Side-channel attacks based on linear approximations

Thomas Roche[1][2][3] and Cédric Tavernier[3]

[1] Laboratoire d'Informatique de Grenoble, 51 av. Jean Kuntzmann, 38330 Montbonnot-Saint-Martin, France,
Thomas.Roche@imag.fr
[2] Université Joseph Fourier
[3] CS, Communication&Systems, 22 avenue Galilée, 92350 Le Plessis Robinson, France
Cedric.tavernier@c-s.fr

Abstract. Power analysis attacks against embedded secret key cryptosystems are widely studied since the seminal paper of Paul C. Kocher, Joshua Jaffe and Benjamin Jun in 1998 where has been introduced the powerful Differential Power Analysis. The strength of DPA is such that it became necessary to develop sound and efficient countermeasures. Nowadays embedded cryptographic primitives usually integrate one or several of these countermeasures (e.g. masking techniques, asynchronous designs, balanced dynamic dual-rail gates designs, noise adding, power consumption smoothing, etc. ...). This document presents new power analysis attacks based on linear approximations of the target cipher. This new type of attacks have several advantages compared to classical DPA-like attacks: first they can use multiple intermediate values by query (i.e. power trace) allowing to reduce data complexity to a minimum, secondly they can be applied on parts of the symmetric cipher that are practically unreachable by DPA-like attacks and finally they can be mounted on an unknown cipher implementation.

Keywords: Side-channel Attacks, Power Analysis, multi-linear cryptanalysis, Reed-Muller codes.

1 Introduction

Under the term "side-channel attacks" are gathered the cryptanalysis of cryptographic primitives involving the study of the mathematical object as well as its execution environment. Indeed, the observation or perturbation of the execution environment at runtime gives to the attacker new inputs — on the internal state of the target algorithm — which were not assumed to be known in *classical cryptanalysis*. Since the late 90's and the discovery of several astonishingly dangerous side-channel attacks, the study of this kind of cryptanalysis and how to prevent it has taken a prodigious takeoff.
Among the side-channel attacks, the *Power Analysis* (PA) attacks were introduced in the seminal paper of Paul C. Kocher, Joshua Jaffe and Benjamin Jun in 1998 [22]. Kocher et al. proposed to use the variation of the circuit power consumption as information to break ciphers implementations and introduced two attacks based on this very idea: the *Simple Power Analysis* (SPA) and the *Differential Power Analysis* (DPA). Two years later Jean-Jacques Quisquater and David Samyde [27] observe that the variations of electromagnetic

radiations give similar information. For more simplicity in the sequel, the term *power consumption* covers also the electromagnetic radiations.
DPA and its variant *High-Order DPA* are certainly the most studied PA attacks on symmetric ciphers nowadays, the development of countermeasures is an active subject of research that try to answer the question: How to make the power consumption variations either useless or null? Although not efficient in practice, several solutions exist: masking techniques [15, 7, 1, 9, 29], desynchronization techniques [33], balanced dynamic dual-rail gates designs [5, 8], noise adding, power consumption smoothing or logic level secure designs [2, 32].
In order to bypass some limitations of DPA-like attacks (discussed in section 2.2), other PA attacks were developed, most of them based on classical cryptanalysis fundamentals such as differential, algebraic cryptanalysis of bloc cipher or collisions.

In this paper we propose to investigate the use of Matsui's linear cryptanalysis [26] in PA attacks. To our knowledge, this is the first use of linear approximations in side-channel. After introducing the symmetric cipher model used for our study and the known PA attacks (section 2), we will present (1) how to use linear approximations in DPA-like attacks, discuss its advantages compared to exact DPA and its complexity, (2) propose a new type of PA attacks, the Multi-Linear Power Analysis attacks (M-LPA), and discuss different practical setups. Finally, experimental results on publicly available consumption traces (the DPA-contest traces `http://www.dpacontest.org/`) prove the feasibility and the efficiency of the M-LPA attacks.

2 Preliminaries on embedded symmetric ciphers and Power Analysis attacks

In this section is first discussed the symmetric cipher design model on which our study has been done and then the way Power Analysis attacks can be applied to those designs.

2.1 Embedded symmetric cipher design model

We want our study to be as general as possible, taking in account symmetric cipher implementations for microcontrolers (we refer to *software implementation*) and smartcards and FPGA devices that are meant to bear a symmetric cipher implementation (we refer then to *hardware implementation*).
Software implementations as well as hardware implementations can take lots of forms considering the microcontroler specifications for the first case and the implementation design for the second one. And in both cases the implementation can be designed for restricted areas, consumption and/or high throughput. For reasons of clarity, we will describe the studied designs using the common shape of symmetric ciphers: Substitution-Permutation Network (SPN) composed in rounds (the key schedule won't be taken in account for our

study, we only suppose the round keys to be available when needed). A symmetric cipher can be represented as on Figure 1 (note that the sub-blocks within a round can be ordered more or less differently).

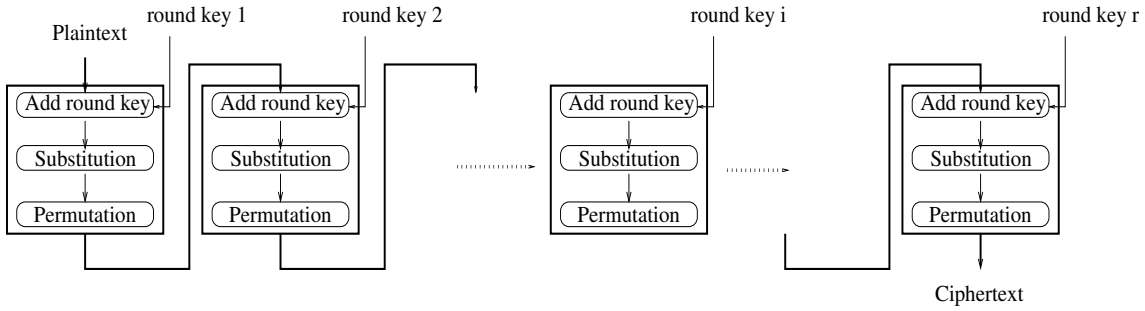

Fig. 1. Schematic of a Symmetric Cipher

As we will see in the next section, the critical parameter when considering side-channel attacks, and specially power analysis attacks, is the data bus size (and then the corresponding data register size). Indeed, these components are considered to be stable points from which the leaked information can be interpreted. Hence, we will consider that the microcontroler and, equivalently the hardware implementation, has a register size of $\boldsymbol{r}$. This simply means that, at each clock cycle, r bits of the current intermediate computation are set in the data register, processed by the microcontroler or via combinational logic, and so on. Moreover, in order to simplify our analyse, we first assume that the knowledge of this high level design (what is computed during each clock cycle) is known by the attacker (e.g. using probing techniques). And finally we will consider that the symmetric cipher's block size m is such that $m = k \times r$ (e.g., for 8-bits microcontrolers $r = 8, k = m/8$, for the implementation in ASIC used for our experiments $r = m, k = 1$). That way, when considering an execution of the cipher algorithm is done in C clock cycles, we can assume that aver the whole execution there is C points where information is leaked on r-bits values, the intermediate values will be noted $\{V_i\}_{1 \leq i \leq C}$ in the sequel (Although not all of them are interesting, e.g. intermediate values that are already known as the plaintext or ciphertext).

