Correlation Power Analysis with a Leakage Model

Eric Brier, Christophe Clavier, and 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 [12, 13] and formalized by Thomas Messerges et al. [16]. The second one has been suggested in various papers [8, 14, 18] 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. This work follows previous studies aiming at either improving the Hamming weight model [2], or enhancing the DPA itself by various means [6, 4].

The proposed approach is based on the Hamming distance model which can be seen as a generalization of the Hamming weight model. All its basic assumptions were already mentioned in various papers from year 2000 [16, 8, 6, 2]. But they remained allusive as possible explanation of DPA defects and never leaded to any complete and convenient exploitation. Our experimental work is a synthesis of those former approaches in order to give a full insight on the data leakage. Following [8, 14, 18] we propose to use the correlation power analysis (CPA) to identify the parameters of the leakage model. Then we show that sound and efficient attacks can be conducted 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. Our conclusion summarizes the advantages and drawbacks of CPA versus DPA and reminds that countermeasures work against both methods as well.

2 The Hamming Distance Consumption Model

Classically, most power analyses found in literature are based upon the Hamming weight model [13, 16], 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, $H(D) = \sum_{j=0}^{m-1} d_j$. Its integer values stand between 0 and m. If D contains m independent and uniformly distributed bits, the whole word has an average Hamming weight $\mu_H = m/2$ and a variance $\sigma_H^2 = m/4$.

It is generally assumed that the data leakage through the power side-channel depends on the number of bits switching from one state to the other [6, 8] at a given time. A microprocessor is modeled as a state-machine where transitions from state to state are triggered by events such as the edges of 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. Only hardware designers are familiar with simulation tools to foresee the current consumption of microelectronic devices.

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. Moreover, it is assumed that switching a bit from 0 to 1 or from 1 to 0 requires the same amount of energy and that all the machine bits handled at a given time are perfectly balanced and consume the same.

These restrictive assumptions are quite realistic and affordable without any thorough knowledge of microelectronic devices. They lead to a convenient expression for the leakage 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 $H(D)$.

We also assume a linear relationship between the current consumption and $H(D \oplus R)$. This can 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 because the bus lines are usually considered as the most consuming elements within a micro-controller. All the remaining things in the power consumption of a chip are assigned to a term denoted b which is assumed independent from the other variables: b encloses offsets, time dependent components and noise. 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.

3 The Linear Correlation Factor

A linear model implies some relationships between the variances of the different terms considered as random variables: $\sigma_W^2 = a^2\sigma_H^2 + \sigma_b^2$. Classical statistics introduce the correlation factor ρ_{WH} between the Hamming distance and the measured power to assess the linear model fitting rate. 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. 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, just like in DPA, that a set of known but randomly varying 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 microcontroller, the case with many of today's smart cards, as only 256 values are to be tested. On 32-bit architectures this exhaustive search cannot be applied as such. But it is still possible to work with partial correlation or to introduce prior knowledge.

Let R be the true reference and $H = H(D \oplus R)$ the right prediction on the Hamming distance. Let R' represent a candidate value and H' the related model $H' = H(D \oplus R')$. Assume a value of R' that has k bits that differ from those of R, then: $H(R \oplus R') = k$. Since b is independent from other variables, the correlation test leads to (see [5]):

$$\rho_{WH'} = \frac{\mathrm{cov}(aH + b, H')}{\sigma_W \sigma'_H} = \frac{a}{\sigma_W} \frac{\mathrm{cov}(H, H')}{\sigma'_H} = \rho_{WH}\rho_{HH'} = \rho_{WH}\frac{m - 2k}{m}$$

This formula shows how the correlation factor is capable of rejecting wrong candidates for 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 $\rho_{WH'} = -\rho_{WH}$. 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 and 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, $D = K \oplus M$, 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 an analysis in the best conditions, we emphasize

the benefit of correctly modeling 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 $H_{i,R} = H(M_i \oplus R)$. An estimate $\hat{\rho}_{WH}$ of the correlation factor ρ_{WH} is given 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 ($i = 1, N$) at each time step within the power traces $W_i(t)$.

It is theoretically difficult to compute the variance of the estimator $\hat{\rho}_{WH}$ with respect to the number of available samples N. In practice a few hundred experiments suffice to provide a workable estimate of the correlation factor. N has to be increased with the model variance $m/4$ (higher on a 32-bit architecture) and in presence of measurement noise level obviously. Next results will show that this is more than necessary for conducting reliable tests. The reader is referred to [5] for further discussion about the estimation on experimental data and optimality issues. It is shown that this approach can be seen as a maximum likelihood model fitting procedure when R is exhausted to maximize $\hat{\rho}_{WH}$.

