Taylor Expansion of Maximum Likelihood Attacks for Masked and Shuffled Implementations

Nicolas Bruneau[1,2], Sylvain Guilley[1,3], Annelie Heuser[1]*, Olivier Rioul[1],
François-Xavier Standaert[4] and Yannick Teglia[5]**

[1] Institut Mines-Télécom, Télécom ParisTech, CNRS LTCI
Department Comelec, Paris, France
{firstname.lastname@telecom-paristech.fr}
[2] STMicroelectronics, AST division, Rousset, France
[3] Secure-IC S.A.S., Rennes, France
[4] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium
[5] Gemalto, France

Abstract. The maximum likelihood side-channel distinguisher of a template attack scenario is expanded into lower degree attacks according to the increasing powers of the signal-to-noise ratio (SNR). By exploiting this decomposition we show that it is possible to build highly multivariate attacks which remain efficient when the likelihood cannot be computed in practice due to its computational complexity. The shuffled table recomputation is used as an illustration to derive a new attack which outperforms the ones presented by Bruneau et al. at CHES 2015, and so across the full range of SNRs. This attack combines two attack degrees and is able to exploit high dimensional leakage which explains its efficiency.

Keywords: Template Attacks, Taylor expansion, Shuffled table recomputation.

1 Introduction

In order to protect embedded systems against side-channel attacks, countermeasures need to be implemented. Masking and shuffling are the most investigated solutions for this purpose [18]. Intuitively, masking aims at increasing the order of the statistical moments (in the leakage distributions) that reveal sensitive information [8, 15], while shuffling aims at increasing the noise in the adversary's measurements [14]. As a result, an important challenge is to develop sound tools to understand the security of these countermeasures and their combination [31]. For this purpose, the usual strategy is to consider template attacks for which one can split the evaluation goals into two parts: offline profiling (building an accurate leakage model) and online attack (recovering the key using the leakage model). As far as profiling is concerned, standard methods range from non-parametric ones

* Annelie Heuser is a Google European Fellow in the field of Privacy and is partially founded by this fellowship.

** Parts of this work have been done while the author was at STMicroelectronics.

(e.g., based on histograms or kernels) of which the cost quite highly suffers from the curse of dimensionality (see e.g., [2] for an application of these methods in the context of non-profiled attacks) to parametric methods, typically exploiting the mixture nature of shuffled and masked leakage distributions [16, 17, 25, 27, 33], which is significantly easier if the masks (and permutations) are known during the profiling phase. Our premise in this paper is that an adversary is able to obtain such a mixture model via one of these means, and therefore we question its efficient exploitation during the online attack phase.

In this context, a starting observation is that the time complexity of template attacks exploiting mixture models increases exponentially with the number of masks (when masking) and permutation length (when shuffling [37]). So typically, the time complexity of an optimal template attack exploiting Q traces against an implementation where each n-bit sensitive value is split into Ω shares and shuffled over Π different positions is in $\mathcal{O}\left(Q \cdot (2^n)^{\Omega-1} \cdot \Pi!\right)$, which rapidly turns out to be intractable. In order to mitigate the impact of this high complexity, we propose a small, well-controlled and principled relaxation of the optimal distinguisher, based on its Taylor expansion (already mentioned in the field of side-channel analysis in [6, 11]) of degree L. Such a simplification leads to various concrete advantages. First, when applied to masked implementations, it allows us to perform the (mixture) computations corresponding to the $(2^n)^{\Omega}$ factor in the complexity formula only once (thanks to precomputation) rather than Q times. Second, when applied to shuffled implementations, it allows us to replace the $\Pi!$ factor in this formula by $\binom{\Pi}{\min(\lceil\frac{\Pi}{2}\rceil, L)} = \binom{\Pi}{L}$, thanks to the bounded degree L.

Additionally it can be noticed that an attacker will only build, during the offline profiling, the leakage models needed for the attack. By applying the Taylor expansion of the optimal distinguisher the complexity of the offline profiling is significantly reduced. In general the complexity of the offline profiling becomes equivalent to the complexity of the online attack.

The resulting "rounded template attacks" additionally carry simple intuitions regarding the minimum degree of the Taylor expansion needed for the attacks to succeed. Namely, this degree L needs to be at least equal to the security order O of the target implementation, defined as the smallest statistical moment in the leakage distributions that are key-dependent.

We then show that these attacks only marginally increase the data complexity (for a given success rate) when applied against a masked (only) implementation. More importantly, we finally exhibit that rounded template attacks are especially interesting in the context of high-dimensional higher-order side-channel attacks, and put forward the significant improvement of the attacks against the masked implementations with shuffled table recomputations from CHES 2015 [7].

Introduction to Shuffled Table Recomputation. Masking the linear parts of a block cipher is straightforward whereas protecting the non-linear parts is less obvious. To solve this issue different methods have been proposed. One can cite algebraic methods [3, 30], using Global Look-Up Table (GLUT) [28] and table recomputation [1, 8, 10, 19]. Table recomputation methods are often used in practice as they represent a good tradeoff between memory consumption and execution time since they precompute a masked substitution box (S-Box) that is stored in a table.

However, some attacks still manage to recover the mask during the table recomputation [6,36]. As a further protection the recomputation can be shuffled. This protection uses a random permutation which is drawn over S_{2^n}, the set of all the permutation of $\mathbb{F}_2^n$. Therefore, some random masks are uniformly drawn over $\mathbb{F}_2^n$ to ensure the security against first-order attacks.

Contributions. We show that the expansion of the likelihood allows attacks with a very high computational *efficiency*, while remaining very *effective* from a key recovery standpoint. This means that the expanded distinguisher requires only little more traces to reach a given success rate, while being much faster to compute.

We also show how to grasp in a multivariate setting several leakages of different orders. In particular, we present an attack on shuffled table recomputation which succeeds with less traces than [7]. Notice that the likelihood attack cannot be evaluated in this setting because it is computationally impossible to average over both the mask and the shuffle (the sole number of shuffles is $2^n! \approx 2^{1684}$ with $n = 8$).

Finally, we show that are our rounded version of the maximum likelihood allows better attacks than the state-of-the-art. Namely, our attack is better than the classical 2O-CPA and the recent attack of CHES'15 [7] in all noise variance settings.

Outline. The remainder of the paper is organized as follows. Sec. 2 provides the necessary notations and mathematical definitions. The theoretical foundation of our method is presented in Sec. 3. The case-study (shuffled table recomputation) is shown in Sec. 4. Sec. 5 evaluates the complexity of our method. The performance results are presented in Sec. 6. Conclusions and perspectives are presented in Sec. 7. Some technical results are deferred to the appendices.

2 Notations

2.1 Parameters

Randomization countermeasures consist in *masking* and *shuffling* protections. When evaluating randomized implementations, there are a number of important parameters to consider. First, the number of shares and the shuffle length in the scheme, next denoted as Ω and Π, are algorithmic properties of the countermeasure. These numbers generally influence the tradeoff between the implementation overheads and the security of the countermeasures. Second, the order of the implementation protected by a randomization countermeasure, next denoted as O, which is a statistical property of the implementation. It corresponds to the smallest key-dependent statistical moment in the leakage distributions. When only masking is applied and the masked implementation is "perfect" (meaning that the leakage of each share is independent of each other), the order O equals to Ω at best. Finally, the number of dimensions (or dimensionality) used in the traces, next denoted as D, is a property of the adversary. In this respect, adversaries may sometimes be interested by using the lowest possible D (since it makes the detection of POIs in the traces easier). But from the measurement complexity point of view, they have a natural incentive to use D as large as possible. A larger dimension D allows to increase the signal to noise ratio [5].