2.2 Power Analysis Attacks

Power analysis attack is a dynamic and involved source of research as the development of resistant cryptographic hardware devices is needed. The study of PA attacks and their countermeasure has taken a prodigious takeoff since the introduction of the very efficient DPA attacks in 1998.

Power consumption in CMOS circuits Without going into the depths of CMOS gates power consumption (a simple, yet enough for our need, presentation can be found in [25] pages 27-60) what we would like to point out here is that the power consumption of CMOS circuits is dependent on the data manipulated as transitions from 0 to 1 and 1 to 0 consume significantly more power than 0 to 0 or 1 to 1 transitions through a logical gate. An attacker observing the overall consumption of a CMOS circuit during two different execution can tell, at a chosen point in time, which execution has led to a greater number of data changes. What is remarkable to note though is that power consumption of combinational logic (in ASIC or FPGA) at a point within a clock cycle won't give the attacker relevant information on the data: The attacker has not a precise enough knowledge of the netlist to be able to predict the glitches occurring throughout the logic circuit. Considering this, power analysis is based on the study of registers and buses power consumption since theirs data transitions are synchronized with the clock fronts and don't involve combinational logic. To our knowledge all PA attacks are based on this principle.

Hamming distance and Hamming weight models When considering the consumption of a bus or register, since the consumption power is significantly higher when a bit value change, in the Hamming distance model (HD) the power consumption is closely related to the Hamming weight of the difference (bitwise Xor) of two successive data values. Note that, of course, absolute values of the measured power traces are not of any use for the attacker, but relative values with respect to other measurement are relevant.
A more simple model, the Hamming weight model (HW), approximates the power consumption directly by the Hamming weight of the manipulated data value.
Hence, with respect to the HD model (resp. the HW model), the leaked information one can retrieve from a symmetric cipher execution is all the Hamming weight of the values $\{V_i\}_{1 \leq i \leq C}$ seen as differences of successive intermediate values (resp. intermediate values). Other models exists, they are basically variants of those models based on some knowledge the attacker might have on the targeted hardware design.

Side-channel attacks

Differential Power Analysis By statistical means, DPA allows the attacker to suppress the measurement noises and bring to light data-dependent operations. Let us borrow the notations of [22] here:

- $\mathbf{T_i[j]}$: The j^{th} sample of T_i, the i^{th} recorded power trace.
- $\mathbf{D(P, B, K_s)}$: DPA selection function, computes B (Hamming weight of intermediates bits at a fixed point of time), as a function of a secret key block K_s and the plaintext P (could also be the ciphertext C). In the original DPA from [22] on DES, B is the Hamming weight of one intermediate bit (i.e. the value of one bit). For now let assume the value of B is 0 or 1.

After observing M executions of the cryptographic primitive, recording each power trace $T_{1\cdots M}[1\cdots k]$ (k samples) and the corresponding plaintexts $P_{1\cdots M}$ (respectively ciphertexts $C_{1\cdots M}$), the attacker computes the value of $\{B_i\}_{1\cdots M}$ using the selection function $D(P_i, B_i, K_s)$ (for an arbitrary fixed K_s). The traces are divided in two sets S_0 and S_1, such that $T_i \in S_0$ iff $B_i = 0$, $T_i \in S_1$ otherwise and the differential trace over the k samples is computed:
$\forall 1 \leq i \leq k$,

$$\Delta_D[i] = \mathbf{E}[T[i] \mid T[i] \in S_1] - \mathbf{E}[T[i] \mid T[i] \in S_0]$$

Where $E[T[i] \mid T[i] \in S]$ is the expected value of the circuit power consumption in sample i over the traces assumed to be in S. It is then easy to see, considering the previous remark on power consumption of CMOS gates, that if the partition S_0 and S_1 is well chosen (i.e. the guess of K_s and then the value of B is right), then there should be a difference in the average power consumption over the set S_1 and S_0. In another hand, if the partition is wrong, then the average power consumption over either S_O and S_1 should approach the global average power consumption and then $\Delta_D[i] \approx 0$.
This statistical technique is called the *difference of means*. Since Kocher's paper, other techniques have been proposed in order to statistically measure dependencies between the consumption values guessed and measured, let us mention the use of the covariance, of Pearson coefficient for the Correlation Power Analysis (CPA) proposed by E. Brier et al. [4] in 2004 and entropy for the Mutual Information Analysis (MIA) proposed by Gierlichs et al.[14] in 2008. Moreover, when our description details a mono-bit DPA (B represent a single bit), a more complicated selection function can be used where B can take more than two values (Hamming weight of an intermediate data value), those kind of attacks (DPA multi-bits) have been gathered under the name Partitioning Power Analysis (PPA) by Thanh-Ha Le [23].

Other side-channel attacks Even though DPA-like attacks are nowadays the most studied and applied power analysis attacks, several other methods, of special interest for us, have also been proposed. Most of them, as the present proposition, introduce concepts from classical cryptanalysis into the side-channel world. For instance, collisions-based side-channel attacks [31, 30, 24], differential cryptanalysis-based side channel attacks [6, 17], and the very recent algebraic cryptanalysis-based side-channel attacks [28].

Remark 1. Interestingly enough, these attacks have a common asset compared to DPA-like attacks: they can target intermediate values $\{V_i\}_{1\leq i\leq C}$ that are not reachable by DPA. As we have seen in the DPA description, an attacker using DPA must make a guess on a subset of the key bits in order to evaluate some intermediate values. This condition restrict drastically the range where targeted intermediate values can be chosen since they should not depend on too many bits of key. Depending on the symmetric cipher diffusion function, a DPA attack is not feasible after the firsts and before the lasts rounds. Hence the classical cryptanalysis-based side-channel attacks are of special interest when considering

an implementation where countermeasures have been added against DPA but limited to few rounds to limit their cost.