6 Experimental Results

This section aims at confronting the leakage model to real experiments. General rules of behavior are derived from the analysis of various chips for secure devices conducted during the passed years.

Our first experience was performed onto a basic XOR algorithm implemented in a 8-bit chip known for leaking information (more suitable for didactic purpose). 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, the second one the opcode of the XOR instruction. These curves bring the experimental evidence of leakage principles that previous works just hint at, without going into more detail [16, 8, 6, 17]. They illustrate the most general case of a transfer sequence

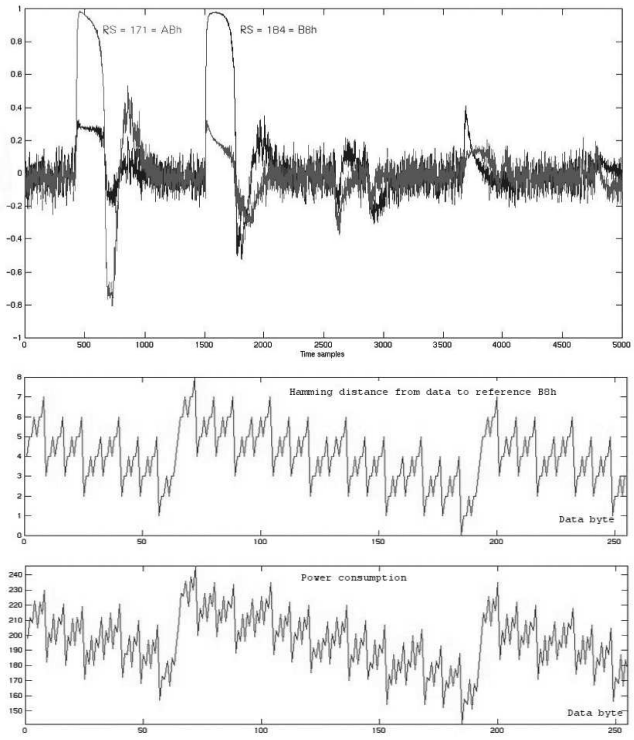

Fig. 1. Upper: consecutive correlation peaks for two different reference states. Lower: for varying data (0-255), model array and measurement array taken at the time of the second correlation peak.

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. Such a 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 displayed correlations.

This sort of results can be observed on various technologies and implementations. Nevertheless the following restrictions have to be mentioned:

– Sometimes the reference state is systematically 0. This can be assigned to the so-called pre-charged logic where the bus is cleared between each 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.

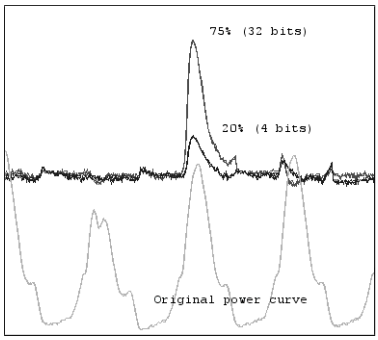

Fig. 2. Two correlation peaks for full word (32 bits) and partial (4 bits) predictions. According to theory the 20% peak should rather be around 26%.

- 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 [6, 18]. In certain cases their effect can be corrected by applying appropriate signal processing.
- Other countermeasures consist in blurring the power traces with additional noise or filtering circuitry [19]. Sometimes they can be bypassed by curves selection and/or averaging or by using another side channel such as electromagnetic radiation [9, 1].
- The data can also be ciphered dynamically during a process by hardware (such as bus encryption) or software means (data masking with a random [11, 7, 20, 10]), so that the handled variables become unpredictable: then no correlation can be expected anymore. In theory sophisticated attacks such as higher order analysis [15] can overcome the data masking method; but they are easy to thwart in practice 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. [16, 17] who formalized the ideas previously suggested by Kocher [12, 13]. A critical study is proposed in [5].

7.1 Practical problems with DPA: the "ghost peaks"

We just consider hereafter the practical implementation of DPA against the DES substitutions (1st round). In fact this well-known attack works quite well only if the following assumptions are fulfilled:

1. Word space assumption: within the word hosting the predicted bit, 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 sub-key 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.

