtain such performance, the second class of attacks has to have a powerful database, which characterizes the power consumption and the noise distribution of the device. Such a powerful database can only be built if

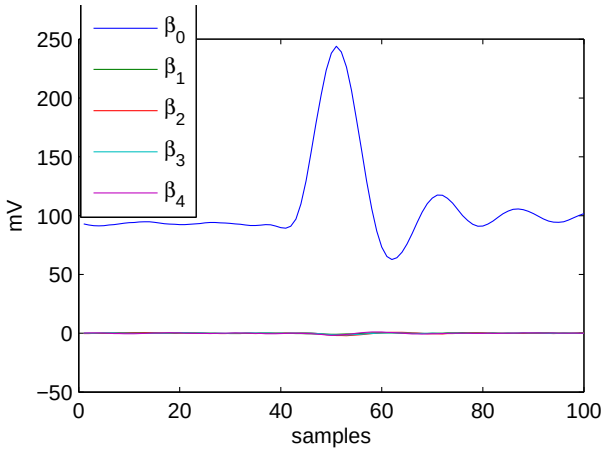

Figure 13: The weights β_n $(n = 0, .., 4)$ in function of time.

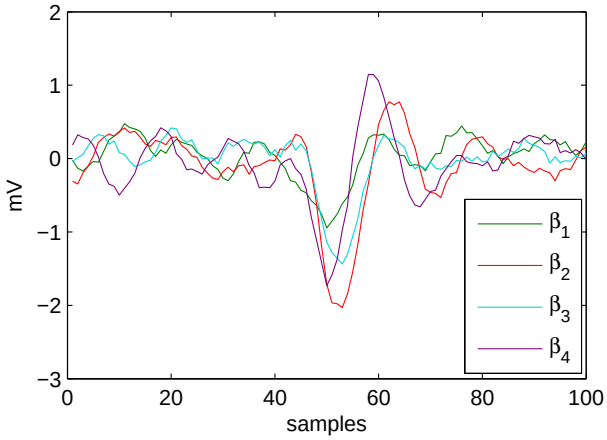

Figure 14: The weights β_1, β_2, β_3,β_4 in function of time (zoom).

the reference device is identical or very close to the attacked device. If this requirement is not fully satisfied, the efficiency of the template and stochastic attacks will be dramatically reduced.

When comparing the template attack and the stochastic model attack, one can easily observe that the second one is much more practical to the first one in the original version (e.g. a template is built for each pair (C_i, K_k)). The detailed comparison of these attacks can be found in [31]. However, if we adopt the bit-value based template and the Hamming weight based template, the template attack becomes more pragmatic and its complexity and its efficiency are equivalent to those of the stochastic method.

Thanks to the use of covariance matrices, the template and stochastic model attacks are well stand up to the noise problem. However, the misalignment of signals is not considered in these methods. If the signal misalignment effect of DPA can be reduced by the sliding window technique [26, 27], this latter, which suppresses also the noise, seems to be difficult to apply in the template and stochastic model attacks.

Following our experiment, the time to inverse covariance matrices that are only used in ML metric is not really long because the number of covariance matrices is small (5 or 16 matrices in our cases) and the number of signals to detect the key is not large. Furthermore, according to the results presented in Sections 2 and 3, the template and stochastic methods work much better with the ML metric than with the MD metric. Therefore, we advise using the maximum likelihood technique to estimate the secret key in the key extraction stage.

The existing points of interest selection strategies presented in the literature aim only at taking the points which have a high dependence with data. However, our results show that not only highly-data-dependent points are useful for the key detection. Among $L = 100$ points around the instant τ, there are only several points which are directly depend on the key and texts but the other points, which in general are not considered, can also contribute to the key detection. The covariance matrices exploit the correlation between points and this correlation is really significant if the considered points are consecutive.

In conclusion, the choice of an attack method depends on the attacker's context. If he has a device which is identical to the attacked one, the template (or stochastic) attack is the best choice. Contrary, if no reference device is available, the attacker should use DPA or CPA attacks. As the DPA signals have a noise level lower than the CPA signals, he can in the first time analyze the signals with DPA to find out significant leakage instants and then apply the CPA to detect the secret key. The DPA and CPA methods can be used for one or several bits, and combined with different signal processing tools to reduce noise and misalignment effects.

5. CONCLUSIONS

The goal of the article is to give readers a deep look into power analysis attacks, particularly the experimental aspect. It begins with an introduction and a state of the art in this domain. The main part of the paper focus on two classes of power analysis: the attacks without reference device such as DPA, CPA and the attacks with a reference device such as the template and the stochastic attacks. We try to present the attacks with a simple manner and a lot of illustration figures based on the same signals.