Remark 2. The attack presented in this article has very strong links with the idea developed by Renauld and Standaert with the algebraic side-channel attack [28]. Indeed, they integrate the information leaked from as many intermediates values $\{V_i\}_{1 \leq i \leq C}$ as possible into the system of equations corresponding to the block cipher (see [10]). In our approach, instead of using exact equations (of possible high degree), we use approximations of degree 1. Our system is much easier to solve since it is linear but the approximations add noise, thus increasing the complexity of the attack (with respect to the number of plaintext or ciphertext needed).
A challenging open problem would be to find the best compromise, either by mixing straightforwardly the two approaches (e.g., add some linear approximations to the algebraic system) or by increasing the degree of the approximations. As the degree increases, the approximations will be tighter (see section 3.1) but the resulting system harder to solve, the two boundaries of this research area being M-LPA and Algebraic side-channel attacks.

3 M-LPA Attack description and complexity

In this section is introduced Linear Power Analysis and Multi-Linear Power analysis attacks. Those attacks correspond strictly to Linear ([26]) and Multi-Linear cryptanalysis ([19]) in the side-channel context. We first introduce some useful notations for the study of linear approximations. Then we will present the idea of M-LPA before describing the attacks algorithms and complexity. Finally we will discuss its practical setup and detail the experimental results.

3.1 Linear approximations of a symmetric cipher

Linear cryptanalysis has been introduced by Matsui in 1993 ([26]), since then it has become one of the most important base of the study of block cipher security. Nowadays new block ciphers must prove some inherent resistance against linear cryptanalysis. Let us remark that many cryptanalysis methods are based on this fundamental discovery, among others, the multi-linear cryptanalysis [19, 3] will be particularly interesting here.

linear cryptanalysis A linear approximation is a boolean linear function that takes plaintext and key bits as input and outputs a combination of ciphertext bits.
Let us denote $|K|$, $|P|$, $|C|$ respectively the bit-lengths of key, plaintext and ciphertext. Let us consider a vector Π of length $|P|$, κ of length $|K|$ and Γ of length $|C|$ and a bit b. Π, κ, Γ and b define a linear approximation of bias ϵ over the symmetric cipher if and only if:

$$\Pr_{P,K}(< P, \Pi > \oplus < K, \kappa > \oplus b =< C(P,K), \Gamma >) \geq 1/2 + \epsilon \tag{1}$$

where $< x, y >$ is the scalar product of two vectors x and y of same length over $GF(2)$ (vector of bits). Given such a linear equation, Matsui showed that a high probability of success to recover the involved key bits in the equation using linear cryptanalysis would require a data-complexity (i.e. number of plaintext-ciphertext pairs) of $N = 1/\epsilon^2$.

Multi-linear cryptanalysis It was shown in [3] that instead of using a single linear approximation, the use of several linear approximations involving the same key bits would significantly improve the performances of the attack. As a matter of fact, given n linearly independent approximations of respective bias $\epsilon_j, j = 1, \cdots, n$ the data-complexity of the attack would be reduced to

$$N \approx 1/\sum_{j=1}^{n} \epsilon_j^2$$

In a very recent paper R. Fourquet, P. Loidreau and C. Tavernier [12] studied the problem of finding all the linear approximations with a given bias of a given Boolean function. The authors showed the equivalence between the problem of finding linear approximations for a fixed output mask (Γ fixed) and a list decoding problem in the first order Reed-Muller code. They were then able to find good linear approximations up to 8 rounds of DES and thus, based on results of [13], break a reduced version of the cipher with low data-complexity (2^{21} plaintext-ciphertext pairs). In the following descriptions or applications of attacks, the linear approximations are generated using the list decoding algorithm of [12]. A brief description of the algorithm and its complexity is presented in Annex 1 of this document.

3.2 Introduction to M-LPA

As mentioned above, M-LPA implies the use of linear approximations to attack a symmetric cipher hardware implementation by power analysis. We will introduce two different ways to use linear approximations by an attacker, the later will be the so called M-LPA attack. Let us denote $H(u)$ the Hamming weight function of a vector of bits u.

A first approach: a DPA approach A very straightforward approach would be to attack by DPA using a linear approximation as base of the selection function. This will render the attack's selection function dependent on the approximation bias ϵ and thus increase the data complexity. The advantage of such an attack will be to find linear approximation that involve few bits of the key (less than 32 in practice) when evaluating data values in registers or going through buses that are strictly dependent on more than 32 key bits from the point of view of the cipher function. Hence it would allow to attack a cipher implementation where a sound countermeasure is used only for the data bits that depend on less than 32 key bits.

For instance let us consider the mono-bit DPA attack presented by Kocher in [22] with the notations introduced in section 2.2. We are now considering a selection function that is probabilistic, in other word, for each trace T_i it has a probability $p = 1/2 + \epsilon$ of choosing the right set (S_0 or S_1) for T_i. Let us denote S_0 and S_1 the sets of traces created with an

exact selection function and $\widehat{S}_0$ and $\widehat{S}_1$ the ones resulting from the use of a probabilistic selection function (of bias ϵ). The differential trace is then:
$\forall 1 \leq i \leq k$,

$$\Delta_D[i] = \mathbf{E}[T[i] \mid T[i] \in \widehat{S}_1] - \mathbf{E}[T[i] \mid T[i] \in \widehat{S}_0]$$

and then, by the properties of the expected function:

$$\begin{aligned}\Delta_D[i] = \quad & (p\mathbf{E}[T[i] \mid T[i] \in \widehat{S}_1, T[i] \in S_1] + (1-p)\mathbf{E}[T[i] \mid T[i] \in \widehat{S}_1, T[i] \in S_0]) \\ - & (p\mathbf{E}[T[i] \mid T[i] \in \widehat{S}_0, T[i] \in S_0] + (1-p)\mathbf{E}[T[i] \mid T[i] \in \widehat{S}_0, T[i] \in S_1])\end{aligned}$$

One has just to remark that for any $(l, j) \in \{1; 2\}^2$, the event $(T[i] \in \widehat{S}_l)$ gives less information than $(T[i] \in S_j)$ on the average value of $T[i]$. We then obtain

$$\begin{aligned}\Delta_D[i] = \quad & (p\mathbf{E}[T[i] \mid T[i] \in S_1] + (1-p)\mathbf{E}[T[i] \mid T[i] \in S_0]) \\ - & (p\mathbf{E}[T[i] \mid T[i] \in S_0] + (1-p)\mathbf{E}[T[i] \mid T[i] \in S_1])\end{aligned}$$

and then
$\forall 1 \leq i \leq k$,

$$\Delta_D[i] = 2\epsilon(\mathbf{E}[T[i] \mid T[i] \in S_1] - \mathbf{E}[T[i] \mid T[i] \in S_0])$$

Finally, the complexity of the attack increases by a factor $1/(2\epsilon)$ as the selection function has a bias ϵ.

Remark 3. The attack described above can be easily extended to multi-linear approximations attack. For each linear approximation, each consumption trace is put in $\widehat{S}_0$ or $\widehat{S}_1$ depending of the approximation evaluation and the consumption measure is weighted with respect to the approximation bias.

