Optimal Statistical Power Analysis

Eric Brier, Christophe Clavier, Francis Olivier

Gemplus Card International, France
Security Technology Department
{eric.brier, christophe.clavier, francis.olivier}@gemplus.com

Abstract. A classical model is used for the power consumption of cryptographic devices. It is based on the Hamming distance of the data handled with regard to an unknown but constant reference state. Once validated experimentally it allows an optimal attack to be derived called Correlation Power Analysis. It also explains the defects of former approaches such as Differential Power Analysis.

Keywords: Correlation factor, CPA, DPA, Hamming distance, power analysis, DES, AES, secure cryptographic device, side channel.

1 Introduction

In the scope of statistical power analysis against cryptographic devices, two historical trends can be observed. The first one is the well known differential power analysis (DPA) introduced by Paul Kocher [2, 3] and formalized by Thomas Messerges et al. [4]. The second one has been suggested in various papers [6, 10] and proposed to use the correlation factor between the power samples and the Hamming weight of the handled data. Both approaches exhibit some limitations due to unrealistic assumptions and model imperfections that will be examined more thoroughly in this paper. Some authors have unsuccessfully tempted to improve the Hamming weight model [12] or to enhance the DPA itself [7, 19] by various means.

The proposed approach is based on the Hamming distance model which can be seen as a generalization of the Hamming weight model. Surprisingly all its basic assumptions were already mentioned in various papers from year 2000 [7, 6, 12]. This synthetic and experimental work that has been conducted since then, shows how a simple extension of those former elements can explain many defects of DPA. Furthermore, a new analysis method is derived by introducing optimality criteria that were missing up to now. Classical statistics show that the correlation power analysis (CPA) with the proposed model is the most relevant indicator to validate the model and conduct efficient and sometimes optimal attacks against unprotected implementations of many algorithms such as DES or AES. This study deliberately restricts itself to the scope of secret key cryptography although it may be extended beyond.

This paper is organized as follows: Section 2 introduces the Hamming distance model and Section 3 proves the relevance of the correlation factor. The

model based correlation attack is described in Section 4 with the impact on the model errors. Section 5 addresses the estimation problem and the experimental results which validate the model are exposed in Section 6. Section 7 contains the comparative study with DPA and addresses more specifically the so-called "ghost peaks" problem encountered by those who have to deal with erroneous conclusions when implementing classical DPA on the substitution boxes of the DES first round: it is shown there how the proposed model explains many defects of the DPA and how the correlation power analysis can help in conducting sound attacks in optimal conditions.

2 The Hamming distance consumption model

Classically, most power analyses found in literature are based upon the Hamming weight model [3, 4], that is the number of bits set in a data word. In a m-bit microprocessor, binary data is coded $D = \sum_{j=0}^{m-1} d_j 2^j$, with the bit values $d_j = 0$ or 1. Its Hamming weight is simply the number of bits set to 1, $H(D) = \sum_{j=0}^{m-1} d_j$. Trivially, the Hamming weight is not linear w. r. t. D and its integer values stand between 0 and m. If D is a uniformly distributed random variable, each independent bit d_j has a probability $1/2$ to be 0 and $1/2$ to be 1. Therefore each bit has an average value of $1/2$ and a variance of $1/4$. As sum of m independent variables the whole word has an average Hamming weight $\mu_H = m/2$ and a variance $\sigma_H^2 = m/4$.

Some papers [12] have tackled the problem of consumption modeling and proposed different laws to represent the information leakage. Usually they conclude that their model is incomplete except in restricted situations or on specific components. Nevertheless, it is generally assumed that at a given time the data leakage through the power side-channel depends on the number of bits switching from one state to the other [7, 6]. Indeed one can model a microprocessor as a state-machine where transitions from state to state are triggered, in this instance, by a clock signal. This seems relevant when looking at a logical elementary gate as implemented in CMOS technology. The current consumed is related to the energy required to flip the bits from one state to the next. It is composed of two main contributions: the capacitor's charge and the short circuit induced by the gate transition. Curiously, this elementary behavior is commonly admitted but has never given rise to any satisfactory model that is widely applicable. Of course, hardware designers use tools to simulate the current consumption of their designs but they usually keep the underlying model as a trade secret; complete representative simulations are difficult to obtain anyway.

If the transition model is adopted, a basic question is posed. What is the reference state from which the bits are switched?

We assume here that this reference state is a constant machine word, R, which is unknown, but not necessarily zero. It will always be the same if the same data manipulation always occurs at the same time, although this assumes the absence of any desynchronizing effect. In addition, two restrictive hypotheses are put forward:

– it is assumed that switching a bit from 0 to 1 requires the same amount of energy as switching it from 1 to 0.
– all the machine bits are assumed to be perfectly balanced and require the same amount of energy to switch on and off.

These assumptions may appear very restrictive (and explain some imperfections of the next results). But they are affordable without any thorough (and almost inaccessible) knowledge of microelectronic devices. Moreover they lead to a convenient expression for the model. Indeed the number of flipping bits to go from R to D is described by $H(D \oplus R)$ also called the Hamming distance between D and R. This statement encloses the Hamming weight model which assumes that $R = 0$. If D is a uniform random variable, so is $D \oplus R$, and $H(D \oplus R)$ has the same mean $m/2$ and variance $m/4$ as $H(D)$.

Another assumption is that the relationship between the current consumption and $H(D \oplus R)$ is linear. This can also be seen as a limitation but considering a chip as a large set of elementary electrical components, this linear model fits reality quite well. It does not represent the entire consumption of a chip but only the data dependent part. This does not seem unrealistic when considering, for instance, that the bus lines are usually considered as the most consuming elements within a micro-controller. In this simple vision all the remaining things in the power consumption of a chip are assigned to a noise term denoted b and assumed independent from the other variables; its mean value is not necessarily zero and can be assigned to an offset component. Therefore the basic model for the data dependency can be written:

$$W = aH(D \oplus R) + b$$

where a is a scalar gain between the Hamming distance and W the power consumed.

Before the relevance of the model is experimentally justified, we are going to discuss some mathematical points based on classical statistics.

