Side Channel Cryptanalysis of a Higher Order Masking Scheme

Jean-Sébastien Coron[1], Emmanuel Prouff[2] and Matthieu Rivain[1,2]

[1] University of Luxembourg
Faculty of Sciences, Technology and Communication
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
[2] Oberthur Card Systems,
71-73 rue des Hautes Pâtures,
92726 Nanterre Cedex, France
jean-sebastien.coron@uni.lu,{m.rivain,e.prouff}@oberthurcs.com

Abstract. In the recent years, DPA attacks have been widely investigated. In particular, 2-nd order DPA have been improved and successfully applied to break many masked implementations. In this context a higher order masking scheme has been proposed by Schramm and Paar at CT-RSA 2006. The authors claimed that the scheme is resistant against d-th order DPA for any arbitrary chosen order d. In this paper, we prove that this assertion is false and we exhibit several 3-rd order DPA attacks that can defeat Schramm and Paar's countermeasure for any value of d.

Keywords : Side Channel Attacks, Differential Power Analysis, Higher Order DPA, Masking Scheme, Template Attacks.

1 Introduction

For a long time, cryptographic algorithms have been studied to thwart *mathematical attacks* which try to recover secret keys from some ciphertexts. Big efforts have been made to design resistant algorithms and to prove their security. In recent years, new attacks have been developed that target physical implementations of cryptographic algorithms. Those *physical attacks* are referred to as *side channel attacks* and are often much more efficient than the mathematical attacks.

Side channel attacks exploit information that leaks from physical implementations of cryptographic algorithms. The analysis of this leakage (*e.g.* the power consumption or the electro-magnetic emanations) reveals information on the secret data manipulated by the implementation. Among the side channel attacks, the *Differential Power Analysis* (DPA) [13] is one of the most powerful against unprotected cryptographic implementations: it allows to recover the value of a secret key with only a few leakage measurements. A DPA is a statistical attack that correlates a physical leakage with the values of intermediate variables (called here *sensitive variables*) that depend on both the plaintext and the secret key. To avoid information leakage, the manipulation of sensitive variables must be protected by adding countermeasures to the algorithm.

A very common countermeasure for block ciphers implementations is to randomize sensitive variables by masking techniques [5, 9]. All of these are essentially based on the same principle which can be stated as follows: every sensitive variable Y is randomly split into d shares V_1,..., V_d in such a way that the *completeness relation* $Y = V_1 \star ... \star V_d$ is satisfied for a group operation $\star$ (*e.g.* the x-or or the modular addition). Such a technique, here called d-th order masking, ensures that every single variable is masked with at least one random value and then, a classical (1-st order) DPA attack cannot be successfully carried out anymore. However other attacks, such as the *Higher Order DPA* (HO-DPA) attacks, exist that can defeat d-th order masking.

Higher order DPA are attacks that combine multiple *leakage signals*. When a d-th order masking is used, a d-th order DPA can be performed to combine the leakage signals $L(V_i)$ resulting from the manipulation of the d shares V_i. This enables the construction of a signal that is correlated to the targeted sensitive variable Y. Such an attack can theoretically bypass any d-th order masking. However, the noise effects imply that the difficulty of carrying out a HO-DPA in practice increases exponentially with its order and an attacker has to deal with several issues.

The main issue of HO-DPA is to determine how to combine the d leakage signals $L(V_i)$ in such a way that the combination is highly correlated to the sensitive variable Y. In [5], Chari *et al.* propose to perform the product $L(V_1) \times ... \times L(V_d)$ of d leakage signals. Messerges proposes in [15] another combining method for $d = 2$. It consists in processing the absolute value of the difference of the two leakage signals $|L(V_1) - L(V_2)|$. This can be generalized to the d-th order as $|L(V_1) - ... \ |L(V_{d-1}) - L(V_d)|...|$. Such attacks, which combine several leakage signals, will be called *Combining HO-DPA* in this paper.

An alternative to these attacks exists when the attacker is allowed to profile the leakage in order to exhibit a relationship between the statistical distribution of the leakage and the value of a sensitive variable. Once this relationship is determined, the *likelihood* of key guesses is estimated given the distribution of the uplet $(L(V_1), \cdots, L(V_d))$. Such attacks are based on the same principle as the *Template attacks* introduced by Chari *et al.* in [6]. They have been successfully applied by Peeters *et al.* in [19] and by Oswald *et al.* in [17] to break some masked implementations more efficiently than any combining 2-nd order DPA. In this paper we will call *Profiling HO-DPA* any HO-DPA attack that assumes a profiling of the leakage.

The recent works [1, 10, 17–19, 24, 23, 26] show that 2-nd order DPA attacks not only allow to theoretically invalidate some countermeasures, but can sometimes break them in practice. HO-DPA of order greater than 2 will also likely become a real practical threat in foreseeable future. Therefore, there is a need for countermeasures thwarting not only 2-nd order DPA but more generally d-th order DPA for $d > 2$.

At CT-RSA 2006, Schramm and Paar propose in [23] a higher order masking scheme of AES which aims to thwart d-th order DPA for any d. However, we show in the present paper (Sections 3 and 4) that Schramm and Paar's Scheme admits several flaws which actually make it vulnerable to 3-rd order DPA for any value of d. Therefore, as opposed to what is claimed in [23], the countermeasure does not protect against d-th order DPA for $d \geq 3$. In Section 5, the flaws of Schramm and Paar's Scheme are used to exhibit 3-rd order DPA attacks. Simulations are provided that demonstrate the practicability of our attacks.

2 Preliminaries

DPA attacks exploit a dependency between a subpart of the secret key and the variations of a physical leakage as function of the plaintext. This dependency results from the manipulation of some sensitive variables by the implementation. We say that a variable is sensitive if it depends on both the plaintext and the secret key. For example, the x-or between a key byte and a plaintext byte is a sensitive variable.