Remark 4. The attack cannot be easily extended to multi-bit DPA attacks. Indeed, the approximations being boolean functions, one need r approximations to build a r-bits selection function. As a result, the probability of the r-bit selection function will be the product of the r boolean approximations probability. Very fast, such multi-bit selection function becomes useless due to its very small bias.

Second approach: a HD and HW models approach An interesting way to use linear approximations would be to directly approximate the Hamming weight of a register since this is the quantity which is the most correlated to what is being measured. Thanks to the work of Fourquet et al. (in [12]), it is possible to find linear approximations of $< H(C(P, K)), \Gamma_H >$ with any chosen vector Γ_H (Γ_H is a vector of length $log_2(|C|)$, with respect to the notations of section 3.1).
If we assume that the actual value of the measurement samples $T_i[j]$ is closely related to the value of the hamming weight of the data manipulated (for the HW model) or the

difference between two successive data manipulated (for the HD model), then the use of linear approximations on the hamming weight value of a register (or a bus) would lead to very efficient attacks (a discussion on this assumption is given in the next section). This important remark is the origin of the new M-LPA attacks that should prove themselves much more dangerous than the previous DPA-like approach.

3.3 The M-LPA attack

As introduced in the previous section, the LPA attack is based on the HW and HD models. If we assume that these models are relevant, then multi-linear approximations can be used in all their strength. As presented in [13, 12] in the context of classical multi-linear cryptanalysis, one can consider the recovering of some key-bits as the decoding problem of a code whose length is equal to the number of available linear relations and over a memoryless channel whose capacity depends on the respective biases of the linear approximations. Let us consider a set of n linear relations of biases $\epsilon_l, l = 1, \cdots, n$ with a form as follow:

$$< P, \Pi_l > \oplus < H(V_i(P, K)), \Gamma_{H_l} > \oplus b_l =< K, \kappa_l > \quad (2)$$

where V_i is one of the intermediate values $\{V_i\}_{1 \leq i \leq C}$ defined in section 2.1 (and then Γ_{H_l} is a $log_2(r)$-bits vector) and where the set of vectors κ_l ($l = 1 \cdots n$) are such that a limited number k of key bits are involved in the equations (in practice less than 32 bits) and form a matrix of rank k
The idea is to reconstruct a code word y of length 2^k from a noisy and erased codeword $\tilde{y}$ which is close enough to y, to be able to decode it in the first order Reed-Muller code.

Remark 5. Decoding in the first order Reed-Muller code is done twice, first a list-decoding algorithm is used to find linear approximations, then a unique decoding algorithm is used to recover the key-bits involved in n linear approximation from their non-exact evaluation.

Attack algorithm

After observing N encryptions and selecting the sample j in each traces $T_{i=1...N}$ where the target intermediate value V_i is manipulated, the attack will proceed as follows:

1. For each linear approximation and each "plaintext-$T_I[j]$" pair, compute the predicted value of $< K, \kappa_l >_i$ using the right member of the equation 2 (which would be "$< P_i, \Pi_l > \oplus < T_i[j], \Gamma_{H_l} > \oplus b_l$" since $T_i[j]$ is considered as corresponding to $H(V_i(P_i, K))$).
2. For each linear approximation, separate the traces into two sets S_0^l and S_1^l for which $< K, \kappa_l >$ has been evaluated to 0 and 1 respectively.
3. Construct the noisy and erased codeword $\tilde{y}$ such that the value of $\tilde{y}$ at position $x_l = \kappa_l$ (κ_l is seen here as its value in $GF(2^k)$) is $\tilde{y}(x_l) = (\#\{S_0^l\} - \#\{S_1^l\})ln(\frac{1/2-\epsilon_l}{1/2+\epsilon_l})$. The position were no linear approximation is defined will be put to zero thus considering it as an erasure position.

4. Decode $\tilde{y}$ in the first order Reed-Muller code, i.e. the most probable codeword y is the one that maximise the inner product $\sum_{x\in\{0,1\}^t}(-1)^{y(x)}\tilde{y}(x)$. The Fast Fourrier Transform would do the trick in a time complexity $O(k2^k)$ and data complexity $O(2^k)$.

For details of Reed-Muller decoding efficiency in a gaussian and erasure channel, the interested reader should refer to the results of I. Dumer-R. Krichevskiy in [11].

Remark 6. The attack algorithm is presented for an intermediate value V_i and, of course, all intermediate values $\{V_i\}_{1\leq i\leq C}$ that is reachable by linear approximations can be simultaneously targeted, hence significantly decreasing the attack complexity in term of number of traces.

Practical setup

The attack presented above may seem unrealistic since it uses directly the value measured as Hamming weight of the data manipulated. Three practical setup would allow this:

1. The first solution is the least efficient one but does not assume access to a training board. This is the setup proposed for the results presented in the next section for the attack based on the publicly available consumption traces from the dpa-contest. We suppose that we have access to a set of N consumption traces. From this set, and at each sample, we can assume the consumption values follow a Gaussian law, similar to the Hamming weight of the target intermediate value. This estimation may seem rough but for clean power traces (as they are in the dpa-contest), the simple thing of separating the traces at sample i into two sets (one for a Hamming weight ≥ 32 and one for the others) is enough to get interesting results (see figure 2).
2. The second solution is a very similar setup than the one proposed in [28]: From a training device, where the attacker can set plaintexts and keys and observe the power consumption, he would be able to construct a library of power consumption with respect to the Hamming weight of the targeted value. With such a preprocessing phase, the attacker would have a more efficient attack than in the previous setup.
3. Last but not least, again assuming that the attacker have access to a training device, it would be possible to run the algorithm that search linear approximations directly on the twin device as a pre-processing phase of our attack. As the algorithm is run on a Boolean function taken as a black box, using the consumption measurement as output value of our Boolean function might render the attack even more efficient than in the model presented above: The consumption model (HD or HW) is not necessary anymore, the linear equations are directly approximating the physical leakage function. Further more, with that setup, it is then possible to mount *unknown cipher attacks* since no knowledge of the symmetric cipher is needed except for its iterative structure (the hardware device is seen as a black box from which the consumption leakage is the output).