The paper presents also some advanced analysis to improve existing attacks. Precisely, we develop two conditions which make it possible to improve a multi-bits DPA method. An evaluation of how to enhance the CPA is also given. Different parameters and metrics of the template and stochastic attacks have also been evaluated.

Finally we compare the template/stochastic model attacks and the DPA/CPA attacks. Depend on each situation, one can choose a suitable attack or combine several attacks to efficiently detect the secret key.

6. REFERENCES

[1] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis", *In proceedings of CRYPTO 1999*, LNCS 1666, pp. 388-397, Springer-Verlag, 1999.

[2] R. Anderson and M. Kuhn. Tamper Resistance - a Cautionary Note. In *Proceedings of 2nd USENIX Workshop on Electronic Commerce*, pp. 1–11, Oakland, California, 1996.

[3] H. Handschul, P. Paillier, and J. Stern. Probing Attacks on Tamper Resistant Devices. In *Proceedings of CHES 1999*, pp. 303–315, Massachusetts, USA, 1999. LNCS 1717.

[4] T. S. Messerges, E. A. Dabbish and R. H. Sloan, "Examining Smart-Card Security under the Threat of Power Analysis Attacks", *IEEE Transactions on Computer*, Vol. 51, N5, pp. 541-552, May 2002.

[5] R. Bevan and E. Knudsen, "Ways to Enhance DPA", *In proceedings of ICISC 2002*, LNCS 2587, pp. 327-342, Springer-Verlag, 2003.

[6] T.H. Le, J. Clédière, C. Canovas, C.Servière, J.L. Lacoume and B. Robisson, "A proposition for Correlation Power Analysis enhancement", *In Proceedings of CHES 2006*, LNCS 4249, Springer-Verlag, Yokohama, Japan 2006

[7] T. Messerges, C. Koc, and P. Christof. Using Second-Order Power Analysis to Attack DPA Resistance Software. *In proceedings of CHES 2000*, pp. 238–251, Massachusetts, USA, 2000. LNCS 1965, Springer Verlag.

[8] J. Waddle and D. Wagner, "Towards efficient second-order power analysis", *In Proceedings of CHES 2004*, LNCS 3156, Springer-Verlag, pp. 1-15, Cambridge (Boston), USA 2004.

[9] M. Joye, P. Paillier and B. Schoenmarkers "On Second-Order Differential Power Analysis", *In Proceedings of CHES 2005*, LNCS 3659, Springer-Verlag, pp. 293-308, Edinburgh, Scotland, USA 2005.

[10] S. Chari, C. Jutla, J. Rao, and P. Rohatgi, "A Cautionary Note regarding Evaluation of AES Candidates on Smart-Cards". In *Proceedings of the 2nd Advanced Encryption Standard Candidate Conference*, Rome, Italy, 1999.

[11] R. Mayer-Sommer, "Smartly analysing the simplicity and the power of simple power analysis on smartcards" In *Proceedings of CHES 2000*, pp. 78–92, Massachusetts, USA, 2000. LNCS 1965, Springer Verlag.

[12] J.S. Coron, P. Kocher and D. Naccache, "Statistics and Secret Leakage", *In proceedings of Financial Cryptography*, LNCS 1972, pp 157-173, Springer-Verlag, 2000.

[13] E. Brier, C. Clavier and F. Olivier, "Correlation Power Analysis with a Leakage Model", *In proceedings of CHES 2004* , LNCS 3156, pp. 16-29, Springer-Verlag, 2004.

[14] J.J. Quisquater and D. Samyde, "Electromagnetic Analysis (EMA): Measures and Countermeasures for Smart Cards", *In proceedings of e-Smart 2001*, LNCS 2140, pp. 200-201, Springer, 2001.

[15] K.Gandolfi, C.Mourtel and F.Olivier, "Electromagnetic Analysis: Concrete Results", *In proceeding of CHES 2001* , LNCS 2162, pp. 252-261, Springer, 2001.

[16] H. Li, A.T. Markettos and S. Moore, "Security Evaluation Against Electromagnetic Analysis at Design Time", *In proceedings of CHES 2005*, LNCS 3659, Edinburgh, Scotland 2005

[17] D. Agrawal, J.R. Rao and P. Rohatgi, "Multi-channel Attacks", *In proceedings of CHES 2003*, LNCS 2779, Springer-Verlag , Cologne, Germany 2003.

[18] E. Peeters, F. Standaert, and J. Quisquater. Power and electromagnetic analysis: Improved model, consequences and comparisons. In *INTEGRATION, the VLSI Journal*, volume 40. Elsevier Science.

[19] P. Fahn and P. Pearson. IPA: A New Class of Power Attacks. In *Proceedings of CHES 1999*, LNCS 1717, pp. 173–186, Massachusetts, USA, 1999. Springer-Verlag.