If an algorithm manipulates a sensitive variable directly, then a physical implementation of this algorithm can be broken by a 1-st order DPA. The implementation can be rendered resistant against 1-st order DPA by masking every sensitive variable with a single random mask. However a higher order DPA is still possible. The next definition formalizes the notion of security with respect to d-th order DPA for a cryptographic algorithm.

Definition 1. *A cryptographic algorithm $\mathcal{A}$ is secure against d-th order DPA if every family of at most d intermediate variables of $\mathcal{A}$ is independently distributed from any sensitive variable.*

If a family of d intermediate variables depends on a sensitive variable then we say that the algorithm admits a *d-th order flaw*. A DPA attack that exploits such a flaw is a d-th order DPA. In Sections 3 and 4, we recall the Schramm and Paar's Scheme and we show that it has 3-rd order flaws.

In the rest of the paper, we will use the calligraphic letters, like $\mathcal{X}$, to denote finite sets. The corresponding large letter X will then be used to denote a random variable over $\mathcal{X}$, while the lowercase letter x - a particular element from $\mathcal{X}$.

3 The Generic Masking Scheme

3.1 Description

Schramm and Paar propose in [23] a masking scheme for AES [7] which aims to thwart d-th order DPA for any arbitrary chosen d. Every sensitive byte Y appearing in the algorithm is never directly manipulated and is represented by $d+1$ values M_0, M_1, ..., M_d. To ensure the DPA-resistance, the shares $(M_i)_{i\geq 1}$ take random values and to ensure completeness, M_0 satisfies

$$M_0 = Y \oplus \bigoplus_{i=1}^{d} M_i \quad . \tag{1}$$

When a transformation S must be applied to Y, $d+1$ new values N_0, N_1, ..., N_d must be processed from the M_i's such that

$$N_0 = S(Y) \oplus \bigoplus_{i=1}^{d} N_i \quad . \tag{2}$$

The critical point of such a method is to deduce the N_i's from the M_i's when S is non-linear, without compromising the security of the scheme against d-th order DPA.

To tackle this issue, Schramm and Paar propose to adapt a method, called *table re-computation*, which has been widely used to protect implementations against 1-st order DPA (see for instance [14, 2]). In their proposal, the d output masks $(N_i)_{i\geq 1}$ are randomly generated and a new table S^* is derived from M_1, ..., M_d and N_1, ..., N_d in such a way that S^* satisfies for every x:

$$S^*(x) = S\left(x \oplus \bigoplus_{i=1}^{d} M_i\right) \oplus \bigoplus_{i=1}^{d} N_i \quad . \tag{3}$$

Then, one lets $N_0 \leftarrow S^*(M_0)$; using (1) this gives $N_0 = S(Y) \oplus \bigoplus_{i=1}^{d} N_i$ as required.

To ensure that the design of $S^\star$ induces no flaw with respect to d-th order DPA, it involves d successive table re-computations from $S_0 = S$ to $S_d = S^*$. For every $j \in \{1, \cdots, d\}$, the j-th re-computation produces a new S-Box S_j from S_{j-1} such that for every x:

$$S_j(x) = S_{j-1}(x \oplus M_j) \oplus N_j = S\left(x \oplus \bigoplus_{i=1}^{j} M_i\right) \oplus \bigoplus_{i=1}^{j} N_i \ , \tag{4}$$

which for $j = d$ satisfies (3).

In [23], different table re-computation algorithms are proposed. The attack described in this paper focus on the straightforward algorithm recalled below. We discuss the security of the other algorithms in Appendix A.

Algorithm 1 re-computation

Input: the look-up table S_{j-1}, the input mask M_j, the output mask N_j
Output: the look-up table S_j

```
for $x$ from $0$ to $255$ do
    $S_j(x) \leftarrow S_{j-1}(x \oplus M_j) \oplus N_j$
end
```

3.2 The 3-rd Order Flaw

Before describing the flaw, and to simplify the presentation, we will denote $M = \bigoplus_{i=1}^{d} M_i$ and $N = \bigoplus_{i=1}^{d} N_i$.

During the re-computation of S_d from S_{d-1}, the variables $S_d(0) = S(M) \oplus N$ and $S_d(1) = S(M \oplus 1) \oplus N$ are respectively manipulated during the first iteration and the second iteration of the loop (see Algorithm 1). The manipulation of these two variables together with M_0 induces a 3-rd order flaw. In fact, recalling that M_0 satisfies $M_0 = Y \oplus M$, we have

$$(M_0, S_d(0), S_d(1)) = (Y \oplus M, S(M) \oplus N, S(M \oplus 1) \oplus N) \ . \tag{5}$$

It can be checked from (5) that $(M_0, S_d(0), S_d(1))$ and Y are not independent, which implies that a 3-rd order DPA is potentially feasible. Namely, given $S_d(0)$ and $S_d(1)$, one can compute $\Delta = S_d(0) \oplus S_d(1) = S(M) \oplus S(M \oplus 1)$. This allows to recover M with high probability since the number of values z satisfying $\Delta = S(z) \oplus S(z \oplus 1)$ is small when S has good cryptographic properties (*e.g.* this equation admits at most 4 solutions if S is the AES S-Box). Then, knowing the value of M allows to recover Y from M_0 since they satisfy $Y = M_0 \oplus M$.

The discussion above demonstrates that the use of Algorithm 1 to perform the table re-computations makes Schramm and Paar's Countermeasure vulnerable to 3-rd order DPA for any value d.

Even if the 3-rd order flaw above has been exhibited for the first and the second loop iterations, the generic scheme admits more generally a flaw $(M_0, S_d(e_1), S_d(e_2))$ for every pair $(e_1, e_2) \in \{0, .., 255\}^2$ of loop indices such that $e_1 \neq e_2$.