3 The linear correlation factor

If the linear Hamming distance model is assumed to be valid, there will exist relationships between the variances of the different terms considered as random variables:

$$\sigma_W^2 = a^2\sigma_H^2 + \sigma_b^2$$

Writing the noise as $b = W - aH(D \oplus R)$, its variance can also be expressed with the following formula:

$$\sigma_b^2 = \sigma_W^2 + a^2\sigma_H^2 - 2a\mathrm{cov}(W, H) = \sigma_W^2 + a^2\sigma_H^2 - 2a\sigma_W\sigma_H\rho_{WH}$$

where ρ_{WH} denotes the correlation factor between the Hamming distance and the power. It is the covariance between both random variables normalized by the

product of their standard deviations. Under the uncorrelated noise assumption, this definition leads to:

$$\rho_{WH} = \frac{\mathrm{cov}(W,H)}{\sigma_W \sigma_H} = \frac{a\sigma_H}{\sigma_W} = \frac{a\sigma_H}{\sqrt{a^2\sigma_H^2 + \sigma_b^2}} = \frac{a\sqrt{m}}{\sqrt{ma^2 + 4\sigma_b^2}}$$

This equation complies with the well known property: $-1 \leq \rho_{WH} \leq +1$: for a perfect model the correlation factor tends to ± 1 if the variance of noise tends to 0, the sign depending on the sign of the linear gain a. The signal to noise ratio of W can be expressed in terms of the correlation factor as:

$$SNR(W) = \frac{a\sigma_H}{\sigma_b} = \frac{a\sqrt{m}}{2\sigma_b} = \frac{\rho_{WH}}{\sqrt{1-\rho_{WH}^2}}$$

Let's notice that if the model applies only to l independent bits amongst m, a partial correlation still exists:

$$\rho_{WH_{l/m}} = \frac{a\sqrt{l}}{\sqrt{ma^2 + 4\sigma_b^2}} = \rho_{WH}\sqrt{\frac{l}{m}}$$

4 Secret inference based on correlation power analysis

The relationships written above show that if the model is valid the correlation factor is maximized when the noise variance is minimum. This means that ρ_{WH} can help to determine the reference state R. Assume a set of known random data D and a set of related power consumption W are available. If the 2^m possible values of R are scanned exhaustively they can be ranked by the correlation factor they produce when combined with the observation W. This is not that expensive when considering an 8-bit micro-controller, the case with many of today's smart cards, as only 256 values are to be tested. On 16-bit or 32-bit architectures the same exhaustive search cannot be applied as such. However it is still possible to work by introducing some special tricks like prior knowledge or partial correlation.

Let R' represent the candidate values and H' the related model for the Hamming distance with the data $H' = H(D \oplus R')$. Knowing that R is the right value and $H = H(D \oplus R)$ the right prediction on the Hamming distance, the correlation test leads to:

$$\rho_{WH'} = \frac{\mathrm{cov}(aH+b,H')}{\sigma_W \sigma'_H} = \frac{a}{\sigma_W}\frac{\mathrm{cov}(H,H')}{\sigma'_H} = \rho_{WH}\rho_{HH'}$$

as H' and b are independent, $\mathrm{cov}(H',b) = 0$. Under this realistic condition $|\rho_{WH'}| \leq |\rho_{WH}|$ since the correlation factors are normalized. No other value than the correct R can produce higher correlation rate.

Now the issue of uniqueness is to be addressed. If the model is valid are there other values $R' \neq R$ that could result in an equivalent correlation factor?

Assume a value of R' that has k bits that differ from those of R, then: $H(R \oplus R') = k$. When both are XORed with the random D, one adopts the following formulation for the Hamming distances:

$$\begin{aligned} H &= H_{m-k} + H_k \\ H' &= H'_{m-k} + H'_k = H_{m-k} - H_k + k \end{aligned}$$

where H_{m-k} denotes the common part ($m-k$ bits large) of $R \oplus D$ and $R' \oplus D$. The different part has a complementary Hamming weight and so $H'_k = k - H_k$. Since k does not vary we have the following equality:

$$\mathrm{cov}(H, H') = \mathrm{cov}(H, H' - k)$$

So using the bracket notation for mathematical expectations this can be developed into:

$$\begin{aligned} \mathrm{cov}(H, H') &= \langle (H_{m-k} + H_k)(H_{m-k} - H_k) \rangle - \langle H_{m-k} + H_k \rangle \langle H_{m-k} - H_k \rangle \\ \mathrm{cov}(H, H') &= \langle H^2_{m-k} \rangle - \langle H^2_k \rangle - \langle H_{m-k} \rangle^2 + \langle H_k \rangle^2 \end{aligned}$$

Replacing the terms by their respective values (see Section 2) gives:

$$\begin{array}{llll} \langle H_{m-k} \rangle = \frac{m-k}{2} & \text{and } \langle H^2_{m-k} \rangle = \sigma^2_{H_{m-k}} + \langle H_{m-k} \rangle^2 & = \frac{m-k}{4} + \frac{(m-k)^2}{4} \\ \langle H_k \rangle = \frac{k}{2} & \text{and } \langle H^2_k \rangle \quad = \sigma^2_{H_k} + \langle H_k \rangle^2 & = \frac{k}{4} + \frac{k^2}{4} \end{array}$$

leading to the following expression of the covariance:

$$\mathrm{cov}(H, H') = \frac{m - 2k}{4}$$

Therefore, since $\sigma_H = \sigma_{H'} = m/4$, we finally obtain:

$$\rho_{HH'} = \frac{\mathrm{cov}(H, H')}{\sigma_H \sigma_{H'}} = \frac{m - 2k}{m}$$

This expression shows how the correlation factor is capable of rejecting wrong values of R. For instance, if a single bit is wrong amongst an 8-bit word, the correlation is reduced by $1/4$. If all the bits are wrong, i-e $R' = \neg R$, then an anti-correlation should be observed with $\rho_{WH'} = -\rho_{WH}$ (This property can be used to reduce the exhaustive search on R to 2^{m-1} values). In absolute value or if the linear gain is assumed positive ($a > 0$), there cannot be any R' leading to a higher correlation rate than R.

This proves the uniqueness of the solution and therefore how the reference state can be determined. This analysis can be performed on the power trace assigned to a piece of code while manipulating known but randomly varying data. If we assume that the handled data is the result of a XOR operation between a secret key word K and a known message word M, the procedure described above, i-e exhaustive search on R and correlation test, should lead to $K \oplus R$ associated with $\max(\rho_{WH})$. Indeed if a correlation occurs when M

is handled with respect to R_1, another has to occur later on, when $M \oplus K$ is manipulated in turn, possibly with a different reference state R_2 (in fact with $K \oplus R_2$ since only M is known).