Results

In this section are presented the results obtained using the above described attacks on the DES. The experiments have been done on real power traces, and validate the practical feasibility of the attack. Table 1 summarizes the results, in the table, "# linear equ." refers to the total number of linear approximations found for the attack, not all of them have been useful, "# Plaintext" or "# Traces" refers to the data complexity of the attack to retrieve "# key bits" of the secret key (over 56 bits).
Thanks to the DPA contest (`http://www.dpacontest.org/`), power consumptions traces of a DES implementation are freely available. These online available traces allowed us to try our attack on real power traces and then prove the feasibility in a real setup of the attack. The attack has been launched on the contest traces (`secmatv1_2006_04_0809`) that yield about 80000 power consumption traces. The linear approximations evaluate the hamming weight of the difference of data stored in the implementation register (LR) its size is equal to the block size (i.e. $m = r = 64$, $k = 1$ with notations of section 2.1). Moreover one complete round of DES is done in one clock cycle, thus $C = 16$. Hence 16 intermediate values V_i are leaking for this implementation (according to our model). A more detailed attack description and setup can be found in Annex 2 of this document. Additional results on simulations are shown in Annexe 3. By the means of Fourquet et al.'s work on finding linear approximations, the two firsts and two lasts rounds outputs differences Hamming weight could be approximated with good enough biases. With about 500 linear approximations we were able to retrieve 40 bits in average (over 56) using 500 traces, when attacking simultaneously on several target values the data complexity has been decreased to about 100. The figure 2 displays the number of key bits that can be retrieved in average with respect to the number of randomly chosen traces (On the figure, the error bars represent the ranges from minimum to maximum number of retrieved key-bits over 1000 tests, the confidence intervals are under 1%).

Remark 7. The attack targeting intermediate values after the second round does not work as well as the others. This is not due to a higher algebraic complexity which would decrease the approximations biases. In fact it can be remarked that the consumption traces are somehow cleaner at the end of the encryption than at the beginning. Also the choice of where to attack in the consumption traces (which sample) might not be the best one, even though we were not able to find a better index.

Remark 8. The DPA-contest is a great success and best DPA attacks (CPA-multi-bits) retrieve the key in less than 100 traces! Although we have to remark that some solutions are using a custom order and fixed set of the traces, hence somehow breaking the statistical analysis complexity. This is not our case since the sets of 500 traces were chosen randomly and the attack launched on thousand of such sets. The results are average results over the different sets.

DES	rounds	# linear equ.	# key bits	# traces
Single intermediate value	1	533	40	500
	2	1452	27	500
	15	488	41	500
	16	446	37	500
Several intermediate values	1||15||16	1467	36	100
	1||15||16	1467	53	500
	1||2||15||16	2919	41	100
	1||2||15||16	2919	54	500

Table 1. Attack on DPA-contest traces Results

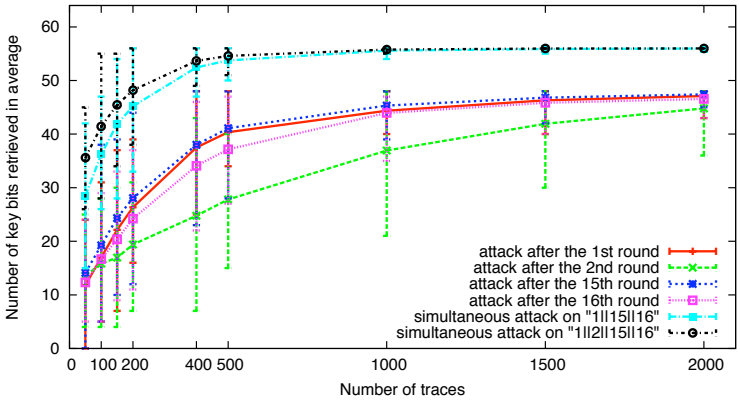

Fig. 2. Complexity of the M-LPA attacks on DES implementation from the dpa-contest

4 Conclusion and future work

The results shown in section 3.3 prove the feasibility of the M-LPA attacks, it is our belief that this set of attacks is a starting point of new results on power analysis attacks on embedded symmetric ciphers. Hence the next steps will be of several kinds:

- The research of better linear approximations in term of bias and which can approximate more rounds of the symmetric cipher. This implies a complexity in time that we did not have for the redaction of this document.
- Complete the work on DES by attacking an 8-bit microcontroler implementation, hence multiplying the intermediate values and then decreasing the data complexity to a minimum.
- The experimentation on an unknown cipher implementation with research of linear approximation directly on the board. This attack may lead to very efficient attacks since it directly approximate the leakage function without using any consumption model.

- Find approximations for other symmetric ciphers (approximations of the last round of AES gave very good results as there is no diffusion).
- On the idea of Renauld and Standaert in [28], the M-LPA could also lead to unknown plaintext-ciphertext attacks and that way bypass boolean masking counter-measure.

References

1. Mehdi-Laurent Akkar and Christophe Giraud. An implementation of des and aes, secure against some attacks. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 2001.
2. Mustafa Badaroglu, Kris Tiri, Stéphane Donnay, Piet Wambacq, Hugo De Man, Ingrid Verbauwhede, and Georges G. E. Gielen. Clock tree optimization in synchronous cmos digital circuits for substrate noise reduction using folding of supply current transients. In *DAC*, pages 399–404. ACM, 2002.
3. Alex Biryukov, Christophe De Cannière, and Michaël Quisquater. On multiple linear approximations. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2004.
4. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.
5. Marco Bucci, Luca Giancane, Raimondo Luzzi, and Alessandro Trifiletti. Three-phase dual-rail pre-charge logic. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 232–241. Springer, 2006.
6. Vincent Carlier, Hervé Chabanne, Emmanuelle Dottax, and Hervé Pelletier. Generalizing square attack using side-channels of an aes implementation on an fpga. In Tero Rissa, Steven J. E. Wilton, and Philip Heng Wai Leong, editors, *FPL*, pages 433–437. IEEE, 2005.
7. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.
8. Zhimin Chen and Yujie Zhou. Dual-rail random switching logic: A countermeasure to reduce side channel leakage. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 242–254. Springer, 2006.
9. Jean-Sébastien Coron, Emmanuel Prouff, and Matthieu Rivain. Side channel cryptanalysis of a higher order masking scheme. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 28–44. Springer, 2007.
10. Nicolas Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In Yuliang Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287. Springer, 2002.
11. Ilya Dumer and Rafail E. Krichevskiy. Soft-decision majority decoding of reed-muller codes. *IEEE Transactions on Information Theory*, 46(1):258–264, 2000.
12. R. Fourquet, P. Loidreau, and C. Tavernier. Finding good linear approximations of block ciphers and its application to cryptanalysis of reduced round des. In *Workshop on Coding Theory and Cryptography, Ullensvang, Norvège*, 2009.
13. B. Gerard and J.-P. Tillich. On linear cryptanalysis with many linear approximations. Technical report, 2007.
14. Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In Elisabeth Oswald and Pankaj Rohatgi, editors, *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.
15. Louis Goubin and Jacques Patarin. Des and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *CHES*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.