The importance of the 3-rd order flaw depends on the amount of information that $(M_0, S_d(e_1), S_d(e_2))$ provides about Y. As proved in Appendix B, this amount depends on the cryptographic properties of S and on the value $e_1 \oplus e_2$. In fact for every S-Box S defined from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$ and for every sub-set $\{e_1, e_2\} \subseteq \mathbb{F}_2^n$, the *mutual information* $\mathcal{I}(Y, (M_0, S_d(e_1), S_d(e_2)))$ between Y and $(M_0, S_d(e_1), S_d(e_2))$ satisfies

$$n - \log(\delta) \leq \mathcal{I}(Y, (M_0, S_d(e_1), S_d(e_2))) \leq n \ , \qquad (6)$$

where δ denotes $\max_{e \in \mathbb{F}_2^{n*}, z \in \mathbb{F}_2^m} \{x \in \mathbb{F}_2^n;\ S(x) \oplus S(x \oplus e) = z\}$ (see Proposition 2 in Appendix B).

To resist against differential cryptanalysis [3], the AES S-Box ($n = 8$, $m = 8$) has been designed in such a way that $\delta = 4$. Consequently, if S is the AES S-Box then (6) implies that the mutual information between Y and $(M_0, S_d(e_1), S_d(e_2))$ is lower bounded by 6. In fact, we computed that this mutual information equals $7 - \frac{1}{64} \approx 6.98$ for every sub-set $\{e_1, e_2\} \subseteq \mathbb{F}_2^n$, which means that knowing the values of M_0, $S_d(e_1)$ and $S_d(e_2)$ reveals almost 7 bits of Y (out of 8).

4 The Improved Masking Scheme

4.1 Description

Schramm and Paar's generic Scheme recalled in Section 3.1 is very costly as it involves d table re-computations for each S-Box access for each round of the cipher (which implies $160 \times d$ table re-computations for AES).

Therefore, Schramm and Paar propose in [23] an improvement of the method. In the new solution, d successive re-computations are still preformed to process the first masked S-Box in the first round. Then, each time S must be applied on a new byte $M_0' = Y' \oplus \bigoplus_{i=1}^{d} M_i'$, a new masked S-Box S_{new}^*, satisfying $S_{new}^*(x) = S(x \oplus \bigoplus_{i=1}^{d} M_i') \oplus \bigoplus_{i=1}^{d} N_i'$ for every byte x, is derived from the previous S^* with a single re-computation. This re-computation firstly requires to process two values called *chains of masks* in [23] and denoted here by ICM and OCM:

$$ICM = \bigoplus_{i=1}^{d} M_i \oplus \bigoplus_{i=1}^{d} M'_i \ , \tag{7}$$

$$OCM = \bigoplus_{i=1}^{d} N_i \oplus \bigoplus_{i=1}^{d} N'_i \ . \tag{8}$$

Once the values of the chains of masks have been computed, the masked S-Box S^*_{new} is derived from S^* by performing one single re-computation such that the following relation is satisfied for every x:

$$S^*_{new}(x) = S^*(x \oplus ICM) \oplus OCM \ . \tag{9}$$

To construct a S-Box S^*_{new} that satisfies (9), a re-computation algorithm may be called with the input parameters (S^*, ICM, OCM). The variable ICM removes the previous sum of input masks $\bigoplus_{i=1}^{d} M_i$ and adds the new sum of input masks $\bigoplus_{i=1}^{d} M'_i$ while OCM removes the previous sum of output masks $\bigoplus_{i=1}^{d} N_i$ and adds the new sum of output masks $\bigoplus_{i=1}^{d} N'_i$.

For the whole AES implementation, this improved scheme replaces the $160 \times d$ table re-computations required in the generic scheme by $d + 159$ table re-computations. For $d \geq 2$, this represents a substantial gain.

4.2 The 3-rd Order Flaws

Here we show that the computation of the chains of masks induces two 3-rd order flaws. In fact, one obtains from (1) and (7) that the input chain of masks ICM satisfies

$$Y \oplus Y' = ICM \oplus M_0 \oplus M'_0 \ . \tag{10}$$

Since $Y \oplus Y'$ is a sensitive variable (because it depends on both the plaintext and the secret key), and since the variables ICM, M_0 and M'_0 are manipulated by the implementation, this immediately gives a 3-rd order flaw.

The second 3-rd order flaw is derived as follows: from (2) and (8) we deduce that the output chain of masks OCM satisfies

$$S(Y) \oplus S(Y') = OCM \oplus N_0 \oplus N'_0 \ . \tag{11}$$

This shows that the manipulation of OCM, N_0 and N_0' gives a 3-rd order flaw which leaks information on the sensitive variable $S(Y) \oplus S(Y')$.

To summarize, we have shown that the improved Schramm and Paar's countermeasure is vulnerable to 3-rd order DPA for any value of d.

5 The 3-rd Order DPA Attacks

In previous sections, we have shown that an attacker who can obtain the exact values of 3 intermediate variables of the (generic or improved) Schramm and Paar's masking Scheme, can recover the value (or a part of the value) of a sensitive variable. This is sufficient to show that the countermeasure is theoretically vulnerable to 3-rd order DPA. However, the physical leakage of an implementation does not reveal the exact values of the variables manipulated but a noisy function of them. Thus, a leakage model must be considered when DPA attacks are addressed. In this section, we firstly recall two generic d-th order DPA attacks in a classical leakage model. Then we apply each of them against Schramm and Paar's Countermeasure and we present experimental results.

5.1 Leakage Model

We assume that the physical leakage $L(V_t)$ resulting from the manipulation of a variable V_t at a time t satisfies

$$L(V_t) = \varphi_t(V_t) + B_t \ , \tag{12}$$

where $\varphi_t(V_t)$ is the deterministic leakage of V_t and B_t is a noise. In the sequel, we refer to the φ_t as *leakage functions*.

In the next section, two generic d-th order DPA attacks are described for the leakage model (12). Both of them assume that there exists a d-uplet $(V_1, ..., V_d)$ of variables manipulated by the algorithm which is correlated to a sensitive variable $Y = f(X, K)$. The V_i's depend on a part of the plaintext X, on a part of the secret key K and possibly on random values generated during the execution of the algorithm. The random values involved in the V_i's are represented by a random variable R which is assumed to be uniformly distributed over $\mathcal{R}$. Thus, the V_i variables considered in the rest of the paper can be expressed as functions of (X, K, R), which will be denoted $V_i(X, K, R)$.