For instance, when considering the first *AddRoundKey* function at the beginning of the AES algorithm embedded on an 8-bit processor, it is obvious that such a method leads to the whole key masked by the constant reference byte R_2. If R_2 is the same for all the key bytes, which is highly plausible, only 2^8 possibilities remain to be tested by exhaustive search to infer the entire key material. This complementary brute force may be avoided if R_2 is determined by other means or known to be always equal to 0 (on certain chips).

This attack is not restricted to the $\oplus$ operation. It also applies to many other operators often encountered in secret key cryptography. For instance, other arithmetic, logical operations or look-up tables (LUT) can be treated in the same manner by using $\mathrm{H}(\mathrm{LUT}(M \star K) \oplus R)$, where $\star$ represents the involved function i.e. $\oplus$, +, -, OR, AND, or whatever operation. Let's notice that the ambiguity between K and $K \oplus R$ is completely removed by the substitution boxes encountered in secret key algorithms thanks to the non-linearity of the corresponding LUT: this may require to exhaust both K and R, but only once for R in most cases.

To conduct the analysis in the best condition, it is important to correctly model the whole machine word that is actually handled and its transition with respect to the reference state R which is to be determined as an unknown of the problem.

5 Estimation

In a real case with a set of N power curves W_i and N associated random data words M_i, for a given reference state R the known data words produce a set of N predicted Hamming distances $H_{i,R} = H(M_i \oplus R)$. The corresponding correlation factor can be estimated by the following formula:

$$\hat{\rho}_{WH}(R) = \frac{N \sum W_i H_{i,R} - \sum W_i \sum H_{i,R}}{\sqrt{N \sum W_i^2 - (\sum W_i)^2} \sqrt{N \sum H_{i,R}^2 - (\sum H_{i,R})^2}}$$

where the summations are taken over the N samples ($i = 1, N$). One reminds that such an estimation has to be done at each time slot within the power traces $W_i(t)$.

The issue concerning the estimation of ρ is not trivial at all. It is theoretically difficult to study the variance of the estimator $\hat{\rho}$ with respect to the number of available samples N. Statistical literature (such as [1]) states that, depending on the true value of ρ, some transforms of $\hat{\rho}$ have statistical behaviors that can be approximately modeled (in law, mean value and variance) provided the number of samples is moderately large (say a few dozens). $\hat{\rho}$ is very slightly biased and this bias tends asymptotically to zero with increasing N. Briefly, for ordinary noise levels, a few hundred experiments should suffice to provide a good

estimate of the true correlation factor. Next results will show that this is more than necessary for conducting reliable tests.

Exhausting R and taking the maximum value $\rho_{max} = \max(\hat{\rho}_{WH}(R))$ can be seen as a model fitting procedure. It becomes meaningful if significant values of the correlation estimate are achieved, typically values higher than 50% with a substantial set of experiments. In such a case the existence of a linear relationship between W and H is likely. According to the linear regression theory, a least square estimate of the gain is given by:

$$\hat{a} = \rho_{max} \frac{\sigma_W}{\sigma_H} = \frac{\mathrm{cov}(W, H)}{\sigma_H^2}$$

which estimates the sensitivity of the measurement with respect to the model. It is not actually useful since it does not help in finding the optimal R. The next formula provides an estimate of the noise variance:

$$\hat{\sigma}_b^2 = (1 - \rho_{max}^2)\sigma_W^2$$

In very noisy conditions, it is often possible to perform averages of several curves with same data before estimating the correlation factor with varying data. This reduces the variance of noise in the same proportion.

The previous formula shows that ρ_{max} is an estimate that minimizes the noise variance or equivalently the probability $p(W = aH(D \oplus R) + \mu_b)$. In other words the correlation factor which is usually referred to as a linearity test between two random variables has turned into a posterior maximum likelihood test for the Hamming distance model with respect to the reference state R.

6 Experimental results

So far the interest of the model based correlation analysis has been attested only according to theoretical considerations. It must be confronted to many experiences to be justified in practice. This section presents some experimental results and synthesizes the general rules of behavior derived from the analysis during the passed years of various chips for secure devices.

Our first experience was performed onto a basic XOR algorithm implemented in an old-fashioned 8-bit chip. The sequence of instructions was simply the following:

- load a byte D_1 into the accumulator
- XOR D_1 with a constant D_2
- store the result from the accumulator to a destination memory cell.

The program was executed 256 times with D_1 varying from 0 to 255. As displayed on Figure 1, two significant correlation peaks were obtained with two different reference states:

- the first one being the address of D_1,

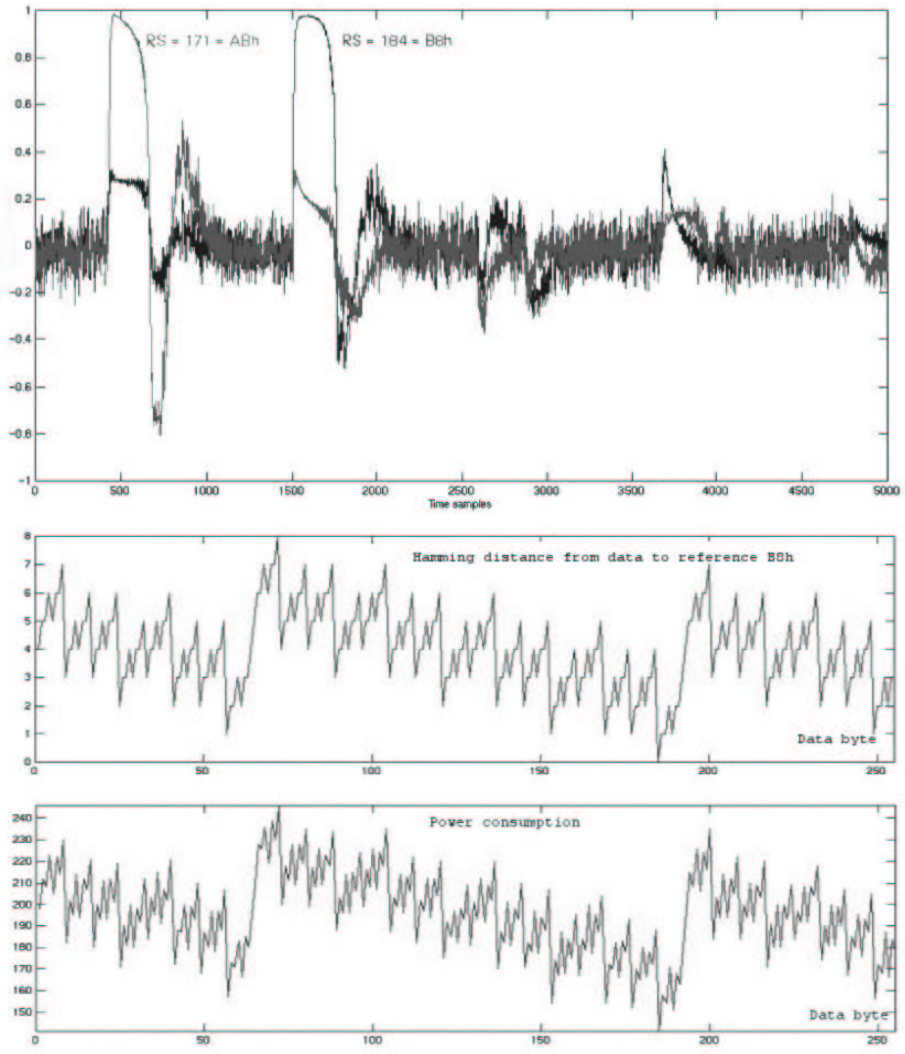

Fig. 1. Upper: consecutive correlation peaks for two different reference states. Lower: for varying data (0-255), model array and measurement array taken at the time of the second correlation peak.

– the second one being the opcode of the XOR instruction

This result typically illustrates the most general case because it reveals the sequence of transfers on a common bus. The address of a data word is transmitted just before its value that is in turn immediately followed by the opcode of the next instruction which is fetched.

This behavior can be observed on a wide variety of chips even those implementing 16- or 32-bit architectures. Correlation rates ranging from 60% to more than 90% can often be obtained. Figure 2 shows an example of partial correlation on a 32-bit architecture: when only 4 bits are predicted among 32, the correlation loss is in about the ratio $\sqrt{8}$ which is consistent with the observed result.

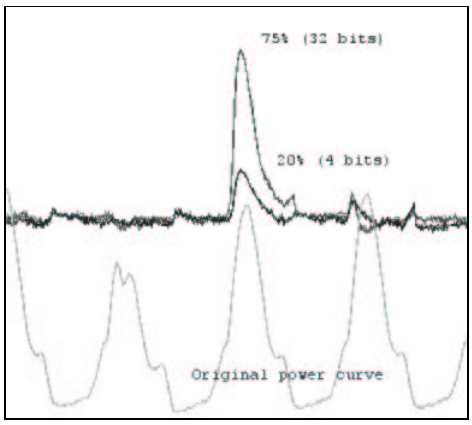

Fig. 2. Two correlation peaks for full word (32 bits) and partial (4 bits) predictions. According to theory the 20% peak should rather be around 26%.

Figures 3 and 4 illustrate the estimation problem of the correlation factor with regard to the population size N.

Figure 3 shows that it requires a certain amount of curves to reduce the ambiant variance in the vicinity of a true correlation peak and prevent from generating false peaks that may lead to wrong interpretation. However those spurious peaks are temporally well defined and easy to recognize as they look more like a random noise or clock edges than like classical peaks.

The graphs presented in Figure 4 shows that the estimation of the maximum value greatly depends on the population size N. They have been constructed from a set of 1000 power traces recorded during the manipulation of a 32-bit random data leaking according to its Hamming weight. The first graph shows how the estimator converges to a given value (here around 75%) when N is increasing in a dependent mode (taking the 10 first samples, then the 20 first ones and so on, up to 1000). In this example, the curve looks chaotic in the beginning and reaches a minimum. Then beyond $N = 100$ it becomes smoother and goes in a monotone manner up to an asymptotic value.

The second one is derived from independent subsets. For instance taking the 1000 samples by packs of 100 (1 to 100, 101 to 200 and so on) we can calculate up to 10 estimates with $N = 100$. This approach has been applied to $N = 50$, 100, 200, 300 and 500; the observed dispersion clearly proves how

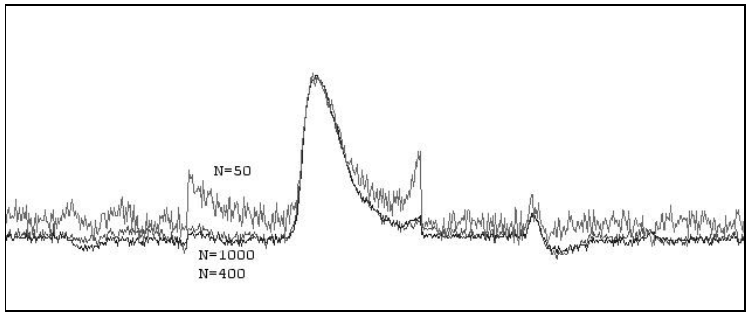

Fig. 3. Correlation curves for 3 values of N.

erroneous the asymptotic value can be. For different populations of a given size (say below 1000), the obtained value can vary with higher magnitude of 10 to 20%. We have deliberately chosen to illustrate the purpose with a 32-bit example that represents the worst case. In an 8-bit example this dispersion is much less important: it rarely exceeds 5% but one has to be aware of it. Anyway some results presented further will show that the estimation issue is not critical to conduct attacks in good conditions.

These results can be observed in many situations involving various technologies and implementations. Nevertheless the following restrictions have to be mentioned:

- Sometimes the reference state is systematically 0 whatever the algorithm is. This can be assigned to the so-called pre-charged logic where the bus is cleared between each significant transferred value. Another possible reason is that complex architectures implement separated busses for data and addresses, that may prohibit certain transitions. In all those cases the Hamming weight model is recovered as a particular case of the more general Hamming distance model.
- The sequence of correlation peaks may sometimes be blurred or spread over the time in presence of a pipe line.
- Some recent technologies implement hardware security features designed to impede statistical power analysis. These countermeasures offer various levels of efficiencies going from the most naive and easy to bypass, to the most effective which merely cancel any data dependency.

There are different kinds of countermeasures which are completely similar to those designed against DPA.

- Some of them consist in introducing desynchronization in the execution of the process so that the curves are not aligned anymore within a same acquisition set. For that purpose there exist various techniques such as fake cycles insertion, unstable clocking or random delays. In many cases their effect can be corrected by applying appropriate signal processing.