[20] A. Chari, J. Rao, and P. Rohatgi, "Template Attacks", In *Proceedings of CHES02*, LNCS 2523, pp. 13–28, San Francisco Bay, USA, 2002. Springer-Verlag.

[21] W. Schindler, K. Lemke, and C. Paar, "A Stochastic Model for Differential Side Channel Cryptanalysis". In *Proceedings of CHES 2005*, LNCS 3659, pages 30–46, Edinburgh, Scotland 2005

[22] C. Clavier, J. Coron, and N. Dabbous, "Differential power analysis in the presence of hardware countermeasures", In *Proceedings of CHES 2000*, volume LNCS 1965, Massachusetts, USA.

[23] S. Mangard, "Hardware countermeasure against DPA - A statistical analysis of their effectiveness", in *CT-RSA 2004*, Springer, LNCS 2964, 2007, pp. 222–235.

[24] L. Goubin and J. Patarin, "DES and Differential Power Analysis: The Duplication Method". In *Proceedings of CHES 1999*, pp. 158–172, Massachusetts, USA, 1999. LNCS 1717, Springer Verlag.

[25] S. Chari, C.S. Jutla, J.R. Rao and P. Rohatgi, "Towards Sound Approaches to Counteract Power-Analysis Attacks". In *Proceedings of CRYPTO 1999*, pp. 398-412, Santa Barbara, California USA, 1999. LNCS 1666, Springer 1999.

[26] T.H. Le, J. Clédière, C.Servière and J.L. Lacoume, "Higher Order Statistics for Side Channel Analysis Enhancement", *In Proceedings of e-Smart 2006*, Sophia Antipolis, France, September 2006

[27] T.H. Le, J. Clédière, C. Servière and J.L. Lacoume, "Efficient solution for Signal Misalignment of Signal in Side Channel Analysis". In *Proceedings of ICASSP 2007*, Honolulu, Hawaii, USA, April 2007.

[28] T.H. Le, Q.T Nguyen-Vuong, C. Canovas and J. Clédière , "Novel Approaches for Improving the Power Consumption Models in Correlation Analysis". In *Cryptology ePrint Archive*, Available online: http://eprint.iacr.org/2007/306

[29] D. Agrawal, J.R Rao, P. Rohatgi and K. Schramm, "Templates as Master Keys" *In proceedings of CHES 2005*, Springer, Edinburgh, UK, 2005.

[30] C. Rechberger and E. Oswald, "Practical Template Attacks", *In Workshop on Information Security Applications, WISA 2004*, Jeju Island Korea, August 2004.

[31] B. Gierlichs, K. Lemke-Rust and C. Paar, "Template vs. Stochastic Methods", In *Proceedings of CHES 2006*, LNCS 4249, Springer-Verlag, Yokohama, Japan 2006

APPENDIX

A. APPENDIX 1

We assume that the relation between the real power consumption of the device and the Hamming weight (or the Hamming distance) $H(C_i, \mathcal{B}, K)$ (simply denoted by H_i) of $\mathcal{B}$ estimated by the correct key K is represented by a function F. It means that $W(C_i) = F(H_i)$. We calculate the value of the PPA signal corresponding to the correct key at the moment τ, denoted by $\Sigma_k^c(\mathcal{B})$. The signals are distributed in $(d+1)$ groups according to the estimated values H_i, which are the real values. We assume also that $a_{j,k} = a_j$.

$$\begin{aligned}\Sigma_k^c(\mathcal{B}) &= \sum_{j=0}^{d} a_j \frac{\sum_{G_{j,k}} W(C_i)}{N_{j,k}} \\ &= \sum_{j=0}^{d} a_j \frac{\sum_{G_{j,k}} F(H_i)}{N_{j,k}} = \sum_{j=0}^{d} a_j \frac{\sum_{G_{j,k}} F(j)}{N_{j,k}} \\ &= \sum_{j=0}^{d} a_j F(j) \end{aligned} \quad (12)$$

We estimate now the value of the PPA signal corresponding to a wrong key at the moment τ, denoted by $\Sigma_k^w(\mathcal{B})$. As the key is wrong, the estimated values H_i can be different from the real values. Therefore, there exists signals which are distributed in wrong groups. We consider that the bits of $\mathcal{B}$ is uniformly distributed. It means that in group $G_{j,k}$, $(j = 0...d)$, there are $N_{j,k}.C_d^k/2^d$ signals which should belong, *a priori*, in group $G_{l,k}$, $(l = 0...d)$. The $\Sigma_k^w(\mathcal{B})$ is given by:

$$\begin{aligned}\Sigma_k^w(\mathcal{B}) &\approx \sum_{j=0}^{d} a_j \frac{\sum_{G_{j,k}} W(C_i)}{N_{j,k}} \\ &\approx \sum_{j=0}^{d} a_j \frac{\sum_{l=0}^{d} N_{j,k} \frac{C_d^l}{2^d} F(l)}{N_{j,k}} \\ &\approx (\sum_{j=0}^{d} a_j)(\sum_{l=0}^{d} \frac{C_d^l}{2^d} F(l)) \end{aligned} \quad (13)$$

According to Eq. (12) and Eq. (13), we determine conditions of weights a_j to detect efficiently the secret key. There are two principal conditions.