16. Sylvain Guilley, Philippe Hoogvorst, and Renaud Pacalet. A fast pipelined multi-mode des architecture operating in ip representation. *Integration*, 40(4):479–489, 2007.
17. Helena Handschuh and Bart Preneel. Blind differential cryptanalysis for enhanced power attacks. In Eli Biham and Amr M. Youssef, editors, *Selected Areas in Cryptography*, pages 163–173, 2006.
18. G. Kabatiansky I. Dumer and C. Tavernier. List decoding of the first order binary reed muller codes. *Problems of Information Transmission*, 43(3):225–232, 2007.
19. Burton S. Kaliski Jr. and Matthew J. B. Robshaw. Linear cryptanalysis using multiple approximations. In Yvo Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 26–39. Springer, 1994.
20. G. Kabatiansky and C. Tavernier. List decoding with reed muller codes of order one. In *nine International Workshop On Algebraic and Combinatorial Coding Theory, Proceedings ACCT-9*, pages 230–236, 2004.
21. G. Kabatiansky and C. Tavernier. List decoding of first order reed-muller codes ii. In *tenth International Workshop On Algebraic and Combinatorial Coding Theory, Proceedings ACCT-10*, 2006.
22. P. Kocher, J. Jaffe, and B. Jun. Introduction to differential power analysis and related attacks. Technical report, Cryptography Research Inc., 1998.
23. Thanh-Ha Le, Jessy Clédière, Cécile Canovas, Bruno Robisson, Christine Servière, and Jean-Louis Lacoume. A proposition for correlation power analysis enhancement. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 174–186. Springer, 2006.
24. Hervé Ledig, Frédéric Muller, and Frédéric Valette. Enhancing collision attacks. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 176–190. Springer, 2004.
25. Stefan Mangard, Thomas Popp, and Maria Elisabeth Oswald. *Power Analysis Attacks - Revealing the Secrets of Smart Cards*. 2007.
26. Mitsuru Matsui. Linear cryptoanalysis method for des cipher. In *EUROCRYPT*, pages 386–397, 1993.
27. Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and countermeasures for smart cards. In Isabelle Attali and Thomas P. Jensen, editors, *E-smart*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer, 2001.
28. Mathieu Renauld and Francois-Xavier Standaert. Algebraic side-channel attacks. Cryptology ePrint Archive, Report 2009/279, 2009. `http://eprint.iacr.org/`.
29. Matthieu Rivain, Emmanuelle Dottax, and Emmanuel Prouff. Block ciphers implementations provably secure against second order side channel analysis. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2008.
30. Kai Schramm, Gregor Leander, Patrick Felke, and Christof Paar. A collision-attack on aes: Combining side channel- and differential-attack. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 163–175. Springer, 2004.
31. Kai Schramm, Thomas J. Wollinger, and Christof Paar. A new class of collision attacks and its application to des. In Thomas Johansson, editor, *FSE*, volume 2887 of *Lecture Notes in Computer Science*, pages 206–222. Springer, 2003.
32. Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure dpa resistant asic or fpga implementation. In *DATE*, pages 246–251. IEEE Computer Society, 2004.
33. Michael Tunstall and Olivier Benoît. Efficient use of random delays in embedded software. In Damien Sauveron, Constantinos Markantonakis, Angelos Bilas, and Jean-Jacques Quisquater, editors, *WISTP*, volume 4462 of *Lecture Notes in Computer Science*, pages 27–38. Springer, 2007.

Annex 1: Linear complexity list decoding of Reed-Muller codes up to their Johnson bound

Preliminaries

We briefly describe the probabilistic list decoding algorithm of [12] for the first order Reed-Muller codes $RM(1,m)$ of length $n=2^m$ correcting up to decoding radius $T>n(1/2-\epsilon)$ with complexity $\mathcal{O}(m^2\epsilon^6\log\frac{1}{\epsilon}(\log m+\log\frac{1}{\epsilon}+\log\frac{1}{P_{err}}))$ in the worst case, where P_{err} is the probability of wrong list decoding.
We recall that $RM(1,m)$ designs the first order binary Reed-Muller codes of length $n=2^m$ which is also the set of m-variate affine Boolean function.

The algorithm of [12] works iteratively in $m+1$ steps. Each step i $(1\leq i\leq m+1)$ corresponds to the determination of the affine Boolean functions on the ith firsts variables that are potential solutions to our problem of approximation.
Let f a m-variate Boolean function. We want to determine the list $L_\epsilon(f)$ of m-variate affine Boolean functions that coincide with f on a fraction greater than $(1/2+\epsilon)n$ of the n possible inputs (i.e. an affine approximation of bias $\geq\epsilon$):

$$L_\epsilon(f)=\{(c_0,c)\in GF(2)\times GF(2)^m|\sum_{x\in GF(2)^m}(-1)^{f(x)\oplus c_0\oplus\langle c|x\rangle}\geq 2\epsilon n\},$$

where $\langle\cdot|\cdot\rangle$ is the usual inner product, and $c=(c_1,\ldots,c_m)$, $x=(x_1,\ldots,x_m)$.
In coding theory, this list can also be defined as

$$L_\epsilon(f)=\{c\in RM(1,m)|d_H(f,c)\leq n(1/2-\epsilon)\}$$

where d_H is the Hamming distance.

List-Decoding Algorithm

Let us detail the list-decoding algorithm steps:

Step number 0**:** An empty list $\mathcal{L}_0^f(\epsilon)$ is defined.

Step number $i-1$**:** We assume that we have already determined a list of elements from $RM(1,m-i+1)$ that contains at least the elements whose coefficients correspond to the first coefficients to the elements from $L_\epsilon(f)$.
If $c\in RM(1,m)$ is in $L_\epsilon(f)$, then the Cauchy-Schwartz inequality implies that

$$\sum_{s\in GF(2)^{m-i}}|\sum_{r\in GF(2)^i}(-1)^{f(r,s)\oplus c_0\oplus\langle c|(r,s)\rangle}|=\sum_{s\in GF(2)^{m-i}}|\sum_{r\in GF(2)^i}(-1)^{f(r,s)\oplus\langle c^{(i)}|r\rangle}|\geq 2\epsilon n,$$

where $c^{(i)} = (c_1, \ldots, c_i)$ and $r = (r_1, \ldots, r_i)$.
Thus in fact the following test is defined: at the step number i, a candidate $c^{(i)}$ is accepted and belongs to the list $\mathcal{L}_i^f(\epsilon)$ if and only if $\sum_{s \in GF(2)^{m-i}} |\sum_{r \in GF(2)^i} (-1)^{f(r,s) \oplus \langle c^{(i)} | r \rangle}| \geq 2\epsilon n$.
By testing exhaustively all elements $c \in RM(1, m-i)$, we are sure that $\mathcal{L}_i^f(\epsilon)$ will contain at least all correct candidates.
Let $\mathcal{L}_{i-1}^f(\epsilon)$ contains at least all correct candidates and $\mathcal{L}_i^f(\epsilon)$ constructed as follows: for any element $c^{(i-1)} \in \mathcal{L}_{i-1}^f(\epsilon)$, we suggest the candidates $(c^{(i-1)}, 0)$, $(c^{(i-1)}, 1)$ and we apply to itself the previous test. By mathematical induction, the algorithm outputs $\mathcal{L}_{m+1}^f(\epsilon) = L_\epsilon(f)$.
Furthermore, the size of the intermediate lists $\mathcal{L}_i^f(\epsilon)$ are upper-bound by $1/2\epsilon^2$ (see [21]).