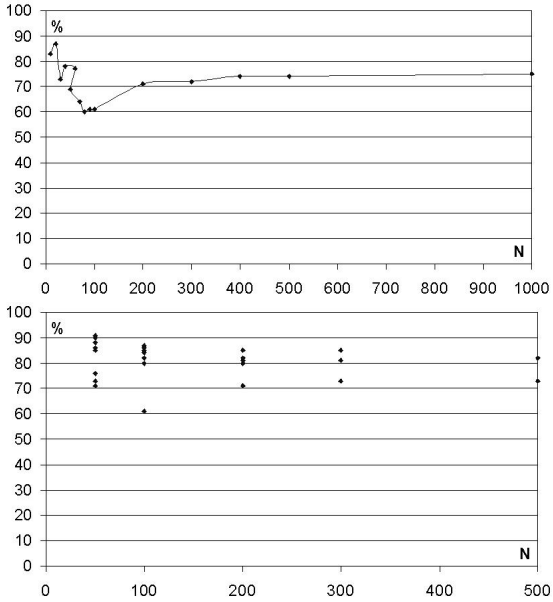

Fig. 4. Estimation of $\max(\rho)$. Upper: convergence versus N for dependent subsets. Lower: various estimates for independent subsets of given sizes.

- Other countermeasures consist in blurring the power traces with additional noise or filtering circuitry [8]. Sometimes they can be bypassed by curves selection and/or averaging or by using another side channel such as electromagnetic radiation [13, 15].
- The data can also be ciphered dynamically during a process by hardware (such as bus encryption) or software means (data masking with a random [5, 11, 17, 18]), so that the handled variables become unpredictable: then no correlation can be expected anymore. In theory sophisticated attacks such as higher order analysis [9] can overcome the data masking method; but they are easy to thwart in practise by using desynchronization for instance.

Indeed, if implemented alone, none of these countermeasures can be considered as absolutely secure against statistical analyses. They just increase the amount of effort and level of expertise required to achieve an attack. However combined defenses, implementing at least two of these countermeasures, prove to be very efficient and practically dissuasive. The state of the art of countermeasures in the design of tamper resistant devices has made big advances in the recent years. It is now admitted that security requirements include sound implementations as much as robust cryptographic schemes.

7 Comparison with DPA

This section addresses the comparison of the proposed CPA method with Differential Power Analysis (DPA). It refers to the former works done by Messerges et al. [4, 16] who formalized the ideas previously suggested by Kocher [2, 3].

7.1 Variance of the DPA bias

In single bit DPA the following bias is computed:

$$T = \frac{\sum_i W_i d_j}{\sum_i d_j} - \frac{\sum_i W_i (1 - d_j)}{\sum_i (1 - d_j)}$$

This indicator is the difference between the averages of two subsets of power curves. They result from partitioning the initial population according to a special selection bit d_j within a data on which predictions are done: this is why d_j is also called "targeted bit". Without loss of generality, both subsets can be assumed as perfectly balanced: $\sum_i d_j = \sum_i (1 - d_j) = N/2$. If 1 is the predicted value for the target bit d_j the corresponding power trace is assigned to the first pack, and to the second pack if its predicted value is 0. T is expected to be maximized when the prediction is right so that a peak is erected within the resulting T signal at the instant when the data is manipulated in a consistent manner. Anywhere else the resulting bias signal is expected to produce only noise since the prediction is not consistent with the measurements. As distance between means, T is a test for two hypotheses with supposed equivalent variances.

If the consumption model is injected in the expression of T and if the bits others than the selection bit are assumed independent of it and uniformly distributed, the following statistical properties can easily be stated:

- The pack of words with $d_j = 0$ has an average weight of $\frac{m-1}{2}$ and a variance of $\frac{m-1}{4}$.
- The pack of words with $d_j = 1$ has an average weight of $\frac{m+1}{2}$ and the same variance of $\frac{m-1}{4}$.
- As a consequence the average of T is $\mu_T = a$.
- Its variance is $\sigma_T^2 = \langle T^2 \rangle - \langle T \rangle^2$ that can be developed into:

$$\sigma_T^2 = \frac{(m-1)a^2 + 4\sigma_b^2}{N}$$

This expression stems from the basic linear properties of mathematical expectations of combined independent variables:

$$\langle xy \rangle = \langle x \rangle \langle y \rangle \text{ and } \langle (\Sigma x_i)^2 \rangle = \sigma_{\Sigma x_i}^2 + (\Sigma \langle x_i \rangle)^2 = n\sigma_x^2 + n^2 \langle x \rangle^2$$

Moreover it is worth noticing that it does not depend on the reference state R due to the statistical properties of the Hamming distance model (see Section 2). Finally the signal to noise ratio in the bias signal T is:

$$SNR(T) = \frac{\langle T \rangle}{\sigma_T} = \frac{a\sqrt{N}}{\sqrt{(m-1)a^2 + 4\sigma_b^2}}$$

This formula is similar to the one established by Messerges et al. [4, 16], except that we have deliberately omitted here what they call algorithmic noise. They define this additive noise as the data dependent variance of the power signal taken when the predicted variable is not handled. Its influence makes the expression of what they refer to as intra-signal SNR slightly more complicated. It is not the purpose of this section to discuss that point. It just aims at observing that in the denominator the number $m-1$ of unpredicted bits can be seen as a penalty that must be compensated by increasing the amount of samples N. It came up to the mind of Messerges et al. to improve things by enlarging the prediction to several bits, say l, and increase the SNR of the multi-bits DPA bias:

$$SNR(T) = \frac{la\sqrt{N}}{\sqrt{(m-l)a^2 + 4\sigma_b^2}}$$

This enhancement is effective but not really interesting on a practical point of view. When dealing with multi-bits DPA, only experiments associated with all zeros and all ones predictions are combined. All the others are not used (say 01 and 10 in a 2-bit case). This forces to record $2^{l-1}N$ experiments on average in order to keep N useful curves for each prediction. In summary what is gained thanks to l is partly lost because of N. Finally multi-bits DPA may be interpreted as a tricky extension of the single bit DPA to improve the SNR in an effective but not actually efficient manner. Indeed nothing proves the relevance of the SNR as optimality criterion.

7.2 Reminder on the DPA attack against the DES

As a matter of fact during the process of making a decision, DPA presents some strange defects that cannot be assigned to the SNR. To develop this idea we are going to address the specific issue of the attack against the DES.