In summary, our notations are:

- Ω: number of shares in the masking countermeasure,
- Π: length of the shuffling countermeasure,
- O: order of the implementation,
- D: dimensionality of the leakages.

Examples. Existing masking schemes combine these four values in a variety of manners. For example, in a perfect hardware masked implementation case with three shares, we may have $\Omega = 3$, $O = 3$ and $D = 1$ (since the three shares are manipulated in parallel). If this implementation is not perfect, we may observe lower order leakages (e.g. $\Omega = 3$, $O = 1$ and $D = 1$, that is a first-order leakage). And in order to prevent such imperfections, one may use a Threshold Implementation [24], in which case one share will be used to prevent glitches (so $\Omega = 3$, $O = 2$ and $D = 1$). If we move to the software case, we may then have more informative dimensions, e.g. $\Omega = 3$, $O = 3$, $D = 3$ if the adversary looks for a single triple of informative POIs. But we can also have a number of dimensions significantly higher than the order (which usually corresponds to stronger attacks). Let us also give an example of S-boxes masking with one mask, where the masking process of the S-box (often called recomputation) is shuffled. A permutation Φ of $\Pi = 2^n$ values is applied while computing the masked table. If the attacker ignores the recomputation step, he can carry out an attack on the already computed table. Hence parameters $\Omega = 2$, $O = 2$, $D = 2$ (also known as “second-order bivariate CPA”). But the attacker can also exploit the shuffled recomputation of the S-box in addition to a table look-up, as presented in [7]; the setting is thus highly multivariate: $\Omega = 2$, $\Pi = 2^n$, $O = 2$, $D = 2 \cdot 2^n + 1$. Interestingly, the paper [7] shows an attack at degree $L = 3$ which succeeds in less traces than attacks at minimal degree $L = O = 2$.

In general, a template attack based on mixture distributions (often used in parametric estimation) would require a summation over all random values of the countermeasure, that is $\mathcal{R}$, which consists in the set of masks and permutations. One can represent $\mathcal{R}$ as the Cartesian product of the set of mask and the set of permutations. Let us denote by $\mathcal{M}$ the set of mask and $\mathcal{S}$ the set of permutations. Then $\mathcal{R} = \mathcal{M} \times \mathcal{S}$. Therefore, the cardinality of $\mathcal{R}$ is $2^{n(\Omega-1)}\Pi!$.

Eventually, the security of a masked implementation depends on its order and noise level. More precisely, the security increases exponentially with the order (with the noise as basis) [12]. So for the designer, there is always an incentive to increase the noise and order. And for adversary, there is generally an incentive to use the largest possible D (given the time constraints of his attack), so that he decreases the noise.

2.2 Model

We characterize the protection level in terms of the most powerful attacker, namely an attacker who knows everything about the design, except the masks and the noise. This means that we consider the case where the templates are known. How the attacker got the templates is related with *security by obscurity*, somehow he will know the model. Of course depending on the learning phase these estimations can be more or less accurate. For the sake of simplicity we assume in this paper the better scenario where all the estimations are exact[6].

Besides, we assume that the noise is independently distributed over each dimension. This is the least favorable situation for the attacker (as there is in this case the most noise entropy). For the sake of simplicity, we assume that the noise variance is equal to σ^2 at each point $d = 1, 2, \dots, D$. This allows for a simple theoretical analysis. Let us give an index $q = 1, 2, \dots, Q$ to each trace. For one trace q, the model is written as:

$$X = y(t, k^*, R) + N, \tag{1}$$

where for notational convenience the dependency in q and d has been dropped. Here X is a leakage measurement; $y = y(t, k^*, R)$ is the deterministic part of the model that depends on the correct key k^*, some known text (plaintext or ciphertext) t, and the unknown random values (masks and permutations) R. Each sample (of index d) of N is a random noise, which follows a Gaussian distribution $p_N(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{z^2}{2\sigma^2}\right)$.

Uppercase letters are generally used for random variables and the corresponding lowercase letters for their realizations. Bold symbols are used to denote vectors that have length Q, the number of measurements. Namely, $\mathbf{X}$ denotes a set of Q random variables i.i.d. with the same law as X. So, $\mathbf{X}$ is a $Q \times D$ matrix; $\mathbf{R}$ denotes a set of random variables i.i.d. with the same law as R; $\mathbf{t}$ denotes the set of input texts of the measurements $\mathbf{X}$; $y(\mathbf{t}, k, \mathbf{R})$ denotes the set of leakage models, where k is a key guess, k^* being the correct key value.

Notations $\mathbf{X}_d$ and $\mathbf{X}^{(q)}$ are used to denote the d-th column and the q-th line of the matrix $\mathbf{X}$, respectively.

We are interested in attacks where each intermediate data is a n-bit vector. In particular, we target S-boxes, denoted by S. Regarding the transduction from the intermediate variable to the real-valued leakage, we take the example of the Hamming weight w_H defined by $w_H(z) = \sum_{i=1}^{n} z_i$ where z_i is the ith bit of z.

3 A Generic Log-Likelihood for Masked Implementations

In this section we derive a rounded version of Template Attack. Namely we expand a particular instantiation of the template attack the so-called optimal distinguisher using its Taylor Expansion. By rounding this expansion at the Lth

[6] We recall that, even if the templates are perfectly known, the online attack phase still requires $\mathcal{O}(Q \cdot 2^{n(\Omega-1)} \cdot \Pi!)$ computations.

degree we are able to build a rounded version of the optimal distinguisher (later defined as ROPT_L). This attack features two advantages: it allows to combine different statistical moments and its complexity becomes manageable.

3.1 Maximum Likelihood (ML) Attack

The most powerful adversary knows exactly the leakage model (but the actual key, the masks, and the noise are unknown during the online step) and computes a likelihood. In the case of masking the optimal distinguisher which maximize the success rate is given by [6]:

Theorem 1 (Maximum Likelihood). *When the $y\left(t,k,R\right)$ are known and the Gaussian noise N is i.i.d. across the queries (measurements) and independent across the dimension, then the optimal distinguisher is:*

$$\begin{aligned} \mathrm{OPT}\colon\ \mathbb{R}^{DQ}\times\mathbb{R}^{DQ} &\longrightarrow \mathbb{F}_2^n \\ \left(\mathbf{x}, y\left(\mathbf{t},k,R\right)\right) &\longmapsto \operatorname*{argmax}_{k\in\mathbb{F}_2^n} \sum_{q=1}^{Q} \log \mathbb{E} \exp \frac{-\|x^{(q)}-y(t^{(q)},k,R)\|^2}{2\sigma^2} \end{aligned} \tag{2}$$

where the expectation operator $\mathbb{E}$ is applied with respect to the random variable $R\in\mathcal{R}$, and the norm is the Euclidean norm $\|x^{(q)}-y(t^{(q)},k,R)\|^2=\sum_{d=1}^{D}(x_d^{(q)}-y_d(t^{(q)},k,R))^2$.

Proof. It is proven in [6] that the Maximum Likelihood distinguisher is:

$$\operatorname*{argmax}_{k\in\mathbb{F}_2^n} \prod_{q=1}^{Q} \sum_{r\in\mathcal{R}} \mathbb{P}\left(r\right) p\left(x^{(q)}|y\left(t^{(q)},k,r\right)\right).$$

Applying (1) for Gaussian noise and taking the logarithm yields (2). □

In the sequel, we denote by $LL^{(q)}=\log\mathbb{E}_R\exp\frac{-\|x^{(q)}-y(t^{(q)},k,R)\|^2}{2\sigma^2}$ the contribution of one trace q of the Log-Likelihood full distinguisher $LL=\sum_{q=1}^{Q}LL^{(q)}$.