5.2 Two Generic Higher Order DPA

We recall hereafter two generic d-th order DPA attacks: the *combining higher order DPA* and the *profiling higher order DPA*. In the first one, the attacker combines the d leakage signals and performs a 1-st order DPA on the obtained combined signal. The second one assumes a stronger adversary model where the attacker is able to profile the implementation leakage. Once it is computed, the profile is involved to launch an optimal probabilistic attack.

Combining Higher Order DPA. A combining d-th order DPA first applies a *combining function* $\mathcal{C}$ (*e.g.* the product or the absolute difference -see Section 1-) to the d leakage signals $L(V_1)$, ..., $L(V_d)$. Then it uses classical DPA techniques (see for instance [4]) to exhibit a correlation between the combined signal $\mathcal{C}\left(L(V_1), ..., L(V_d)\right)$ and the *prediction* P_k of this signal, according to a guess k on the value of the targeted key part K. To perform such a prediction, the attacker needs a mathematical representation of the leakage functions φ_i. Usually, he supposed that $\varphi_i(v)$ is an affine function of the Hamming weight $H(v)$ for every pair (i, v). Thus, we will consider in the sequel that for every $(k, x) \in \mathcal{K} \times \mathcal{X}$ the attacker prediction equals the expected value of the random variable $\mathcal{C}\left(H\left(V_1\left(x, k, R\right)\right), .., H\left(V_d\left(x, k, R\right)\right)\right)$ when R ranges over $\mathcal{R}$:

$$P_k(x) = E_R\left[\mathcal{C}\left(H\left(V_1\left(x, k, R\right)\right), .., H\left(V_d\left(x, k, R\right)\right)\right)\right] \ . \tag{13}$$

The attack consists in the following steps:

1. Perform the leakage measurements $(l_j(v_1), .., l_j(v_d))_{j=1..N}$ corresponding to random plaintexts $(x_j)_{j=1..N}$.
2. For every $x \in \mathcal{X}$, process the *average leakage*:

$$A(x) = \frac{1}{\#\{j \mid x_j = x\}} \sum_{\substack{j=1 \\ x_j = x}}^{N} \mathcal{C}(l_j(v_1), .., l_j(v_d)) \ . \tag{14}$$

3. For every key guess $k \in \mathcal{K}$, compute the *empirical correlation coefficient* ρ_k between the prediction and the average leakage:

$$\rho_k = \frac{2^n \sum_x P_k(x) \cdot A(x) \ - \ \sum_x P_k(x) \cdot \sum_x A(x)}{\sqrt{2^n \sum_x P_k(x)^2 - (\sum_x P_k(x))^2}\sqrt{2^n \sum_x A(x)^2 - (\sum_x A(x))^2}} \ . \tag{15}$$

4. Select the key guess k such that ρ_k is maximal.

Profiling Higher Order DPA. In a profiling attack (see for instance [6, 22]), the attacker has unrestricted access to an implementation for which he knows all the parameters (*i.e.* the plaintext, the secret key and eventually the random values generated). The attack consists in two steps. In the first step (the profiling step), the leakage functions and the noises are characterized *via* the implementation under control. This allows to precisely estimate the leakage distribution according to some manipulated variables. In the second step, the leakage of the implementation under attack is measured and a *maximum likelihood test* [8] is performed to recover the secret parameter (namely the secret key).

We assume hereafter that the profiling step provides the attacker with the exact distribution $(L(V_i))_i$ of the leakage corresponding to the manipulation of the V_i's. The knowledge of this distribution allows him to compute the probability density function $f(.|x,k)$ of $(L(V_i))_i$ given $X = x$ and $K = k$. As the V_i's satisfy (12) for every i, assuming that the B_i's have independent Gaussian distributions, $f(.|x,k)$ satisfies

$$f(l(v_1),..,l(v_d)|x,k) = \frac{1}{\#\mathcal{R}} \sum_{r\in\mathcal{R}} \prod_{i=1}^{d} \phi_\sigma\left(l(v_i) - \varphi_i(V_i(x,k,r))\right) \ , \qquad (16)$$

where $\#\mathcal{R}$ denotes the cardinality of $\mathcal{R}$ and ϕ_σ denotes the probability density function of the Gaussian distribution $\mathcal{N}(0,\sigma)$ which satisfies $\phi_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)$.

Then, the attack consists in the following steps:

1. Perform the leakage measurements $(l_j(v_1),..,l_j(v_d))_{j=1,..,N}$ corresponding to random plaintexts $(x_j)_{j=1,..,N}$.
2. For every $k \in \mathcal{K}$, process the likelihood $\mathcal{L}(k|(l_j,x_j)_j)$ of the key guess k given the observations of the leakage $(l_j(v_1),..,l_j(v_d))_{j=1,..,N}$ corresponding to the plaintexts $(x_j)_{j=1,..,N}$:

$$\mathcal{L}(k|(l_j,x_j)_j) = \prod_{j=1}^{N} f(l_j(v_1),..,l_j(v_d)|x_j,k) \ . \qquad (17)$$

3. Select the key guess k such that $\mathcal{L}(k|(l_j,x_j)_j)$ is maximal.

5.3 Application to Schramm and Paar's Scheme

We launch hereafter the two attacks described in Section 5.2 against the Schramm and Paar's countermeasure recalled in Sections 3 and 4. Each attack is a 3-rd order DPA targeting three variables V_1, V_2 and V_3 appearing during the computation. The measurements $(l_j(v_1), l_j(v_2), l_j(v_3))_j$ are simulated according to a noisy Hamming weight model. Thus for our simulations, the leakage is assumed to satisfy

$$L(V_i) = \varepsilon H(V_i) + B_i \ , \tag{18}$$

where the B_i's have independent Gaussian distributions $\mathcal{N}(0, \sigma)$. The coefficient ε is set to 3.72 and the noise standard deviation σ is set to 1.96[3].