Basically the substitution boxes at the first round of the algorithm can be aimed at in order to retrieve the 8 sub-keys involved at this stage. The sub-keys are 6-bit values; so in principle 48 bits of the key (56 bits) can be inferred this way, that is to say almost all the secret. The remaining part (8 bits) can be recovered either by using the gleaned information to attack the second round or by brute force cryptanalysis.

The method consists in guessing the sub-keys at the input of the substitutions and predict their output $\mathrm{SBox}(D \oplus K)$, where:

- D denotes the known input data (6 bits) derived through several permutations (IP, EP) from the clear text which is chosen randomly.
- K denotes the sub-key (6 bits) at the input of a substitution box. Its knowledge reveals directly 6 bits of the key by inverting the key schedule operation.
- SBox stands for any substitution table amongst the 8 used in the DES. The input is 6 bits large (values from 0 to 63) whereas the output is only 4 bits large (values from 0 to 15).

The brilliant idea behind this procedure is that the right prediction of a single output bit is capable of disclosing 6 bits of the input sub-key thanks to the non-linearity of the substitution table. Predicting a bit of $D \oplus K$ is also possible but less efficient since one bit is sensitive to only one sub-key bit (moreover the interpretation is not that obvious). For each substitution the 64 possible sub-keys are guessed exhaustively and each related prediction validated by the maximization of the corresponding DPA signal. In the end performing 8×64 DPA tests like this should lead to the 48 secret bits involved.

In fact such an attack works quite well only if the following assumptions are fulfilled:

1. Word space assumption: within the predicted word, the contribution of the non-targeted bits is independent of the targeted bit value. Their average influence in the curves pack of 0 is the same as that in the curves pack of 1. So the attacker does not need to care about these bits.
2. Guess space assumption: the predicted value of the targeted bit for any wrong guess does not depend on the value associated to the correct guess.
3. Time space assumption: the power consumption W does not depend on the value of the targeted bit except when it is explicitly handled.

7.3 The "ghost peaks" problem

When confronted to the experience, the DPA attack described above comes up against the following facts.

- *Fact A.* For the correct guess, DPA peaks appear also when the targeted bit is not explicitly handled. This is worth being noticed albeit not really embarrassing. However this contradicts the third assumption.
- *Fact B.* Some DPA peaks also appear for wrong guesses: they are called "ghost peaks". This fact is more problematic for making a sound decision and comes in contradiction with the second assumption.
- *Fact C.* The true DPA peak given by the right guess may be smaller than some ghost peaks, and even null or negative! This seems somewhat amazing and quite confusing for an attacker. The reasons must be searched for inside the crudeness of the optimistic first assumption.

7.4 The "ghost peaks" explanation

With the help of a thorough analysis of substitution boxes and the Hamming distance model it is now possible to explain the observed facts and show how wrong the basic assumptions of DPA can be.

- *Fact A.* As a matter of fact some data handled along the algorithm may be partially correlated with the targeted bit. This is not that surprising when looking at the structure of the DES. A bit taken from the output nibble of a SBox has a lifetime lasting at least until the end of the round (and beyond

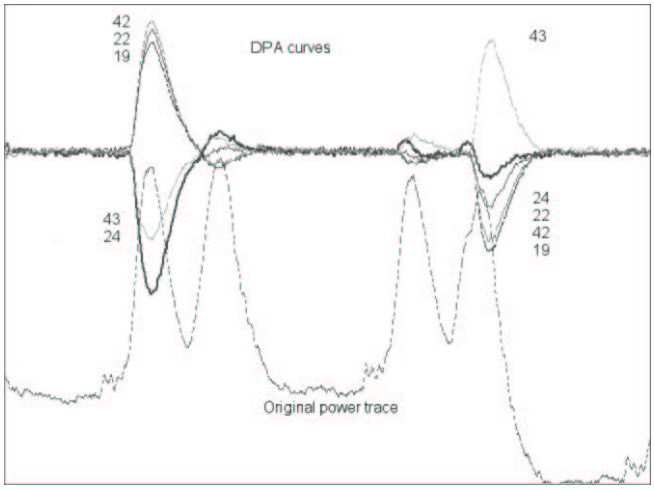

Fig. 5. DPA curves on bit 1 of SBox_1 for various guesses (right guess is 24).

if the left part of the IP output does not vary too much). A DPA peak rises each time this bit and its 3 peer bits undergo the following P permutation since they all belong to the same machine word.

– *Fact B.* The reason why wrong guesses may generate DPA peaks is that the distributions of an SBox output bit for two different guesses are deterministic and so possibly partially correlated. The following example is very convincing about that point. Let's consider the leftmost bit of the fifth SBox of the DES when the input data D varies from 0 to 63 and combined with two different sub-keys : $\mathrm{MSB}(\mathrm{SBox}_5(D \oplus 0x00))$ and $\mathrm{MSB}(\mathrm{SBox}_5(D \oplus 0x36))$. Both series of bits are respectively listed hereafter, with their bitwise XOR on the third line:

```
1101101010010110001001011001001110101001011011010101001000101101
1001101011010110001001011101001010101101011010010101001000111001
0100000001000000000000000100000100000100000001000000000000010100
```

The third line contains 8 set bits, revealing only eight errors of prediction among 64. This example shows that a wrong guess, say 0, can provide a good prediction at a rate of 56/64, that is not that far from the correct one $0x36$. The result would be equivalent for any other pair of sub-keys K and $K \oplus 0x36$. Consequently a substantial concurrent DPA peak will appear at the same location than the right one. The weakness of the contrast will disturb the guesses ranking especially in presence of high SNR.

– *Fact C.* DPA implicitly considers the word bits carried along with the targeted bit as uniformly distributed and independent from the targeted one. This is erroneous because implementation introduces a deterministic link between their values. Their asymmetric contribution may affect the height

and sign of a DPA peak. This may influence the analysis on the one hand by shrinking relevant peaks, on the other hand by enhancing meaningless ones. There exists a well known trick to bypass this difficulty as mentioned in [19]. It consists in shifting the DPA attacks a little bit further in the processing and perform the prediction at the end of the first round of the DES instead of just at the output of the substitution boxes. At this stage the right part of the data (32 bits) is XORed with the left part of the IP output. If the message can be chosen freely, this represents an opportunity to re-balance the loss of randomness by bringing new refreshed random data. But this does not fix *Fact B* in a general case .

In order to get rid of the above-mentioned ambiguities it is more convenient although challenging to try to predict the entire handled word and to take the whole information into account in order to conduct an attack in optimal conditions. This requires to introduce the notion of algorithmic implementation that DPA assumptions completely occult.