Remark 1. Notice that for each trace q, the Maximum Likelihood distinguisher involves a summation over $\#\mathcal{R}$ values, which correspond to $\#\mathcal{R}$ accesses to precharacterized templates.

If $D=1$, then the signal-to-noise ratio (SNR) is defined in a natural way as the ratio between the variance of the model Y and the variance of the noise N. But when the setup is multivariate, it is more difficult to quantify a notion of SNR. For this reason, we use the following quantity

$$\gamma=\frac{1}{2\sigma^2}, \tag{3}$$

which is actually proportional to an SNR, in lieu of SNR. In practice, we assume that γ is small. It is indeed a condition for masking schemes to be efficient (see for instance [12]).

Proposition 1 (Taylor Expansion of Optimal Attacks in Gaussian Noise). *The attack consists in maximizing the sum over all traces $q = 1, \ldots, Q$ of*

$$\sum_{\ell=1}^{+\infty} \frac{\kappa_\ell}{\ell!}(-\gamma)^\ell, \tag{4}$$

where κ_ℓ is the ℓth-order cumulant of the random variable $\|x - y(t, k, R)\|^2$, which can be found inductively from ℓth-order moments:

$$\mu_\ell = \mathbb{E}_R\big(\|x - y(t, k, R)\|^{2\ell}\big), \tag{5}$$

using the relation:

$$\kappa_\ell = \mu_\ell - \sum_{\ell'=1}^{\ell-1} \binom{\ell-1}{\ell'-1} \kappa_{\ell'} \mu_{\ell-\ell'} \qquad (\ell \geq 1). \tag{6}$$

Proof. The log-likelihood can be expanded according to the increasing powers of the SNR as:

$$\log \mathbb{E} \exp\big(-\gamma \|x - y(t, k, R)\|^2\big) = \sum_{\ell=1}^{+\infty} \frac{\kappa_\ell}{\ell!}(-\gamma)^\ell, \tag{7}$$

where we have recognized the cumulant generating function [34]. The above relation (6) between cumulants and moments is well known [39]. □

Definition 1. *The Taylor expansion of the log-likelihood truncated to the Lth degree LL_L in SNR is*

$$\mathrm{LL}_L = \sum_{\ell=1}^{L} (-1)^\ell \kappa_\ell \frac{\gamma^\ell}{\ell!}. \tag{8}$$

Put differently, we have $\mathrm{LL} = \mathrm{LL}_L + o(\gamma^L)$ (using the Landau notation). The optimal attack can now be “rounded” in the following way:

Definition 2 (Rounded OPTimal Attack of Degree L in γ). *The rounded optimal L*th-degree attack *consists in maximizing over the key hypothesis the sum over all traces of the Lth order Taylor expansion LL_L in the SNR of the log-likelihood :*

$$\begin{aligned} \mathrm{ROPT}_L \colon\ \mathbb{R}^{DQ} \times \mathbb{R}^{DQ} &\longrightarrow \mathbb{F}_2^n \\ (\mathbf{x}, y(\mathbf{t}, k, R)) &\longmapsto \underset{k \in \mathbb{F}_2^n}{\operatorname{argmax}}\, \mathrm{LL}_L. \end{aligned} \tag{9}$$

Proposition 2. *If the degree L is smaller than the order O of the countermeasure then the attack fails to distinguish the correct key.*

Proof. One can notice that μ_ℓ combines (by a product) a most ℓ terms following the formula:

$$\mu_\ell = \sum_{k_1+\ldots+k_D=\ell} \binom{\ell}{k_1,\ldots,k_D} \mathbb{E} \prod_{0<i<D+1} (x_i - y_i)^{2\cdot k_i},$$

with $k_1 + \ldots + k_d = \ell$. It implies that it exits at most ℓ different $k_i > 0$ and as a consequence there are at most ℓ different variables in the expectation. Therefore by definition of a perfect masking scheme μ_L does not depend on the key. As a consequence LL_L with $L < O$ neither depends on the key.

□

Theorem 2. *Let an implementation be secure at order O. The lowest-degree successful attack is the one at degree $L = O$ which maximizes LL_L. This is equivalent to summing*

$$\mu_L = \mathbb{E}_R\big(\|x - y(t,k,R)\|^{2L}\big),$$

over all traces and

- *maximize the result over the key hypotheses, if L is even;*
- *minimize the result over the key hypotheses, if L is odd.*

Proof. Since κ_ℓ is independent of k for all $\ell \leq L$, the first sensitive contribution to the log-likelihood is

$$(-1)^L \kappa_L \frac{\gamma^L}{L!}.$$

Now, $\kappa_L = \mu_L +$ lower order terms (which do not depend on the key as the implementation is secure at order O), and removing constants independent of k the contribution to the log-likelihood reduces to $(-1)^L \mu_L$. □

Theorem 3 (Mixed Degree Attack). *Assuming an implementation secure at order O, the next degree successful attack is the one at degree $L + 1 = O + 1$ which maximizes LL_{L+1}. This is equivalent to summing*

$$\mu_L(1 + \gamma\mu_1) - \gamma\frac{\mu_{L+1}}{L+1},$$

over all traces and

- *maximize the result over the key hypotheses, if L is even;*
- *minimize the result over the key hypotheses, if L is odd.*

Proof. The $(L + 1)$th-order term in the log-likelihood becomes

$$(-1)^L \kappa_L \frac{\gamma^L}{L!} + (-1)^{L+1} \frac{\kappa_{L+1}}{(L+1)!} \gamma^{L+1}.$$

Now from (6) we have, for $L > 0$

$$\kappa_{L+1} = \mu_{L+1} - (L+1)\mu_L\mu_1 + \text{ lower-order terms.}$$

Removing terms that do not depend on k, we obtain:

$$(-1)^L\gamma^L\Big(\mu_L - \gamma(\frac{\mu_{L+1}}{L+1} - \mu_L\mu_1)\Big).$$

Compared to a Lth-degree attack, we see that μ_L is replaced by a corrected version:

$$\mu_L(1+\gamma\mu_1) - \gamma\frac{\mu_{L+1}}{L+1},$$

where μ_1 is independent of k. However, μ_1 cannot be removed as it scales the relative contribution of μ_L and μ_{L+1} in the distinguisher. □

Remark 2. In contrast to LL_L, implementing LL_{L+1} requires knowledge of the SNR parameter $\gamma = 1/2\sigma^2$.

Remark 3. In general, when $L \geq O$ the rounded optimal attack ROPT_L exploits all key dependent terms of degree ℓ, where $O \leq \ell \leq L$, whereas an LO-CPA [8] or MCP-DPA [22] only exploit the term of degree L.

4 Case Study: Shuffled Table Recomputation

In this section we apply the ROPT_L formula of Eq. (9) of Def. 2 to the particular case of a block cipher with a shuffled table recomputation stage. We show that in this scenario our new method allows to build a better attack than that from the state-of-the-art. By combining the second and the third cumulants we construct an attack which is better than:

- any second-order attack;
- the attack presented at CHES 2015. Following the notations of [7] we denote this attack by MVA_{TR} (which stands for Multi-Variate Attack on Table Recomputation) in the rest of this article. This is a third-order attack that achieves better results than 2O-CPA when the noise level σ is below a given threshold (namely $\sigma^2 \leq 2^{n-2} - n/2$).

4.1 Parameters of the Randomization Countermeasure

In order to validate our results we take as example a first order ($O = 2$), masking scheme where the sensitive variables are split into two shares ($\Omega = 2$). The nonlinear part of this scheme is computed using a table recomputation stage. This step is shuffled ($\Pi = 2^n$) for protection against some known attacks [26,36].