For the combining 3O-DPA attacks, we selected among the product and the absolute difference, the combining function which allows the most efficient attack.

Before presenting the attacks, we recall that during the first round, every input Y of the S-Box S satisfies $Y = X \oplus K$, where X is a plaintext byte and K is a secret key byte.

Attacks on the Generic Scheme. We have shown in Section 3.2 that a 3-rd order flaw results from the manipulation of $V_1 = M_0$, $V_2 = S_d(e_1)$ and $V_3 = S_d(e_2)$. Hereafter, we apply our attacks for $e_1 = 0$ and $e_2 = 1$. In this case, we recall that V_1, V_2 and V_3 satisfy:

$$\begin{aligned} V_1(X, K, R) &= X \oplus K \oplus M \ , \\ V_2(X, K, R) &= S(M) \oplus N \ , \\ V_3(X, K, R) &= S(M \oplus 1) \oplus N \ . \end{aligned}$$

where R denotes the pair (M, N) of involved random masks.

Figure 1 shows the result of a combining 3O-DPA which uses the product as combining function to exploit the flaw. The different curves represent the different key guesses; the curve corresponding to the correct key guess is plotted in black. We noticed that this curve also corresponds to three other wrong key hypotheses (additionally, four wrong key hypotheses result in correlation peaks with equal magnitude and opposite sign). It can be observed that the correlation for the correct key guess comes out after about 4.10^6 measurements. This implies that several millions of measurements are required to recover the secret key byte. However this assertion must be mitigated. Indeed, we noticed that the correlation curve corresponding to the correct key guess is quickly among the top curves, which implies a significant loss of entropy for the secret key value.

[3] These values are the ones used by Schramm and Paar in their experiments [23].

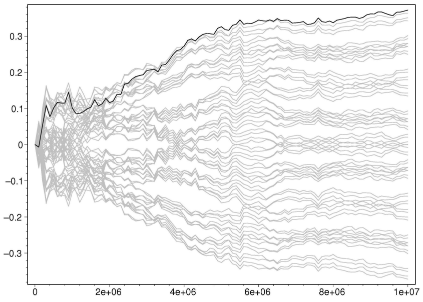

Fig. 1. Combining 3O-DPA : evolution of the correlation (ordinate axis) over an increasing number of measurements (abscissa axis).

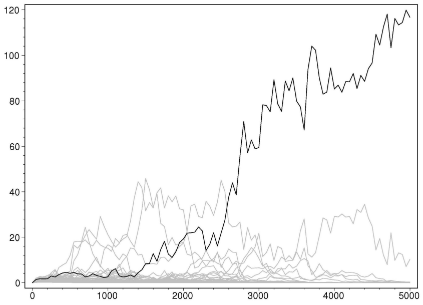

Fig. 2. Profiling 3O-DPA : evolution of the likelihood (ordinate axis) over an increasing number of measurements (abscissa axis).

Figure 2 shows the results of a profiling 3O-DPA. The likelihood of the correct key guess is clearly remarkable after 2800 measurements which shows that the profiling 3O-DPA is much more efficient than the combining 3O-DPA.

These attacks allow to recover the value of the targeted key byte K. They must be performed 16 times to recover the whole first round key.

Attacks on the Improved Scheme. As argued in Section 4.2, a 3-rd order flaw results from the manipulation of $V_1 = ICM$, $V_2 = M_0$ and $V_3 = M_0'$. We recall that these 3 variables satisfy

$$\begin{aligned} V_1(X'', K'', R) &= X'' \oplus K'' \oplus M_0 \oplus M_0' \ , \\ V_2(X'', K'', R) &= M_0 \ , \\ V_3(X'', K'', R) &= M_0' \ . \end{aligned}$$

where X'' denotes the plaintext part $X \oplus X'$, K'' denotes the secret key part $K \oplus K'$ and R denotes the pair (M_0, M_0') of involved random masks.

The flaw above corresponds to a “standard” 3-rd order flaw since the sensitive variable $X'' \oplus K''$ is masked with two random masks (M_0 and M_0').

Figure 3 shows the result of a combining 3O-DPA which uses the absolute difference as combining function and Figure 4 shows the result of a profiling 3O-DPA. The combining 3O-DPA allows to recover the targeted secret key part with 2.10^5 measurements, whereas the profiling 3O-DPA only requires 600 measurements.

These attacks allow to recover the value of the targeted key part $K'' = K \oplus K'$, where K and K' correspond to two successive key bytes.

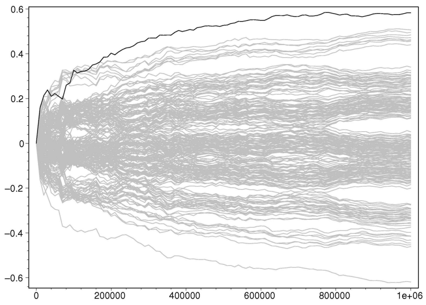

Fig. 3. Combining 3O-DPA : evolution of the correlation (ordinate axis) over an increasing number of measurements (abscissa axis).

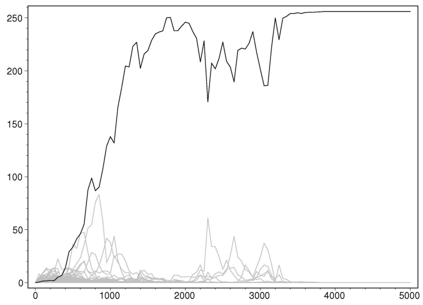

Fig. 4. Profiling 3O-DPA : evolution of the likelihood (ordinate axis) over an increasing number of measurements (abscissa axis).

As for the attacks against the generic scheme, the entropy of the round key is decreased by 8 bits. If performed for the 15 pairs of successive key bytes, the attacks decrease the entropy of the first round key by 120 bits and an exhaustive search can be carried out to recover the remaining 8 bits.