But when confronted to the experience, the attack 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.

7.2 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 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:

$$\begin{array}{l}
1101101010010110001001011001001110101001011011010101001000101101 \\
1001101011010110001001011101001010101101011010010101001000111001 \\
0100000001000000000000000100000100000100000001000000000000010100
\end{array}$$

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 [4]. It consists in shifting the DPA attacks a little bit further in the processing and perform the prediction just after the end of the first round when the right part of the data (32 bits) is XORed with the left part of the IP output. As the message is 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 .

To get rid of these ambiguities the model based approach aims at taking the whole information into account. 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 figure 3 shows, the simulated and experimental DPA biases match particularly well. One can notice the following points:

- 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 quite confuse (see 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.

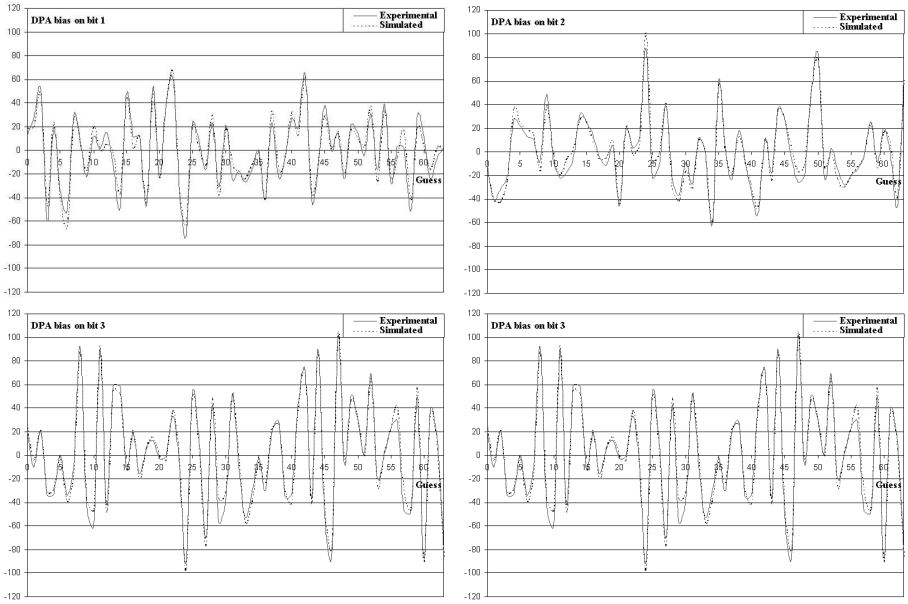

Fig. 3. DPA biases on SBox_1 versus guesses for selection bits 1, 2, 3 and 4, on modeled and experimental data; the correct guess is 24.

7.3 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 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 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. As next table shows, the contrast is good between the correct and the most competing wrong guess (around 40% on boxes 1 to 4). The correlation rate is not that high on boxes 5 to 8, definitely because of partial and imperfect modeling, but it proves to remain exploitable and thus a robust indicator. When the number of bits per machine word is greater, the contrast between the guesses is relatively enhanced, but finding the right model could be more difficult in a black box mode.

$SBox_1$		$SBox_2$		$SBox_3$		$SBox_4$		$SBox_5$		$SBox_6$		$SBox_7$		$SBox_8$	
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%

8 Conclusion

Our experience on a large set of smart card chips over the last years has convinced us on the validity of the Hamming distance model and the advantages of the CPA method against DPA, in terms of efficiency, robustness and number of experiments. 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.

The main drawback of CPA regards the characterization of the leakage model parameters. As it is more demanding than DPA, the method may seem more difficult to implement. However it may be objected that:

- A statistical power analysis of any kind is never conducted blindly without any preliminary reverse engineering (process identification, bit tracing): this is the opportunity to quantify the leakage rate by CPA on known data.
- DPA requires more sample curves anyway since all the unpredicted data bits penalize the signal to noise ratio (see [5]).
- If DPA fails by lack of implementation knowledge (increasing the number of curves does not necessarily help), we have shown how to infer a part of this information without excessive efforts: for instance the reference state is to be found by exhaustive search only once in general.
- There exists many situations where the implementation variants (like SBox implementation in DES) are not so numerous because of operational constraints.
- If part of the model cannot be inferred (SBox implementation in DES, hardware co-processor), partial correlation with the remainder may still provide exploitable indications.