The beginning of this combined countermeasure is given in Algorithm 1. The table is recomputed in a random order from line 3 to line 7.

Algorithm 1: Beginning of computation of a block cipher masked by table recomputation in a random order

```
   input  : $t$, one byte of plaintext, and $k$, one byte of key
   output : The application of AddRoundKey and SubBytes on $t$, i.e., $S[t \oplus k]$

   // Table precomputation protected by shuffling ...............
1  $m \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$, $m' \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$     // Draw of random input and output masks
2  $\varphi \leftarrow_{\mathcal{R}} \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$              // Draw of random permutation of $\mathbb{F}_2^n$
3  for $\varphi(\omega) \in \{\varphi(0), \varphi(1), \dots, \varphi(2^n - 1)\}$ do        // S-box masking
4      $z \leftarrow \varphi(\omega) \oplus m$                                  // Masked input
5      $z' \leftarrow S[\varphi(\omega)] \oplus m'$                             // Masked output
6      $S'[z] = z'$                           // Creating the masked S-box entry
7  end
   // Masked computation ..........................................
8  $t \leftarrow t \oplus m$                                       // Plaintext masking
9  $t \leftarrow t \oplus k$                                      // Masked AddRoundKey
10 $t \leftarrow S'[t]$                                           // Masked SubBytes
11 $t \leftarrow t \oplus m'$                                            // Demasking
12 return $t$
```

We used lower case letter (e.g., m, φ) for the realizations of random variables, written upper-case (e.g., M, Φ). For the sake of simplicity in the rest of this case study, we assume that $m = m'$.

An overview of the leakages over time is given in Fig. 1.

We detail below the mathematical expression of these leakages. The randomization consists in one mask M chosen randomly in $\{0,1\}^n$, and one shuffle (random permutation of $\{0,1\}^n$) denoted by Φ. Thus, we denote $R = (M, \Phi)$, which is uniformly distributed over the Cartesian product $\{0,1\}^n \times S_{2^n}$ (i.e. $\mathcal{M} = \{0,1\}^n$ and $\mathcal{S} = S_{2^n}$), where S_m is the symmetric group of m elements. We have $D = 2^{n+1} + 2$ leakage models, namely:

- $X_0 = y_0(t,k,R) + N_0$ with $y_0(t,k,R) = w_H(M)$,
- $X_1 = y_1(t,k,R) + N_1$ with $y_1(t,k,R) = w_H(S[T \oplus k] \oplus M)$,
- $X_i = y_i(t,k,R) + N_i$, for $i = 2, \dots, 2^n + 1$ with $y_i(t,k,R) = w_H(\Phi(i-2) \oplus M)$,
- $X_j = y_j(t,k,R) + N_j$, for $j = 2^n + 2, \dots, 2^{n+1} + 1$ with $y_j(t,k,R) = w_H(\Phi(j - 2^n - 2))$.

We recall that we assume the noises N are i.i.d. Clearly, there is a second-order leakage, as the pair (X_0, X_1) does depend on the key. But there is also a large multiplicity of third-order leakages, such that $(X_1, X_i, X_{j=i+2^n})$, as will be analyzed in this case-study.

The following side-channel attacks are applied on a set of Q realizations. Let us define I and J as $I = [\![2, 2^n + 1]\!]$ and $J = [\![2^n + 2, 2 \times 2^n + 1]\!]$. Then the maximal dimensionality is $D = 2 + 2 \times 2^n$, and we denote a sample d as

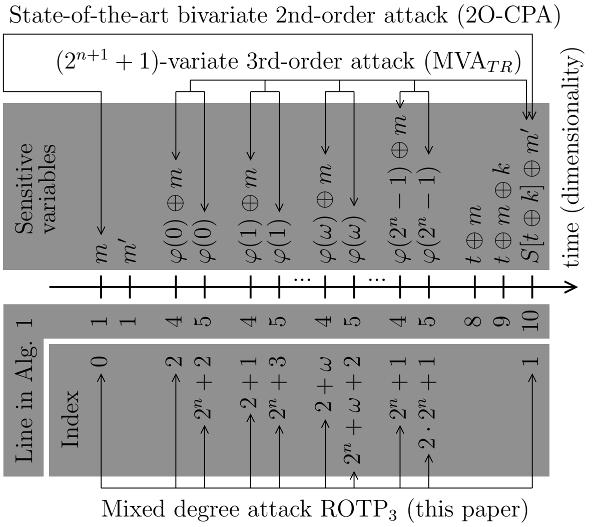


Fig. 1: Leakages of the shuffled table recomputation scheme

$d \in \{0,1\} \cup I \cup J$. The Q leaks (resp. models) at sample d are denoted as $\mathbf{x}_d$ and $\mathbf{y}_d = y_d(\mathbf{t}, k, R)$.

In order to simplify the notations we introduce

$$f_d^{(q)} = \left(x_d^{(q)} - y_d\left(t^{(q)}, k, R\right)\right)^2, \tag{10}$$

with $d \in \{0,1\} \cup I \cup J$. The $^{(q)}$ can be omitted where there is no ambiguity.

4.2 Second-Order Attacks

As any other high order masking scheme, our example can be defeated by High Order Attacks [8,20,29,38]. As our scheme is a first order masking scheme with two shares it can be defeated using a second order attack [8,20] which combines the leakages of the two shares using a *combination function* [8,20,25] such as the second order CPA (2O-CPA) with the centered product as combination function.

Using our notation it implies $D = 2$.

Definition 3 **(**2O-CPA **[29]).** *We denote by* 2O-CPA *the* CPA *using the centered product as combination function. Namely:*

$$\begin{aligned} \text{2O-CPA}\colon \mathbb{R}^Q \times \mathbb{R}^Q \times \mathbb{R}^Q &\longrightarrow \mathbb{F}_2^n \\ (\mathbf{x_0}, \mathbf{x_1}, \mathbf{y}) &\longmapsto \operatorname*{argmax}_{k \in \mathbb{F}_2^n} \widehat{\rho}\left[\mathbf{x_0} \circ \mathbf{x_1}, \mathbf{y}\right], \end{aligned} \tag{11}$$

where $\mathbf{y} = \mathbb{E}_M\left(y_0\left(\mathbf{t}, k, R\right) \circ y_1\left(\mathbf{t}, k, R\right)\right)$*,* $\circ$ *is the element wise product and* $\widehat{\rho}$ *is an estimator of the Pearson coefficient. It can be noticed that as the terms*

$y_0(\mathbf{t}, k, R)$ and $y_1(\mathbf{t}, k, R)$ only depend on M the expectation is only computed over $\mathcal{M}$.

Remark 4. Here we have assumed without loss of generality that the leakages and the model are centered.

An attacker can restrict himself in order to ignore the recomputation stage. Since such attacker ignores the table recomputation no random shuffle is involved. As a consequence the optimal distinguisher restricted to these leakages becomes computable. Nevertheless as we will see in Sec. 6 this approach is not the best. Indeed a lot of exploitable information is lost by not taking into account the table recomputation.