Finally if m is rather small, this computation can be viewed as an specialized Fast-Fourier-Transform (see [18]).
Otherwise, if m is big, the deterministic sum $\sum_{s \in GF(2)^{m-i}} |\sum_{r \in GF(2)^i} (-1)^{f(r,s) \oplus \langle c^{(i)} | r \rangle}|$ will be transformed in a probabilistic sum $\sum_{s \in S} |\sum_{r \in R} (-1)^{f(r,s) \oplus \langle c^{(i)} | r \rangle}|$ where S and R are respectively some random subsets of $GF(2)^i$ and $GF(2)^{m-i}$ (see [12, 20]). We note that the list size increases exponentially during the first $\mathcal{O}(1/\epsilon^2)$, fortunately, a trick based on FFT computation (see [12]) allows to avoid the worst case complexity. As it is mentioned in [12], practically this algorithm works with a complexity close to $\mathcal{O}(m/\epsilon^2)$

Annex 2: The attack on DPA-context traces setup

This annex describes an M-LPA attack on power traces found on the dpa-contest website: `http://www.dpacontest.org/`. The traces used for our attack are stored under the name `secmatv1_2006_04_0809`, there is 81089 power traces that have been measured from a straightforward DES implementation detailed in [16].

The implementation is described in the figure 3 (from [16]). Let us denote $H(X)$ the Hamming weight function, $IP(X)$ the initial permutation of DES cipher and $DES_n(X, K)$ the first n rounds of the DES encryption on a 64-bits vector X and a $(n \times 64)$-bits K. The power measurement samples we are interested in are the ones corresponding to the load of the register LR, after round 1 and 2 (for the attack on the intermediate value after the second round, the other intermediate values are treated similarly). According to the Hamming Distance model, they should correspond to $H(IP(X)$ `XOR` $DES_1(X, K))$ (noted $C_1(X, K)$) and $H(DES_1(X, K)$ `XOR` $DES_2(X, K))$ (noted $C_2(X, K)$) respectively. The sample indexes were found by just simulating a DPA attack on the first round, the

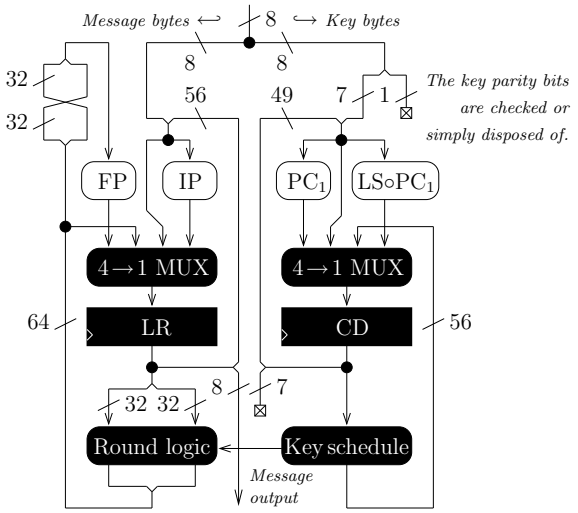

Fig. 3. Schematic of DES implementation

second round, the last round and the round 15. It is our belief that these informations could have been found by an attacker using simple timing measurement, anyways it is a hypothesis of the M-LPA attack that these informations are known. Hence, the load of register LR after the second round was found to be corresponding to the 6374th sample of the power

traces. The figure 4 represent the measured power consumption at sample 6374 for the entire set of traces as a function of the real value of $H(DES_1(X,K)$ XOR $DES_2(X,K))$. It shows clearly the dependence between power consumption and Hamming distance of the intermediate values.

Linear approximations have been generated corresponding to $< C_i(P,K), \Gamma_H >, i \in \{1,2\}$.

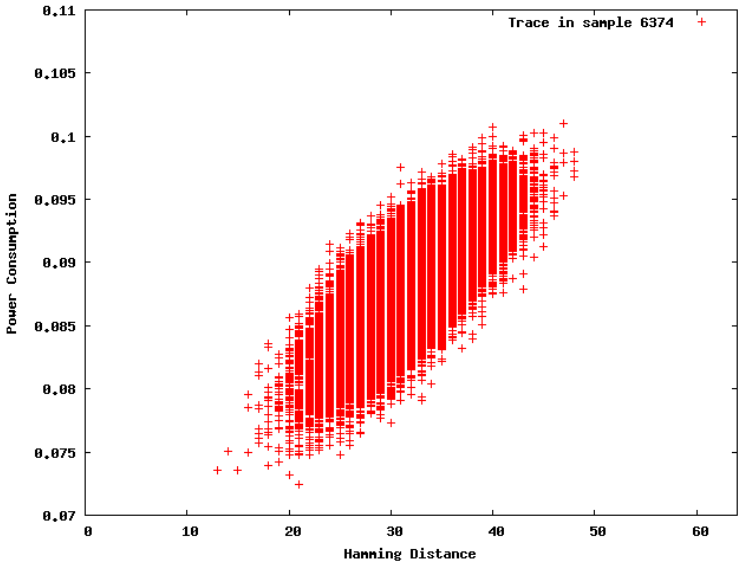

Fig. 4. Repartition of power consumption with respect to the Hamming Distance model

Only the ones where Γ_H equals to $0x10$ or $0x20$ were kept. The Table 2 give an example of 11 of these approximations for the second round (C_2). Over these 11 equation, only 6 key bits are involved ($K[j]$ is the jth bit of the secret key). The last thing we now have to do in order to apply the M-LPA algorithm is a way to tell the value $< C_i(P,K), \Gamma_H >, i \in \{1,2\}$ from the consumption measurement at the selected sample. That is why, to simplify this attack, we only select the output mask (Γ_H) to be $0x10$ or $0x20$ because then, we just have to separate the traces in two, the ones that have power measures greater than the average power measure S_1 and the others S_0, assuming that the power traces in S_1 are such that $< C_i(P,K), 0x20 >= 1$ and $< C_i(P,K), 0x20 >= 0$ for the others. We then assume that the power traces in S_1 are such that $< C_i(P,K), 0x10 >= 0$ and $< C_i(P,K), 0x10 >= 1$ for the others since there is very few chance to have $C_i(P,K) < 0x10$ or $C_i(P,K) \geq 0x30$ from random plaintexts.