When considering the substitution boxes of the DES, it cannot be avoided to remind that the output values are 4-bit values. Although these 4 bits are in principle equivalent as DPA selection bits, they live together with 4 other bits in the context of an 8-bit microprocessor. Efficient implementations use to exploit those 4 bits to save some storage space in constrained environments like smart card chips.

A trick referred to as "SBox compression" consists in storing 2 SBox values within a same byte. Thus the required space is halved. There are different ways to implement this. Let's consider for instance the 2 first boxes: instead of allocating 2 different arrays, it is more efficient to build up the following look-up table: $\mathrm{LUT}_{12}(k) = \mathrm{SBox}_1(k) \parallel \mathrm{SBox}_2(k)$. For a given input index k, the array byte contains the values of two neighboring boxes. Then according to the Hamming distance consumption model, the power trace should vary like:

- $\mathrm{H}(\mathrm{LUT}_{12}(D_1 \oplus K_1) \oplus R_1)$ when computing SBox_1.
- $\mathrm{H}(\mathrm{LUT}_{12}(D_2 \oplus K_2) \oplus R_2)$ when computing SBox_2.

If the values are bind like this, their respective bits cannot be considered as independent anymore. To prove this assertion we have conducted an experiment on a real 8-bit implementation that was not protected by any DPA countermeasures. Working in a "white box" mode, the model parameters had been previously calibrated with respect to the measured consumption traces. The reference state $R = 0xB7$ had been identified as the Opcode of an instruction transferring the content of the accumulator to RAM using direct addressing. The model fitted the experimental data samples quite well; their correlation factor even reached 97%. So we were able to simulate the real consumption of the Sbox output with a high accuracy. Then the study consisted in applying a classical single bit DPA to the output of SBox_1 in parallel on both sets of 200 data samples: the measured and the simulated power consumptions.

As the next figure shows, the simulated and experimental DPA biases match particularly well. One can notice the following points:

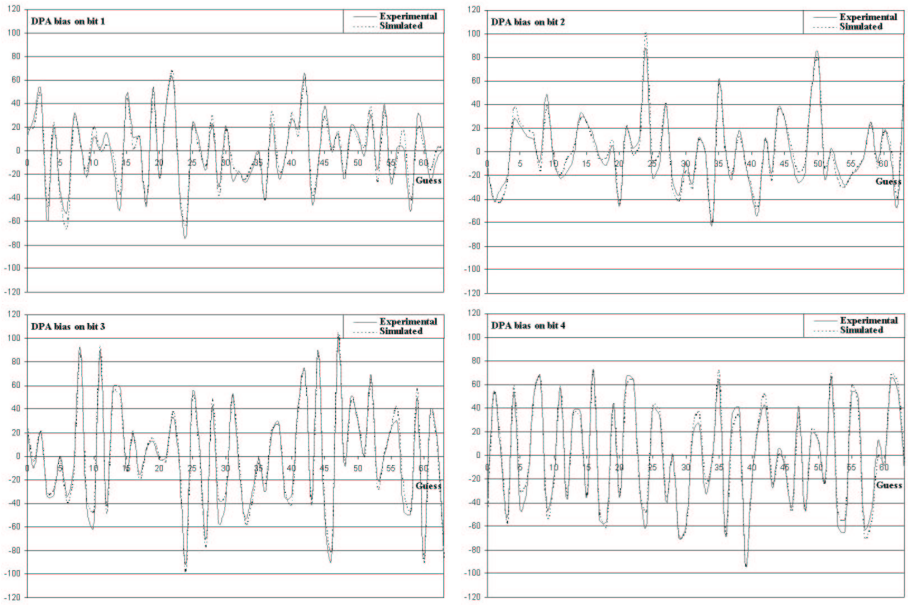

Fig. 6. DPA biases on SBox_1 versus guesses for selection bits 1, 2, 3 and 4, on modeled and experimental data; the correct guess is 24.

- The 4 output bits are far from being equivalent.
- The polarity of the peak associated to the correct guess 24 depends on the polarity of the reference state. As $R = 0xB7$ its leftmost nibble aligned with SBox_1 is $0xB$ = '1011' and only the selection bit 2 (counted from the left) results in a positive peak whereas the 3 others undergo a transition from 1 to 0, leading to a negative peak.
- In addition this bit is a somewhat lucky bit because when it is used as selection bit only guess 50 competes with the right sub-key. This is a particular favorable case occurring here on SBox_1, partly due to the set of 200 used messages. It cannot be extrapolated to other boxes.
- The dispersion of the DPA bias over the guesses is especially confuse when looking at bit 4.

The quality of the modeling proves that those facts cannot be incriminated to the number of acquisitions. Increasing it much higher than 200 does not help: the level of the peaks with respect to the guesses does not evolve and converges to the same ranking. This particular counter-example proves that the ambiguity of DPA does not lie in imperfect estimation but in wrong basic hypotheses.

7.5 Results of model based CPA

For comparison the table hereafter provides the ranking of the 6 first guesses sorted by decreasing correlation rates. This result is obtained with as few as

only 40 curves! The full key is $0x$11 22 33 44 55 66 77 88 in hexadecimal format and the corresponding sub-keys at the first round are 24, 19, 8, 8, 5, 50, 43, 2 in decimal representation.

$SBox_1$		$SBox_2$		$SBox_3$		$SBox_4$		$SBox_5$		$SBox_6$		$SBox_7$		$SBox_8$	
K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}
24	92%	19	90%	8	87%	8	88%	5	91%	50	92%	43	89%	2	89%
48	74%	18	77%	18	69%	44	67%	32	71%	25	71%	42	76%	28	77%
01	74%	57	70%	05	68%	49	67%	25	70%	05	70%	52	70%	61	76%
33	74%	02	70%	22	66%	02	66%	34	69%	54	70%	38	69%	41	72%
15	74%	12	68%	58	66%	29	66%	61	67%	29	69%	0	69%	37	70%
06	74%	13	67%	43	65%	37	65%	37	67%	53	67%	30	68%	15	69%

This table shows that the correct guess always stands out with a good contrast. Therefore a sound decision can be made without any ambiguity despite a rough estimation of ρ_{max}.

A similar attack has also been conducted on a 32-bit implementation, in a white box mode with a perfect knowledge of the implemented substitution tables and the reference state which was 0. The key was $0x$7C A1 10 45 4A 1A 6E 57 in hexadecimal format and the related sub-keys at the 1st round were 28, 12, 43, 0, 15, 60, 5, 38 in decimal representation. The number of curves is 100.