Definition 4 ($\mathrm{OPT}_{2\mathrm{O}}$ Distinguisher — Eq. (2) for $D = 2$). *We define by $\mathrm{OPT}_{2\mathrm{O}}$ the optimal attack which targets the mask and the masked sensitive value.*

$$\begin{aligned}\mathrm{OPT}_{2\mathrm{O}}\colon \quad & \mathbb{R}^{2Q} \times \mathbb{R}^{2Q} \to \mathbb{F}_2^n \\ & (\mathbf{x}_d, y_d(\mathbf{t}, k, R))_{d\in\{0,1\}} \mapsto \underset{k\in\mathbb{F}_2^n}{\mathrm{argmax}} \sum_{q=1}^{Q} \log \mathbb{E} \exp\left(-\gamma \sum_{d\in\{0,1\}} f_d^{(q)}\right),\end{aligned} \tag{12}$$

with $f_d^{(q)}$ as defined in Eq. (10).

4.3 Exploiting the Shuffled Table Recomputation Stage

It is known that the table recomputation step can be exploited to build better attacks than second order attacks [6,36]. Recently a new attack has been presented which remains better than the 2O-CPA even when the recomputation step is protected [7]. Let us recall the definition of this attack:

Definition 5 (MVA_{TR} [7]). *The MultiVariate Attack (MVA) exploiting the leakage of the table recomputation (TR) is given by the function:*

$$\begin{aligned}\mathrm{MVA}_{TR}\colon & \mathbb{R}^{Q(2^{n+1}+1)} \times \mathbb{R}^{Q} \longrightarrow \mathbb{F}_2^n \\ & (\mathbf{x}_d, \mathbf{y})_{d\in\{1\}\cup I\cup J} \longmapsto \underset{k\in\mathbb{F}_2^n}{\mathrm{argmax}}\, \widehat{\rho}\left[\left(-\frac{1}{2}\sum_{i\in I, j=i+2^n} \mathbf{x_i} \circ \mathbf{x_j}\right) \circ \mathbf{x_1}, \mathbf{y}\right],\end{aligned} \tag{13}$$

where, like for Def. 3, $\mathbf{y} = \mathbb{E}_M(y_0(\mathbf{t}, k, R) \circ y_1(\mathbf{t}, k, R))$, $\circ$ is the element wise product and $\widehat{\rho}$ is an estimator of the Pearson coefficient.

Let us now apply our new ROPT_L on a block cipher protected with a shuffled table recomputation. In this case the lower moments are given by:

$$\mu_\ell = \mathbb{E}\left[\left(\sum_d f_d\right)^\ell\right] = \mathbb{E}\left[\Big(\underbrace{f_0}_{S[t\oplus k]\oplus M} + \underbrace{f_1}_{M} + \sum_{i\in I}\underbrace{f_i}_{\Phi(\omega)\oplus M} + \sum_{j\in J}\underbrace{f_j}_{\Phi(\omega)}\Big)^\ell\right].$$

Proposition 3. *The second degree rounded optimal attack on the table recomputation is:*

$$\begin{aligned} \mathrm{ROPT}_2\colon \quad & \mathbb{R}^{2Q} \times \mathbb{R}^{2Q} && \longrightarrow \mathbb{F}_2^n \\ & (\mathbf{x}_d, y_d\,(\mathbf{t}, k, R))_{d\in\{0,1\}} && \longmapsto \operatorname*{argmax}_{k\in\mathbb{F}_2^n} \sum_{q=1}^{Q} \mathbb{E}(f_0^{(q)} \times f_1^{(q)}). \end{aligned} \tag{14}$$

Proof. Combine Theorem 2 and Eq. (30) of Appendix A.2. □

Remark 5. The ROPT$_2$ which targets the second order moment happens not to take into account the terms of the recomputation stage. Naturally the only second order leakages are also the ones used by 2O-CPA and OPT$_{2\mathrm{O}}$ distinguishers.

Proposition 4. *The third degree rounded optimal attack on the table recomputation is:*

$$\begin{aligned} \mathrm{ROPT}_3\colon \quad & \mathbb{R}^{(2^{n+1}+2)Q} \times \mathbb{R}^{(2^{n+1}+2)Q} && \longrightarrow \mathbb{F}_2^n \\ & (\mathbf{x}_d, y_d\,(\mathbf{t}, k, R))_{d\in\{0,1\}\cup I\cup J} && \longmapsto \operatorname*{argmax}_{k\in\mathbb{F}_2^n} \sum_{q=1}^{Q} \mu_2^{(q)}(1+\gamma\mu_1^{(q)}) - \gamma\frac{\mu_3^{(q)}}{3}, \end{aligned} \tag{15}$$

where the values of $\mu_1^{(q)}$, $\mu_2^{(q)}$ *and,* $\mu_3^{(q)}$ *are respectively provided in Eq.* (22) *of Appendix A.1, Eq.* (30) *of Appendix A.2 and Eq.* (33) *of Appendix A.3.*

Proof. Combining Theorem 2 and Appendix A. □

Proposition 5. *To compute* μ_1, μ_2 *and* μ_3 *an attacker does not need to compute the expectation over* S_{2^n}.

Proof. Proof given in Appendix A. □

5 Complexity

In this section we give the *time* complexity needed to *compute* OPT and ROPT$_L$. We also show that when $L \ll D$ the complexity of ROPT$_L$ remains manageable whereas the complexity of OPT is prohibitive. In this section all the complexities are computed for one key guess.

5.1 Complexity in the General Case

Let us first introduce an intermediate lemma.

Lemma 1. *The complexity of computing* μ_ℓ *(for one trace) is lower than:*

$$\mathcal{O}\left(\binom{D+\ell-1}{\ell} \cdot 2^{(\Omega-1)n} \cdot \binom{\Pi}{\min\left(\left\lceil \frac{\Pi}{2} \right\rceil, \ell\right)}\right). \tag{16}$$

Proof. See Appendix B.1. □

Proposition 6. *The complexity of* OPT *is:*

$$\mathcal{O}\left(Q \cdot (2^n)^{\Omega-1} \cdot \Pi! \cdot D\right). \tag{17}$$

The complexity of ROPT_L *is lower than:*

$$\mathcal{O}\left(Q \cdot L \cdot \binom{D+L-1}{L} \cdot 2^{(\Omega-1)n} \cdot \binom{\Pi}{\min\left(\left\lceil \frac{\Pi}{2} \right\rceil, L\right)}\right). \tag{18}$$

Proof. The proof is given in Appendix B.2. □

Prop. 6 allows to compare the complexity of the two attacks. One can notice that there are still terms with $\Pi!$ or $D!$ in ROPT_L such as $\binom{D+L-1}{L}$ or $\binom{\Pi}{\min(\lceil \frac{\Pi}{2} \rceil, L)}$. Nevertheless these two terms can be seen as constants where $L \ll D$. As a consequence we have the following remark.

Important Remark. When the degree L of the attack ROPT_L is such that $L \ll D$ the complexity of OPT is much higher than the complexity of ROPT_L. Indeed the main term for OPT is $\Pi!$ whereas the one for ROPT_L is $2^{(\Omega-1)n}$.

Proposition 7. *The complexity of* ROPT_L *can be reduced to* $\mathcal{O}\left(Q \cdot L \cdot \binom{D+L-1}{L}\right)$ *with a precomputation in* $\mathcal{O}\left(L \cdot \binom{D+L-1}{L} \cdot 2^{(\Omega-1)n} \cdot \binom{\Pi}{\min(\lceil \frac{\Pi}{2} \rceil, L)}\right)$.

Proof. See Appendix B.3. □

This means that for Q large enough i.e. when γ is low enough this computational "trick" allows a speed-up factor of $2^{(\Omega-1)n}\binom{\Pi}{\min(\lceil \frac{\Pi}{2} \rceil, L)}$. The idea is to output the values depending on the queries from the computation of the expectations. These expectations only depend on the model which can be computed only once.