Implementation	Attack	Measurements
No countermeasure	DPA	100
S&P generic scheme	combining 3O-DPA	6.10^6
S&P generic scheme	profiling 3O-DPA	2.10^3
S&P improved scheme	combining 3O-DPA	10^5
S&P improved scheme	profiling 3O-DPA	10^3

Table 1. Number of measurements required to achieve a success rate of 50%.

Results Analysis. We performed each attack 100 times and we recorded the obtained success rates[4]. Table 1 summarizes the number of measurements required to reach a success rate equal to 50%. We list hereafter our observations:

- The most efficient of our 3O-DPA requires a number of measurements which is only 10 times larger than for a 1-st order DPA against an unprotected implementation.

[4] A success is obtained if the attack selects the correct key guess.

- The profiling 3O-DPA is much more efficient than the combining 3O-DPA. This result was predictable. Indeed, the profiling 3O-DPA exploits all the information provided by the 3 leakage signals to derive the likelihood of a key candidate, whereas combining the 3 leakage signals in a single signal implies a significant loss of information whatever the combining function. However, the adversary model of profiling 3O-DPA is very strong and in such a model, an attacker may break an implementation without exploiting the kind of flaws exhibited in the paper.
- The profiling 3O-DPA requires a quite small number of measurements. This shows the practicability of such an attack when the attacker owns a profile that matches well the real leakage of the implementation.
- The combining 3O-DPA is fairly efficient against the improved scheme but is less suitable against the generic scheme. This is not surprising: combining techniques have been especially designed to attack Boolean masking and the flaw in the improved scheme involves a doubly masked variable and two Boolean masks. The flaw in the generic scheme has not this particularity and the combining techniques involved in this paper are less appropriate to exploit it.

6 Conclusion

In this paper, we have exhibited several flaws in Schramm and Paar's higher order masking scheme that makes it vulnerable to 3-rd order DPA. In particular, the general approach consisting in processing d table recomputations has been invalidated. Indeed, we have pointed out that such an approach is vulnerable to 3-rd order DPA. We have also invalidated the Schramm and Paar's improvement of the general approach and we have argued that its use also makes the countermeasure vulnerable to 3-rd order DPA. Finally, simulations have been provided which show the practicability of our attacks. To summarize, the scheme is always vulnerable to 3-rd order DPA for any value of d, but it can be used for $d = 2$ to thwart 2-nd order DPA.

The conclusion of this paper is that the design of a higher order DPA-resistant scheme is still an open problem. Moreover, we think that the DPA-resistance of the future proposals should be proved as other security properties. This field needs to be more investigated to determine the best efficiency/security trade-offs.

Acknowledgements

We would like to thank Christophe Giraud as well as the anonymous referees of CHES 2007 for their fruitful comments and suggestions on this paper.

References

1. D. Agrawal, J. R. Rao, P. Rohatgi, and K. Schramm. Templates as master keys. In Rao and Sunar [21], pages 15–29.
2. M.-L. Akkar and C. Giraud. An Implementation of DES and AES, Secure against Some Attacks. In Ç. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 2001.
3. E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.
4. E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In Joye and Quisquater [11], pages 16–29.
5. S. Chari, C. Jutla, J. Rao, and P. Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In Wiener [27], pages 398–412.
6. S. Chari, J. Rao, and P. Rohatgi. Template Attacks. In Kaliski Jr. et al. [12], pages 13–29.
7. FIPS PUB 197. *Advanced Encryption Standard*. National Institute of Standards and Technology, 2001.
8. R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society*, 1922.
9. L. Goubin and J. Patarin. DES and Differential Power Analysis – The Duplication Method. In Ç. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES '99*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.
10. M. Joye, P. Paillier, and B. Schoenmakers. On Second-Order Differential Power Analysis. In Rao and Sunar [21], pages 293–308.
11. M. Joye and J.-J. Quisquater, editors. *Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*. Springer, 2004.
12. B. Kaliski Jr., Ç. Koç, and C. Paar, editors. *Cryptographic Hardware and Embedded Systems – CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*. Springer, 2002.
13. P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In Wiener [27], pages 388–397.
14. T. Messerges. Securing the AES Finalists Against Power Analysis Attacks. In B. Schneier, editor, *Fast Software Encryption – FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 2000.
15. T. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant software. In Ç. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume 1965 of *Lecture Notes in Computer Science*, pages 238–251. Springer, 2000.

16. K. Nyberg. Differentially uniform mappings for cryptography. In T. Helleseth, editor, *Advances in Cryptology – EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 55–64. Springer, 1993.
17. E. Oswald and S. Mangard. Template attacks on masking–resistance is futile. In M. Abe, editor, *Topics in Cryptology – CT-RSA 2007*, Lecture Notes in Computer Science, pages 562–567. Springer, 2007.
18. E. Oswald, S. Mangard, C. Herbst, and S. Tillich. Practical Second-Order DPA Attacks for Masked Smart Card Implementations of Block Ciphers. In Pointcheval [20].
19. E. Peeters, F.-X. Standaert, N. Donckers, and J.-J. Quisquater. Improving Higher-Order Side-Channel Attacks with FPGA Expermiments. In Rao and Sunar [21], pages 309–321.
20. D. Pointcheval, editor. *Topics in Cryptology – CT-RSA 2006*, volume 3860 of *Lecture Notes in Computer Science*. Springer, 2006.
21. J. Rao and B. Sunar, editors. *Cryptographic Hardware and Embedded Systems – CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*. Springer, 2005.
22. W. Schindler, K. Lemke, and C. Paar. A stochastic model for differential side channel cryptanalysis. In Rao and Sunar [21].
23. K. Schramm and C. Paar. Higher Order Masking of the AES. In Pointcheval [20], pages 208–225.
24. F.-X. Standaert, E. Peeters, and J.-J. Quisquater. On the masking countermeasure and higher-order power analysis attacks. In *ITCC '05: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume I*, pages 562–567. IEEE Computer Society, 2005.
25. E. Trichina, D. DeSeta, and L. Germani. Simplified Adaptive Multiplicative Masking for AES. In Kaliski Jr. et al. [12], pages 187–197.
26. J. Waddle and D. Wagner. Toward Efficient Second-order Power Analysis. In Joye and Quisquater [11], pages 1–15.
27. M. Wiener, editor. *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.