1. The signal $\Sigma_k^w(\mathcal{B})$ is equal to zero:

$$(\sum_{j=0}^{d} a_j)(\sum_{k=0}^{d} \frac{C_d^k}{2^d} F(k)) = 0 \Leftrightarrow \sum_{j=0}^{d} a_j = 0 \quad (14)$$

2. The signal-to-noise ratio (SNR) of the PPA signal corresponding to the correct key is maximized. If the standard deviation of noise of each electromagnetic signal is σ, the standard deviation of noise present in the signal $\Sigma_k^c(\mathcal{B})$ is $\sqrt{\sum_{j=0}^{d} \sigma.a_j^2/N_{j,k}}$. According to Eq. (12), maximizing the SNR of $\Sigma_H^c(\mathcal{B})$ is equivalent to maximizing the ratio:

$$SNR = \frac{|\sum_{j=0}^{d} a_j F(j)|}{\sqrt{\sum_{j=0}^{d} \sigma.a_j^2/N_{j,k}}} \quad (15)$$

B. APPENDIX 2

In Appendix 1, we determine the relation between consumption model F and the weights α_i of the PPA method. In this section, we estimate the optimal weights for a linear model F. As the first condition is independent to F, we are interested in the second condition about the signal-to-noise ratio:

$$SNR = \frac{|\sum_{j=0}^{d} a_j F(j)|}{\sqrt{\sum_{j=0}^{d} \sigma.a_j^2/N_j}} \quad (16)$$

For a linear model, we have $F(j) = a.j + c$ where a and c are constants. The SNR is given by:

$$SNR = \frac{|\sum_{j=0}^{d} a_j (a.j + c)|}{\sqrt{\sum_{j=0}^{d} \sigma.a_j^2/N_j}} \quad (17)$$

Consider the case where $d = 4$ and the texts are uniformly distributed in groups. The number of elements of group G_j is $N_j = C_d^j/2^d N$. Maximizing the SNR becomes maximizing

$$SNR = \frac{|a_0.0 + a_1.1 + a_2.2 + a_3.3 + a_4.4|}{\sqrt{\frac{a_0^2}{1} + \frac{a_1^2}{4} + \frac{a_2^2}{6} + \frac{a_3^2}{4} + \frac{a_4^2}{1}}} \quad (18)$$

As the SNR is always positive, we search the weights which maximize SNR^2. Under the condition $\sum_{i=0}^{d} a_i = 0$, we have $a_0.0+a_1.1+a_2.2+a_3.3+a_4.4 = -2.a_0-a_1+a_3+2.a_4$, and

$$SNR^2 = \frac{(-2.a_0 - a_1 + a_3 + 2.a_4)^2}{\frac{a_0^2}{1} + \frac{a_1^2}{4} + \frac{a_2^2}{6} + \frac{a_3^2}{4} + \frac{a_4^2}{1}} \quad (19)$$

The numerator does not depend on a_2. The SNR^2 is maximized if $a_2 = 0$ and it is equal to:

$$\begin{aligned} SNR^2 &= 4\frac{(-2.a_0 - a_1 + a_3 + 2.a_4)^2}{4.a_0^2 + a_1^2 + a_3^2 + 4.a_4^2} \\ &= 4\frac{(-2.a_0 - a_1 + a_3 + 2.a_4)^2}{(-2.a_0)^2 + (-a_1)^2 + a_3^2 + (-2.a_4)^2} \end{aligned} \quad (20)$$

Denote $x = (-2.a_0)$, $y = (-a_1)$, $z = a_3$, $w = a_4$, we obtain:

$$SNR^2 = 4\frac{(x + y + z + w)^2}{x^2 + y^2 + z^2 + w^2} \quad (21)$$

According to Cauchy-Schwartz inequality, $\forall x, y, z, w \in R$ we have:

$$(x + y + z + w)^2 \leq 4(x^2 + y^2 + z^2 + w^2)$$

In consequence: $SNR^2 \leq 16$. The equality occurs when $x = y = z = w$, or $-2a_0 = a_1 = a_3 = 2a_4$. This relation validates also the condition $\sum_{i=0}^{d} a_i = 0$. Therefore, the maximal value of SNR is obtained if $-2a_0 = a_1 = a_3 = 2a_4$ and $a_2 = 0$ (for example: $a_0 = -1, a_1 = -2, a_2 = 0, a_3 = 2, a_4 = 1$)