5.2 Complexity of our Case Study

Let us now compute the complexity of these two distinguishers applied to our case study. Of course an approach could be to use the formula of the previous section 5.1. But one can notice that a lot of terms could be independent of the key and as consequence not needed in an attack. Another approach is to use the formula of the distinguisher.

Proposition 8. *The complexity of* OPT *is:*

$$\mathcal{O}\left(Q \cdot (2^n) \cdot 2^n! \cdot \left(2^{n+1} + 2\right)\right). \tag{19}$$

The complexity of ROPT_2 *is:*

$$\mathcal{O}\left(Q \cdot 2^n\right). \tag{20}$$

The complexity of ROPT_3 *is lower than:*

$$\mathcal{O}\left(Q \cdot 2^{4n}\right). \tag{21}$$

Proof. See Appendix B.4. □

Remark 6. As already mentioned an attacker can ignore the leakages of the table recomputation and only target the two shares. In such case the complexity of $\mathrm{OPT}_{2\mathrm{O}}$ (Def. 4) is $\mathcal{O}\left(Q \cdot (2^n)\right)$. With the result of Prop. 7 the complexity of ROPT_2 reduces to $\mathcal{O}\left(Q\right)$.

Remark 7. Using the result of Prop. 7 the complexity of ROPT_3 can be reduced to $\mathcal{O}\left(Q \cdot 2^{2n}\right)$ with a precomputation step of $\mathcal{O}\left(2^{2n}\right)$.

Remark 8. A summary of the complexity, and the computation time of the distinguishers are provided in Appendix B.5 in Table 1.

6 Simulation Results

In this section we validate in simulation the soundness of our approach for the case study described in Sec. 4.1. The results of these simulations are expressed in success rate (defined in [32] and denoted by SR). All simulations are computed using the Hamming weight model as a leakage model. As we assume an attacker with a perfect knowledge, the leakages are the model (denoted by y) plus some noise. The noise is Gaussian with a standard deviation of σ.

In Subsec 6.1 we assume that the attacker does not take into account the table recomputation stage. He only targets the leakages of the mask and the masked share (the leakage of masked S-Box). Namely the leakages which occurs in lines 1 and 10 of Algorithm 1. This approach allows to compute the restricted version of the maximum likelihood. We compare the results of the maximum likelihood, our rounded version and the high order attacks.

In Subsec 6.2 we present our main results. In this subsection the attacker can exploit the leakage of the mask, the masked share and all the leakages of the table recomputation. In this scenario we show that our rounded version of the optimal distinguisher outperforms all the attacks of the state-of-the-art.

6.1 Exploiting only Leakage of the Mask and the Masked Share

In this subsection all the attacks are computed using only the leakages of the line 1 and the line 10 of Algorithm 1.

In this case study we assume a perfect masking scheme with: $Y_0 = w_H(M)$ and $Y_1 = w_H(S[T \oplus k] \oplus M)$.

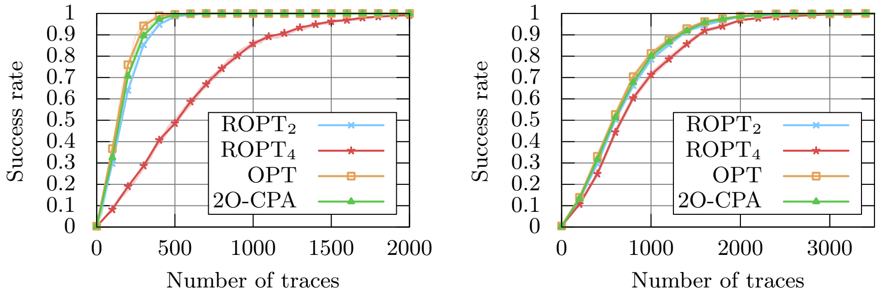
It can be seen in Fig. 2 that even for small noise ($\sigma = 1$, Fig. 2a) the 2O-CPA and ROPT_2 are equivalent. Indeed the two curves superimpose almost perfectly (in order to better highlight a difference, as many as 1000 attacks have been carried out for the estimation of the success rate). Moreover these two attacks are nearly equivalent to the optimal distinguisher (we recover here the results of [6]). We can notice that for both $\sigma = 1$ and $\sigma = 2$, ROPT_4 is not as good as ROPT_2. This means that the noise standard deviation is not large enough for approximations of higher degrees to be accurate. Indeed when the noise is not low enough the weight of each term of the decomposition can be such that some useful terms vanish due to the alternation of positive and negative terms in the Taylor expansion.

Let us recall that the decomposition of Eq. (8) is valid only for low $\gamma = 1/(2\sigma^2)$ i.e. high noise. The error term ($o(\gamma^L)$) in the Taylor expansion gives the asymptotic evolution of this error when the noise increases but does not provide information about the error for a fixed value of noise variance. This means that the noise is too small for ROPT_4 to be a good approximation of OPT although ROPT_2 is nearly equivalent to OPT.

For $\sigma = 2$ the noise is high enough to have a good approximation of OPT by ROPT_4. For this noise all the attacks are close to OPT (Fig. 2b).

In the context where only the mask and the masked share are used it is equivalent to compute the 2O-CPA, ROPT_2 and OPT. As a consequence in the rest of this article only the 2O-CPA will be displayed.

To conclude our ROPT_L is in this scenario at least as good as the HO-CPA of order L, which validates the optimality of state-of-the-art attacks against perfect masking schemes of order $O = L$.



(a) $\sigma = 1$

(b) $\sigma = 2$

Fig. 2: Bivariate attacks

6.2 Exploiting the Shuffled Table Recomputation

In this subsection the attacker can target the leakage of the mask, the masked share and all the leakages occurring during the table recomputation. As a consequence the attacks of Subsec 6.1 remain possible. It has been shown in [6, 33] that the 2O-CPA with the centered product becomes close to the $\text{OPT}_{2\text{O}}$ (the Maximum Likelihood) when the noise becomes high. It is moreover confirmed by our simulation results as it can be seen in Fig. 2. We choose as attack reference for the Fig. 3 the 2O-CPA and not the $\text{OPT}_{2\text{O}}$ because it performs similarly Fig. 2 and it is much faster to compute (see Table 1) which is mandatory for attacks with high noise (e.g. for $\sigma = 12$) which involve many traces.

Following the formulas provided previously empirical validations have been done. For $\sigma \leq 8$ the attacks have been redone 1000 times to compute the SR. For $\sigma > 8$ the attacks have been done 250 times. Results are plotted in Fig. 3. In these figures the results of the 2O-CPA, the MVA$_{TR}$ and ROPT$_3$ are plotted. Noticed that the likelihood is not represented because we cannot average over R.