Γ_H	Bias	Equation
0x10	0.0219	0 + P[5]+ P[26]+ P[27]+ P[31]+ P[45]+ P[53]+ P[61]+ K[6]+ K[7]+ K[29]+ K[38]+ K[52]
0x20	0.0215	1 + P[5]+ P[26]+ P[27]+ P[31]+ P[45]+ P[53]+ P[61]+ K[6]+ K[7]+ K[29]+ K[38]+ K[52]
0x20	0.0134	0 + P[28]+ P[29]+ P[31]+ P[37]+ P[45]+ P[53]+ K[6]+ K[7]+ K[29]+ K[61]
0x20	0.0156	1 + P[5]+ P[28]+ P[29]+ P[31]+ P[37]+ P[45]+ K[6]+ K[29]+ K[38]+ K[61]
0x10	0.0142	0 + P[5]+ P[28]+ P[29]+ P[31]+ P[37]+ P[45]+ K[6]+ K[29]+ K[38]+ K[61]
0x20	0.0189	1 + P[5]+ P[28]+ P[29]+ P[31]+ P[37]+ P[53]+ K[7]+ K[29]+ K[38]+ K[61]
0x10	0.0189	0 + P[5]+ P[28]+ P[29]+ P[31]+ P[37]+ P[53]+ K[7]+ K[29]+ K[38]+ K[61]
0x10	0.0126	1 + P[26]+ P[27]+ P[37]+ P[45]+ P[53]+ P[61]+ K[6]+ K[7]+ K[52]+ K[61]
0x20	0.0163	0 + P[5]+ P[8]+ P[9]+ P[37]+ P[45]+ P[53]+ P[61]+ K[6]+ K[7]+ K[38]+ K[52]+ K[61]
0x10	0.0167	1 + P[5]+ P[8]+ P[9]+ P[37]+ P[45]+ P[53]+ P[61]+ K[6]+ K[7]+ K[38]+ K[52]+ K[61]
0x10	0.0215	1 + P[5]+ P[14]+ P[15]+ P[31]+ P[37]+ P[45]+ P[61]+ K[6]+ K[29]+ K[38]+ K[52]+ K[61]
0x10	0.0146	0 + P[5]+ P[28]+ P[29]+ P[31]+ P[37]+ P[45]+ P[61]+ K[6]+ K[29]+ K[38]+ K[52]+ K[61]
0x10	0.0148	1 + P[5]+ P[8]+ P[9]+ P[31]+ P[37]+ P[45]+ P[61]+ K[6]+ K[29]+ K[38]+ K[52]+ K[61]
0x20	0.0223	0 + P[5]+ P[14]+ P[15]+ P[31]+ P[37]+ P[45]+ P[61]+ K[6]+ K[29]+ K[38]+ K[52]+ K[61]
0x20	0.0182	0 + P[5]+ P[28]+ P[29]+ P[31]+ P[37]+ P[53]+ P[61]+ K[7]+ K[29]+ K[38]+ K[52]+ K[61]
0x10	0.0152	0 + P[5]+ P[26]+ P[27]+ P[31]+ P[37]+ P[53]+ P[61]+ K[7]+ K[29]+ K[38]+ K[52]+ K[61]
0x10	0.0187	1 + P[5]+ P[28]+ P[29]+ P[31]+ P[37]+ P[53]+ P[61]+ K[7]+ K[29]+ K[38]+ K[52]+ K[61]
0x20	0.0157	1 + P[5]+ P[26]+ P[27]+ P[31]+ P[37]+ P[53]+ P[61]+ K[7]+ K[29]+ K[38]+ K[52]+ K[61]
0x20	0.0191	0 + P[5]+ P[26]+ P[27]+ P[31]+ P[37]+ P[45]+ P[53]+ P[61]+ K[6]+ K[7]+ K[29]+ K[38]+ K[52]+ K[61]
0x10	0.0183	1 + P[5]+ P[26]+ P[27]+ P[31]+ P[37]+ P[45]+ P[53]+ P[61]+ K[6]+ K[7]+ K[29]+ K[38]+ K[52]+ K[61]

Table 2. Attack on DPA-contest traces Results

Annex 3: Attack simulations on DES and AES

Apart from the experiments on real traces, simulations have been run. In simulations it is assumed that the Hamming weight of the intermediate value is precisely retrieved from the power consumption observation. The results are depicted in the following figure 5 with the same linear equations used for the previous attacks on the dpa-contest traces. The results are very similar to the ones obtained from the real traces, thus showing that the error induced by the observation of the dpa-contest power consumption measures is negligible compared to the linear equations' probabilities of error (for all intermediate values but the one after the 2nd round, see remark 7).
Furthermore, the last round of AES has been approximated and attacked, the results are also shown on figure 5. The last round of AES has the good property of having no `Mix column` function (diffusion function of AES). Similarly to the experiments on DES, the simulations assume that the AES implementation executes one round at each clock cycle (i.e. register size is 128-bits). Hence the attack is mounted using the same model than the one for last round DES.

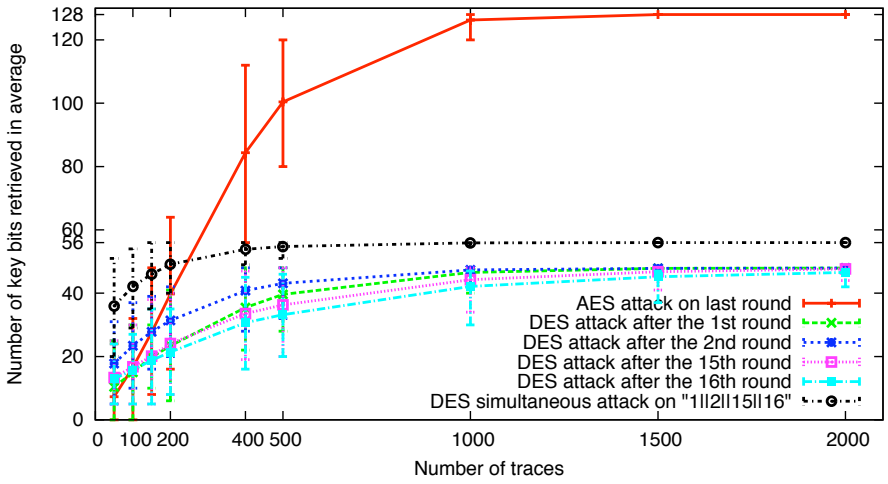

Fig. 5. Complexity of the M-LPA attacks on DES and AES by simulation