A Further Re-computation Algorithms

In this appendix we focus on the different re-computation algorithms given by Schramm and Paar in [23] and we analyze how they impact the security of the Schramm and Paar's countermeasure recalled in Sections 3.1 and 4.1.

In [23], a variant of Algorithm 1 is given in which Step 2 is replaced by

$$S_j(x \oplus M_j) \leftarrow S_{j-1}(x) \oplus N_j \ .$$

If this variant is used in Schramm and Paar's countermeasure, the 3-rd order flaw presented in Section 3.2 becomes a 4-th order flaw. Indeed, the values stored in memory during the first and the second loop iteration of the d-th table re-computation are not more $S_d(0)$ and $S_d(1)$ but $S_d(M_d)$ and $S_d(M_d \oplus 1)$. The two last variables satisfy

$$S_d(M_d) = S(M \oplus M_d) \oplus N \quad \text{and} \quad S_d(M_d \oplus 1) = S(M \oplus M_d \oplus 1) \oplus N \ .$$

Thus, by analogy with Section 3.2, knowing the values of these two variables reveals information about $M \oplus M_d$ (instead of M in Section 3.2). Therefore, in addition to these two variables, an attacker needs to target not only $M_0 = Y \oplus M$ but also M_d in order to unmask Y. This results in a 4-th order flaw.

Schramm and Paar recall in [23] another algorithm which has been introduced in [25]. However, this algorithm is not suitable as its execution time depends on the input mask value. Such a dependency induces a flaw with respect to 1-st order DPA. Indeed, as the re-computation duration depends on the mask value, the manipulation date of the masked variable after the re-computation also depends on the mask value. This implies that the distribution of the mask given the manipulation date of the masked variable is not uniform. Consequently, a first order flaw occurs at this date.

Finally, Schramm and Paar propose in [23] a new table re-computation algorithm. This algorithm does not require to allocate memory for the output table because it modifies the input table itself to compute the new one.

Algorithm 2 Schramm and Paar's re-computation

INPUT: the look-up table S^*, the input mask M_j, the output mask N_j
OUTPUT: the modified look-up table S^*

```
1. $l = \lfloor \log_2(M_j) \rfloor$
2. for $x_1$ from 0 to 255 by $2^{l+1}$ do
3.     for $x_2$ from 0 to $2^l - 1$ do
4.         $A \leftarrow S^*(x_1 \oplus x_2) \oplus N_j$
5.         $B \leftarrow S^*(x_1 \oplus x_2 \oplus M_j) \oplus N_j$
6.         $S^*(x_1 \oplus x_2) \leftarrow B \oplus N_j$
7.         $S^*(x_1 \oplus x_2 \oplus M_j) \leftarrow A \oplus N_j$
8.     end
9. end
```

Despite its practical interest, this algorithm cannot be used because it does not take the case $M_j = 0$ into account. This is problematic since the mask M_j must be uniformly distributed to ensure the DPA-resistance. Moreover Algorithm 2 cannot be patched to take this case into account. Indeed, when M_j equals 0, the re-computation should apply the output mask N_j to every value in the table : $S^*(x) \leftarrow S^*(x) \oplus N_j$. However, for $M_j = 0$ and whatever the value of l, it can be checked that Steps 4 to 7 of Algorithm 2 perform twice the operation $S^*(x_1 \oplus x_2) \leftarrow S^*(x_1 \oplus x_2) \oplus N_j$. Thus, when M_j equals 0, Steps 2 to 9 apply the output mask N_j only to the half of the table values. Therefore the only solution to patch Algorithm 2 is to perform a particular re-computation when M_j equals 0. This would induce a dependency between the value of M_j and the execution time of the re-computation algorithm which, as remarked above, is a flaw with respect to 1-st order DPA.

B The Flaw vs. the S-Box Properties

In what follows, we show how the 3-rd order flaw presented in Section 3.2 interestingly depends on the S-Box properties. We firstly notice that the mutual information $\mathcal{I}(Y,(Y \oplus M, S(M \oplus e_1) \oplus N, S(M \oplus e_2) \oplus N))$ can be rewritten $\mathcal{I}(Y,(Y \oplus M, S(M) \oplus N, S(M \oplus e_1 \oplus e_2) \oplus N))$ when M is uniformly distributed and mutually independent with Y and N.

Proposition 1. *Let S be a (n,m)-function and let e be an element of $\mathbb{F}_2^n$. Let Y and M be two random variables defined over $\mathbb{F}_2^n$ and let N be a random variable defined over $\mathbb{F}_2^m$. If the three variables Y, M and N are mutually independent and have a uniform distribution, then the mutual information $\mathcal{I}((Y \oplus M, S(M) \oplus N, S(M \oplus e) \oplus N), Y)$ satisfies:*

$$\mathcal{I}(Y,(Y \oplus M, S(M) \oplus N, S(M \oplus e) \oplus N)) = n - \frac{1}{2^n} \sum_{z \in \mathbb{F}_2^m} \delta_S(e,z) \log\left(\delta_S(e,z)\right) \ ,$$

where, for every $z \in \mathbb{F}_2^m$, $\delta_S(e,z)$ denotes the cardinality of the set $\{x \in \mathbb{F}_2^n;\ D_eS(x) = z\}$