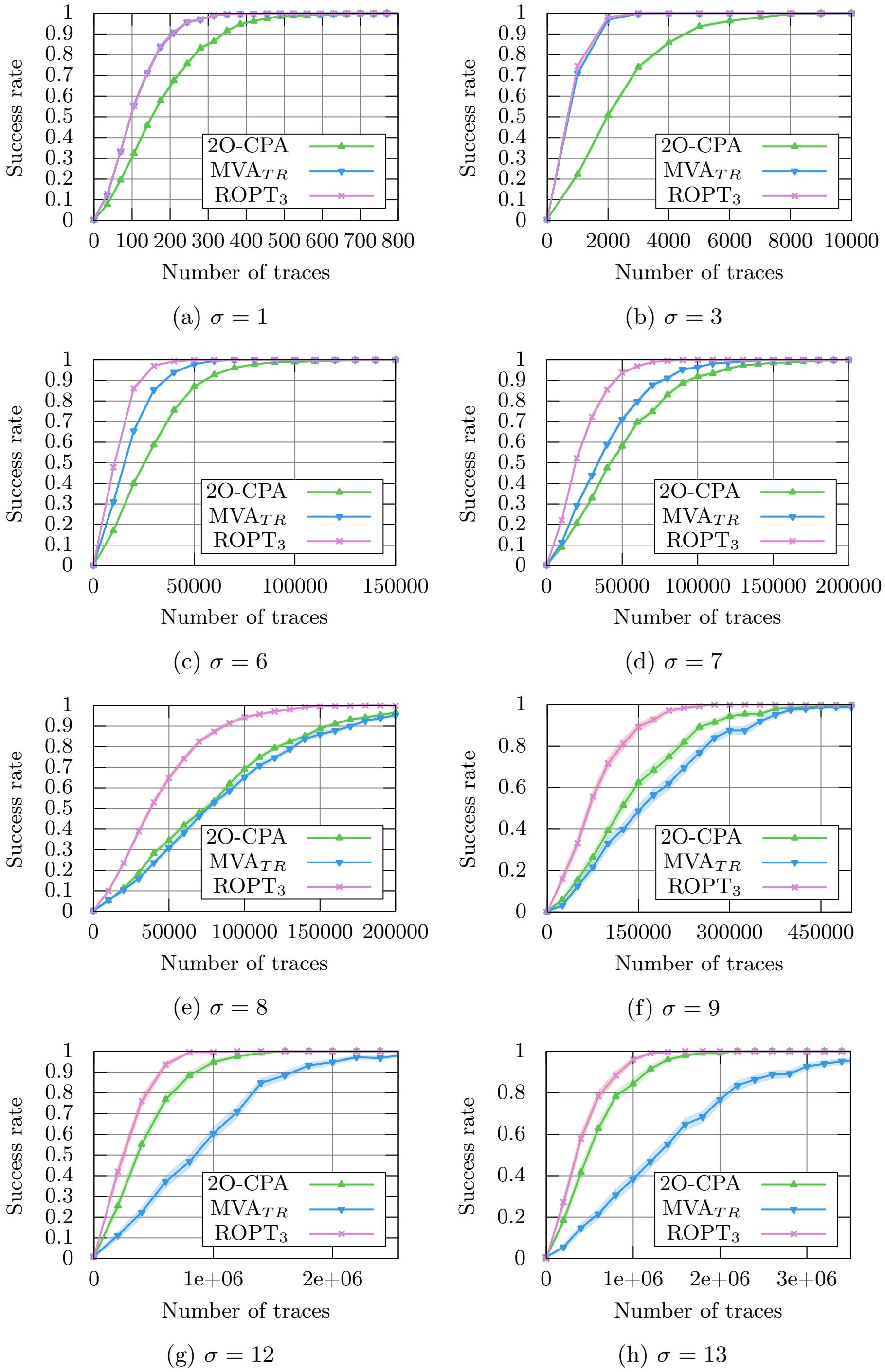
Recall that the cardinality of the support of R is $2^n \times 2^n!$. It can be first noticed that for all the noises ROPT$_3$ is the best attack.

Let us analyze how much better ROPT$_3$ is than 2O-CPA and MVA$_{TR}$. The comparison with our new attack can be divided in three different categories. For low noise $\sigma = 3$ (see Fig. 3b) the results of ROPT$_3$ are similar to the results of MVA$_{TR}$. This means that the leakage of the shuffled table recomputation is the most leaking term in this case. At the opposite when the noise is high (for $\sigma = 12$ see Fig. 3g) ROPT$_3$ becomes close to 2O-CPA which means that as expected the most informative part is the second order term. For medium noise $7 \leq \sigma \leq 9$ (see Fig. 3d, Fig. 3e and Fig. 3f) the results of ROPT$_3$ are much better than the result of 2O-CPA and MVA$_{TR}$. Moreover, the gain compared to the second best attack is maximum when the results of 2O-CPA and MVA$_{TR}$ are the same. Indeed for $\sigma = 7$ (see Fig. 3d), ROPT$_3$ needs 35000 traces to reach 80% of success whereas MVA$_{TR}$ (the second best attack) needs 60000 traces. This represents a gain of 71%. For $\sigma = 8$ (see Fig. 3e), ROPT$_3$ needs 65000 traces to reach 80% of success whereas the MVA$_{TR}$ and the 2O-CPA needs 120000 traces. This represents a gain of 85%. And when the noise increases to $\sigma = 9$ (see Fig. 3f), ROPT$_3$ needs 120000 traces to reach 80% of success whereas 2O-CPA (the second best attack) needs 200000 traces, which is a gain of 66%.

These results can be interpreted as follows: The MVA$_{TR}$ is a third order attack which depends on the third order moment. The 2O-CPA is a second order attack which depends on the second order moment. The new ROPT$_3$ attack combines these two moments. When the noise is low the MVA$_{TR}$ and the ROPT$_3$ performs similarly; this shows that the dominant term in the Taylor expansion is the third order one. At the opposite when the noise increases the ROPT$_3$ becomes close to the 2O-CPA which indicates that the important term in the Taylor expansion is the second order one. As ROPT$_3$ combines the second and the third order moment weighted by the SNR it is always better than any attack exploiting only one moment.

7 Conclusions and Perspectives

In this article, we derived new attacks based on the Lth degree Taylor expansion in the SNR of the optimal Maximum Likelihood distinguisher. We have shown that this Lth degree truncation allows to target a moment of order L. The new attack outperforms the optimal distinguisher with respect to time complexity. In fact as we have theoretically shown, the Taylor approximation can be effectively computed whereas the fully optimal maximum likelihood distinguisher, was not computationally tractable.



(a) $\sigma = 1$

(b) $\sigma = 3$

(c) $\sigma = 6$

(d) $\sigma = 7$

(e) $\sigma = 8$

(f) $\sigma = 9$

(g) $\sigma = 12$

(h) $\sigma = 13$

Fig. 3: Attack on shuffled table recomputation

We have illustrated this property by applying our new method in a complex scenario of "shuffled table recomputation" and have compared the time complexity of the new attack and the optimal distinguisher. In addition, we have shown that in this context our attack has a higher success rate than all the attacks of the state-of-art over all possible noise variances.

An open question is how to quantify the accuracy of the approximation LL $\longrightarrow$ LL_ℓ as a function of the noise. In other words, what is the optimal degree of the Taylor expansion of the likelihood for a given SNR? Another interesting extension of this framework would be on hardware devices which are known to leak at various orders (see the real-world examples in [21–23]).

A Computation of the Moments

A.1 Computation of μ_1

There is no computational difficulty:

$$\mu_1 = \mathbb{E}(f_0) + \mathbb{E}(f_1) + \sum_{i\in I} \mathbb{E}(f_i) + \sum_{j\in J} \mathbb{E}(f_j). \tag{22}$$

Now, when there is no φ in the R.V., then the expectation is only on M (indeed, $\frac{1}{2^n!}\sum_{\varphi\in S_{2^n}} 1 = 1$). Thus,

$$E(f_0) = \frac{1}{2^n}\sum_{m\in\mathbb{F}_2^n} (x_0 - w_H(S[t\oplus k]\oplus m))^2 = \frac{1}{2^n}\sum_{m\in\mathbb{F}_2^n} (x_0 - w_H(m))^2, \tag{23}$$

which cannot further be simplified (in the simulations, it will be computed by the computer).