$SBox_1$		$SBox_2$		$SBox_3$		$SBox_4$		$SBox_5$		$SBox_6$		$SBox_7$		$SBox_8$	
K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}
28	77%	12	69%	43	73%	0	82%	15	52%	60	51%	5	51%	38	47%
19	36%	27	29%	40	43%	29	43%	03	33%	10	34%	15	40%	05	29%
42	35%	24	27%	36	35%	20	35%	58	30%	58	33%	6	29%	55	26%
61	31%	58	27%	06	33%	60	32%	10	30%	18	31%	12	29%	39	25%

Here again we can notice the good contrast (around 40% on boxes 1 to 4) between the correct and the most competing wrong guess. The correlation rate is not that high on boxes 5 to 8, definitely because of partial and imperfect modeling. However the indication remains quite exploitable. This result shows that correlating the measurements with the model is a robust approach. When the number of bits per machine word is greater, the contrast between the guesses is relatively enhanced, but of course finding the right model could be more difficult in a black box mode.

8 Conclusion

The results presented above have all been obtained in white box mode for didactic purpose. But our experience on a large set of experiments over the last years shows that model based correlation power analysis works quite well even with very few prior information available. Anyway the proposed CPA method always provides better and unambiguous results with fewer acquisitions than DPA. Should it fail, so would DPA. An important and reassuring conclusion

is that all the countermeasures designed against DPA offer the same defensive efficiency against the model based CPA attack. This is not that surprising since those countermeasures aim at undermining the common prerequisites that both approaches are based on: side-channel observability and intermediate variable predictability.

A slight drawback of the correlation analysis is that the calculation of $\hat{\rho}$ is more costly than the computation of the DPA bias T. More critical features are the retrieval of the reference state by exhaustive search and the algorithmic assumptions that have to be validated first (like SBox implementation). As it is more demanding the method may seem more difficult than DPA. However with some expertise and a bit of astuteness these complementary elements can be inferred without excessive efforts knowing that:

- There exists many situations where the implementation variants are not so numerous due to operational constraints. Moreover correlation analysis can help a lot as a reverse engineering tool in order to validate some implementation assumptions before applying the attack itself.
- Regarding the inference of the reference state, say on a 8-bit microprocessor, the exhaustive search over the 256 possible values of R can be performed only once on a single SBox. In general the same value remains valid for the seven other SBoxes.

All these observations confirm the model based CPA method as a sound and optimal approach that brings sensitive improvements to DPA. Its reliability suggests to propose it as characterization means for the testing of cryptographic devices. A particularly interesting point is that it provides a quantitative information because the maximum correlation factor can be interpreted as a leakage rate, under standard experimental conditions. The test can be performed very easily on a simple process, such as a memory to memory known message transfer, in order to assess the behavior of a particular hardware device (or piece of device) with regard to the side channel involved.

References

1. Numerical Recipes in C: The Art of Scientific Computing. *Second Edition 1992. Cambridge University Press.* pp. 636-639. http://nr.com.
2. P.Kocher, J.Jaffe, B. Jun: Introduction to Differential Power Analysis and Related Attacks. http://www.cryptography.com.
3. P.Kocher, J.Jaffe, B. Jun: Differential Power Analysis. *Advances in Cryptology , CRYPTO'99 Proceedings.* Springer-Verlag, LNCS 1666, pp. 388-397, 1999.
4. T. Messerges, E. Dabbish, R. Sloan: Investigation of Power Analysis Attacks on Smartcards. *Advances in Cryptology , CRYPTO'99 Proceedings, Usenix Workshop on Smartcard Technology 1999.* http://www.usenix.org.
5. L. Goubin and J. Patarin: DES and Differential Power Analysis. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 1999).* Springer LNCS 1717, p. 158 ff.
6. J. S. Coron, P. Kocher, D. Naccache: Statistics and Secret Leakage. *Proceedings of Financial Cryptography, 2000.* Springer-Verlag LNCS 1972, pp. 157-173.

7. C. Clavier, J. S. Coron, N. Dabbous: Differential Power Analysis in the Presence of Hardware Countermeasures. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 2000)*. Springer LNCS 1965, pp. 252-263.
8. A. Shamir: Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 2000)*. LNCS 1965, p. 71 ff.
9. Thomas S. Messerges: Using Second-Order Power Analysis to Attack DPA Resistant Software. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 2000)*. LNCS 1965, p. 238 ff.
10. R. Mayer-Sommer: Smartly Analysing the Simplicity and the Power of Simple Power Analysis on Smartcards. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 2000)*. Springer LNCS 1965, p. 78.
11. Jean-Sébastien Coron and Louis Goubin: On Boolean and Arithmetic Masking against Differential Power Analysis. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 2000)*. LNCS 1965, p. 231 ff.
12. M. L. Akkar, R. Bévan, P. Dischamp, D. Moyart: Power Analysis, what is now possible... *ASIACRYPT 2000*. LNCS 1976, pp. 489-502, 2000.
13. K. Gandolfi, C. Mourtel, F. Olivier: Electromagnetic Attacks: Concrete Results. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 2001)*. Proceedings in Springer LNCS 2162 pp. 252-261.
14. M. L. Akkar, C. Giraud: An Implementation of DES and AES secure against some attacks. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 2001)*. Proceedings in Springer LNCS 2162 pp. 309-318.
15. D. Agrawal, B. Archambeault, J. R. Rao, P. Rohatgi: The EM Side Channel(s): Attacks and Assessment Methodologies. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002)*. Springer LNCS 2523, pp. 29-45. See also http://www.research.ibm.com.intsec/emf-paper.ps.
16. T. Messerges, E. Dabbish, R. Sloan: Examining Smart-Card Security under the Threat of Power Analysis Attacks. *IEEE Transactions on Computers*, Vol. 51, N5, pp. 541-552. May 2002.
17. E. Trichina, D. De Seta, L. Germani: Simplified Adaptive Multiplicative Masking for AES. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002)*. Springer LNCS 2523, pp. 187-197.
18. J. Golic, C. Tymen: Multiplicative Masking and Power Analysis of AES. *In Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002)*. Springer LNCS 2523, pp. 198-212.
19. R. Bévan, R. Knudsen: Ways to Enhance Differential Power Analysis. *ICISC 2002*. Springer-Verlag LNCS 2587 pp. 327-342.