Proof. Let V denote the 3-uplet $(Y \oplus M, S(M) \oplus N, S(M \oplus e) \oplus N)$ and let us denote by $\mathcal{H}()$ the entropy of a random variable. The mutual information $\mathcal{I}(V,Y)$ equals $\mathcal{H}(V) - \mathcal{H}(V|Y)$. As V equals $(Y \oplus M, S(M) \oplus N, S(M \oplus e) \oplus N)$, it can be easily checked that the conditional entropy $\mathcal{H}(V|Y)$ equals $\mathcal{H}(M) + \mathcal{H}(N)$, which is equivalent to

$$\mathcal{H}(V|Y) = m + n \ . \tag{19}$$

From $\mathcal{H}(V) = -\sum_{v=(v_1,v_2,v_3)} \mathrm{P}(V = (v_1,v_2,v_3)) \log(P(V = (v_1,v_2,v_3))$, we deduce that the probability $P(V = (v_1,v_2,v_3))$ can be rewritten $\mathrm{P}(M = Y \oplus v_1, N = S(Y \oplus v_1) \oplus v_2, D_eS(Y \oplus v_1) = v_2 \oplus v_3)$, we have

$$\mathrm{P}(V = v|Y = y) = \mathrm{P}(M = y \oplus v_1, N = S(y \oplus v_1) \oplus v_2, D_eS(y \oplus v_1) = v_2 \oplus v_3)$$

As M and N are independent, the right-hand side of the relation above equals $\mathrm{P}(M = y \oplus v_1)\mathrm{P}(N = S(y \oplus v_1) \oplus v_2)$ if $v_1 \in \{x \in \mathbb{F}_2^n;\ D_eS(x \oplus$

$y) = v_2 \oplus v_3\}$ and equals 0 otherwise. After noticing that M and N are uniformly distributed over $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$ respectively, we get

$$\mathrm{P}(V = v \mid Y = y) = \begin{cases} \frac{1}{2^{n+m}} & \text{if } v_1 \in \{x \in \mathbb{F}_2^n;\ D_eS(x \oplus y) = v_2 \oplus v_3\} \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

From relation $P(V = v) = \sum_{y \in \mathbb{F}_2^n} P(Y = y)P(V = v \mid Y = y)$ and since Y has a uniform distribution over $\mathbb{F}_2^n$, (20) implies $\mathrm{P}(V = v) = \frac{\delta_S(e, v_2 \oplus v_3)}{2^{2n+m}}$. One deduces $\mathcal{H}(V) = -\frac{1}{2^{2n+m}} \sum_{v_1 \in \mathbb{F}_2^n} \sum_{v_2, v_3 \in \mathbb{F}_2^m} \delta_S(e, v_2 \oplus v_3) \log\left(\frac{\delta_S(e, v_2 \oplus v_3)}{2^{2n+m}}\right)$ that is

$$\mathcal{H}(V) = 2n + m - 2^{-n} \sum_{v_3 \in \mathbb{F}_2^m} \delta_S(e, v_3) \log(\delta_S(e, v_3)) \ , \tag{21}$$

since $\sum_{v_3 \in \mathbb{F}_2^m} \delta_S(e, v_3)$ equals 2^n.

As a consequence of (19) and (21), the mutual information $\mathcal{I}(V, Y)$ satisfies the Inequality of Proposition 1.

□

From Proposition 1, one deduces that the greater the summation $\sum_{z \in \mathbb{F}_2^m} \delta_S(e, z) \log(\delta_S(e, z))$, the smaller the amount of information $(Y \oplus M, S(M) \oplus N, S(M \oplus e) \oplus N)$ brings about Y. The summation is upper bounded by $n2^n$ and the bound is tight for $e = 0$ whatever the function S. Indeed, if e equals 0, then D_eS is the null function and $\delta_S(e, z)$ equals 2^n if $z = 0$ and equals 0 otherwise. However, the case $e = 0$ has no interest from an attacker viewpoint, since it is already clear that the mutual information between $(Y \oplus M, S(M) \oplus N)$ and Y is null. For every $e \in \mathbb{F}_2^{n*}$, summation $\sum_{z \in \mathbb{F}_2^m} \delta_S(e, z) \log(\delta_S(e, z))$ is smaller than or equal to $\sum_{z \in \mathbb{F}_2^m} \delta_S(e, z) \max_{(e,z) \in \mathbb{F}_2^{n*} \times \mathbb{F}_2^m} (\log(\delta_S(e, z)))$ and we get

$$\sum_{z \in \mathbb{F}_2^m} \delta_S(e, z) \log(\delta_S(e, z)) \leq 2^n \max_{(e,z) \in \mathbb{F}_2^{n*} \times \mathbb{F}_2^m} (\log(\delta_S(e, z)) \, ,$$

since $\sum_{z \in \mathbb{F}_2^m} \delta_S(e, z)$ equals 2^n. The value $\max_{(e,z) \in \mathbb{F}_2^{n*} \times \mathbb{F}_2^m} \delta_S(e, z)$ is usually denoted by δ and S is said to be δ-uniform. It plays a central role in the area of block ciphers since differentially δ-uniform SBoxes with smallest possible value of δ are those (n, m)-functions which contribute to a maximum resistance to *differential cryptanalysis* [16]. The number δ is lower bounded by 2^{n-m} and the bound is tight if and only if S is *perfect nonlinear.*

In the following proposition, we exhibit a relationship between the differential properties of S and the mutual information $\mathcal{I}((Y \oplus M, S(M) \oplus N, S(M \oplus e) \oplus N), Y)$.

Proposition 2. *Let S be a δ-uniform (n, m)-function. Let Y and M be two random variables defined over $\mathbb{F}_2^n$ and let N be a random variable defined over $\mathbb{F}_2^m$. If the three variables Y, M and N are mutually independent and have uniform distributions, then for every $e \neq 0$, we have*

$$\mathcal{I}((Y \oplus M, S(M) \oplus N, S(M \oplus e) \oplus N), Y) \geq n - \log(\delta) \ . \tag{22}$$

Moreover, if S is perfect nonlinear then $\mathcal{I}((Y \oplus M, S(M) \oplus N, S(M \oplus e) \oplus N), Y)$ equals m for every $e \in \mathbb{F}_2^{n*}$.

The proposition above shows that the quantity of information the uplet $(Y \oplus M, S(M) \oplus N, S(M \oplus e) \oplus N)$ provides on Y increases when the value δ decreases. This establishes that the resistance against differential attacks and the resistance against the attack described in Section 3.2 are two opposite notions.