Eventually DPA remains relevant in case of very special architectures for which the model may be completely out of reach, like in certain hard wired co-processors.

References

1. D. Agrawal, B. Archambeault, J.R. Rao, and P. Rohatgi. The EM side channel(s): Attacks and assessment methodologies. In *Cryptographic Hardware and Embedded Systems — CHES 2002*, LNCS 2523, pp. 29–45, Springer-Verlag, 2002. See also http://www.research.ibm.com.intsec/emf-paper.ps.
2. M.L. Akkar, R. Bévan, P. Dischamp, and D. Moyart. Power analysis, what is now possible... In *Advances in Cryptology — ASIACRYPT 2000*, LNCS 1976, pp. 489–502, Springer-Verlag, 2000.
3. M.L. Akkar and C. Giraud. An Implementation of DES and AES secure against some attacks. In *Cryptographic Hardware and Embedded Systems — CHES 2001*, LNCS 2162 pp. 309–318, Springer-Verlag, 2001.
4. R. Bévan and R. Knudsen. Ways to enhance differential power analysis. In *Information Security and Cryptology — ICISC 2002*, LNCS 2587, pp. 327–342, Springer-Verlag, 2002.
5. E. Brier, C. Clavier, and F. Olivier. Optimal statistical power analysis. http://eprint.iacr.org/2003/152/.
6. C. Clavier, J.-S. Coron, and N. Dabbous. Differential power analysis in the presence of hardware countermeasures. In *Cryptographic Hardware and Embedded Systems — CHES 2000*, LNCS 1965, pp. 252–263, Springer-Verlag, 2000.
7. J.-S. Coron and L. Goubin. On Boolean and arithmetic masking against differential power analysis. In *Cryptographic Hardware and Embedded Systems — CHES 2000*, LNCS 1965, pp. 231–237, Springer-Verlag, 2000.
8. J.-S. Coron, P. Kocher, and D. Naccache. Statistics and secret leakage. In *Financial Cryptography (FC'2000)*, LNCS 1972, pp. 157–173, Springer-Verlag, 2001.
9. K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic attacks: Concrete results. In *Cryptographic Hardware and Embedded Systems — CHES 2001*, LNCS 2162, pp. 252–261, Springer-Verlag, 2001.
10. J. Goliç and C. Tymen. Multiplicative masking and power analysis of AES. In *Cryptographic Hardware and Embedded Systems — CHES 2002*, LNCS 2523, pp. 198–212, Springer-Verlag, 2002.
11. L. Goubin and J. Patarin. DES and differential power analysis. In *Cryptographic Hardware and Embedded Systems (CHES '99)*, LNCS 1717, pp. 158–172, Springer-Verlag, 1999.
12. P. Kocher, J. Jaffe, and B. Jun. Introduction to differential power analysis and related attacks. http://www.cryptography.com.
13. P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Advances in Cryptology — CRYPTO '99*, LNCS 1666, pp. 388–397, Springer-Verlag, 1999.
14. R. Mayer-Sommer. Smartly analysing the simplicity and the power of simple power analysis on smartcards. In *Cryptographic Hardware and Embedded Systems — CHES 2000*. LNCS 1965, pp. 78–92, Springer-Verlag, 2000.
15. T.S. Messerges. Using second-order power analysis to attack DPA resistant software. In *Cryptographic Hardware and Embedded Systems — CHES 2000*. LNCS 1965, pp. 238–252, Springer-Verlag, 2000.
16. T. Messerges, E. Dabbish, and R. Sloan. Investigation of power analysis attacks on smartcards. In *Usenix Workshop on Smartcard Technology 1999*. http://www.usenix.org.

17. T. Messerges, E. Dabbish, and R. Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, 51(5): 541–552, May 2002.
18. E. Oswald. On Side-Channel Attacks and the Application of Algorithmic Countermeasures. PhD Thesis, Faculty of Science of the University of Technology Graz (IAIK-TUG), Austria, May 2003.
19. A. Shamir. Protecting smart cards from passive power analysis with detached power supplies. In *Cryptographic Hardware and Embedded Systems — CHES 2000*. LNCS 1965, pp. 71–77, Springer-Verlag, 2000.
20. E. Trichina, D. De Seta, and L. Germani. Simplified adaptive multiplicative masking for AES. In *Cryptographic Hardware and Embedded Systems — CHES 2002*, LNCS 2523, pp. 187–197, Springer-Verlag, 2002.