Similarly

$$E(f_1) = \frac{1}{2^n}\sum_{m\in\mathbb{F}_2^n} (x_1 - w_H(S[t\oplus k]\oplus m))^2 = \frac{1}{2^n}\sum_{m\in\mathbb{F}_2^n} (x_1 - w_H(m))^2. \tag{24}$$

When there is an expectation on Φ, then at order one, it considers **only one value** $\Phi(\omega)$. It is uniformly distributed, hence one can replace the expectation on Φ by an expectation on one value of φ, we call M'. For instance:

$$\begin{aligned}\mathbb{E}(f_i) &= \frac{1}{2^n!}\sum_{\varphi\in S_{2^n}} (x_i - w_H(\varphi(\omega)))^2 \\ &= \frac{1}{2^n}\sum_{m'\in\mathbb{F}_2^n} (x_i - w_H(m'))^2, \end{aligned}\tag{25}$$

which can thus be computed with the same *average* method as $\mathbb{E}(f_0)$.

Lastly, when there is both M and $\Phi(\omega)$, then whichever variable can absorb the other one, since both are uniformly distributed on $\mathbb{F}_2^n$. This means that:

$$\begin{aligned}
\mathbb{E}(f_j) &= \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} \frac{1}{2^n!} \sum_{\varphi \in S_{2^n}} (x_j - w_H(\varphi(\omega) \oplus m))^2 \\
&= \frac{1}{2^{2n}} \sum_{m,m' \in \mathbb{F}_2^n} (x_j - w_H(m \oplus m'))^2 \\
&= \frac{1}{2^{2n}} \sum_{\widetilde{m},m' \in \mathbb{F}_2^n} (x_j - w_H(\widetilde{m} \oplus m' \oplus m'))^2 \qquad \text{where } \widetilde{m} = m \oplus m' \qquad (26) \\
&= \frac{1}{2^n} \sum_{\widetilde{m} \in \mathbb{F}_2^n} (x_j - w_H(\widetilde{m}))^2, \qquad (27)
\end{aligned}$$

which is once again a similar computation as done for computing $\mathbb{E}(f_0)$.

A.2 Computation of μ_2

Recall that only the key dependent terms of μ_2 are needed for ROPT$_2$ and ROPT$_3$.

Notice that the square terms are computed as the non-square terms. For instance,

$$\mathbb{E}(f_0^2) = \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} (x_0 - w_H(S[t \oplus k] \oplus m))^4 = \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} (x_0 - w_H(m))^4, \qquad (28)$$

which we drop since it does not depend on k. All in one, the only key-dependent term is:

$$\mathbb{E}(f_0 \times f_1) = \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} (x_0 - w_H(S[t \oplus k] \oplus m))^2 (x_1 - w_H(m))^2, \qquad (29)$$

which cannot be further simplified and will be computed by the computer. So, for the purpose of the attack, we have:

$$\mu_2 = \mathbb{E}(f_0 \times f_1) + \text{cst}. \qquad (30)$$

A.3 Computation of μ_3

We shall consider only terms which depend on the key, hence product of three terms, one of which (at least) is f_0. Obviously, $\mathbb{E}(f_0^3)$ does not depend on k, for the same reason as given in Eqn. (28). But the two terms:

1. $\mathbb{E}(f_0^2 f_1)$ and
2. $\mathbb{E}(f_0 f_1^2)$

Notice that they are present $\binom{3}{2} = 3$ times each when developing the cube.

Interestingly, those are **not** the only cases where f_0 and f_1 are selected.

$$
\begin{aligned}
&\mathbb{E}(f_0 f_1 f_j) \\
&= \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} \frac{1}{2^n!} \sum_{\varphi \in S_{2^n}} (x_0 - w_H(S[t \oplus k] \oplus m))^2 (x_1 - w_H(m))^2 (x_j - w_H(\varphi(\omega) \oplus m))^2 \\
&= \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} \frac{1}{2^n} \sum_{m' \in \mathbb{F}_2^n} (x_0 - w_H(S[t \oplus k] \oplus m))^2 (x_1 - w_H(m))^2 (x_j - w_H(m' \oplus m))^2 \\
&= \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} (x_0 - w_H(S[t \oplus k] \oplus m))^2 (x_1 - w_H(m))^2 \frac{1}{2^n} \sum_{m' \in \mathbb{F}_2^n} (x_j - w_H(m' \oplus m))^2 \\
&= \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} (x_0 - w_H(S[t \oplus k] \oplus m))^2 (x_1 - w_H(m))^2 \frac{1}{2^n} \sum_{\widetilde{m'} \in \mathbb{F}_2^n} (x_j - w_H(\widetilde{m'}))^2 \quad \text{(As in Eq. (26))} \\
&= \mathbb{E}(f_0 f_1) \mathbb{E}(f_j).
\end{aligned}
$$

Similarly, we have:

$$\mathbb{E}(f_0 f_1 f_i) = \mathbb{E}(f_0 f_1) \mathbb{E}(f_i).$$

Now, we consider products without f_1. Obviously, taking only f_0 and f_i is not enough, since: $\mathbb{E}(f_0^2 f_i) = \mathbb{E}(f_0^2)\mathbb{E}(f_i)$ and $\mathbb{E}(f_0 f_i^2) = \mathbb{E}(f_0)\mathbb{E}(f_i^2)$ are key independent. The same goes for $\mathbb{E}(f_0^2 f_j)$ and $\mathbb{E}(f_0 f_j^2)$. We are left with $\mathbb{E}(f_0 f_i f_{i'})$, $\mathbb{E}(f_0 f_j f_{j'})$, and $\mathbb{E}(f_0 f_i f_j)$.

The term $\mathbb{E}(f_0 f_i f_{i'}) = \mathbb{E}(f_0)\mathbb{E}(f_i f_{i'}))$ does not depend on k, because there is no M in f_i.

The term $\mathbb{E}(f_0 f_j f_{j'})$ can also factorize as $\mathbb{E}(f_0)\mathbb{E}(f_j f_{j'}))$, hence it does not depend on k. The reason is more subtle, so we detail it:

$$
\begin{aligned}
\mathbb{E}(f_0 f_j f_{j'}) = {} & \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} (x_0 - w_H(S[t \oplus k] \oplus m))^2 \\
& \times \frac{1}{2^n(2^n - 1)} \sum_{\substack{(m', m'') \in \mathbb{F}_2^n \times \mathbb{F}_2^n \\ \text{s.t. } m' \neq m''}} (x_j - w_H(m' \oplus m))^2 (x_{j'} - w_H(m'' \oplus m))^2.
\end{aligned}
$$

Now, the second sum does not depend on m, as shown below:

$$
\begin{aligned}
& \frac{1}{2^n(2^n - 1)} \sum_{\substack{(m', m'') \in \mathbb{F}_2^n \times \mathbb{F}_2^n \\ \text{s.t. } m' \neq m''}} (x_j - w_H(m' \oplus m))^2 (x_{j'} - w_H(m'' \oplus m))^2 = \\
& \frac{1}{2^n} \sum_{m' \in \mathbb{F}_2^n} (x_j - w_H(m' \oplus m))^2 \frac{1}{2^n - 1} \sum_{m'' \in \mathbb{F}_2^n \setminus \{m'\}} (x_{j'} - w_H(m'' \oplus m))^2 = \\
& \frac{1}{2^n} \sum_{\widetilde{m'} \in \mathbb{F}_2^n} (x_j - w_H(\widetilde{m'}))^2 \frac{1}{2^n - 1} \sum_{m'' \in \mathbb{F}_2^n \setminus \{\widetilde{m'} \oplus m\}} (x_{j'} - w_H(m'' \oplus m))^2 = \\
& \frac{1}{2^n} \sum_{\widetilde{m'} \in \mathbb{F}_2^n} (x_j - w_H(\widetilde{m'}))^2 \frac{1}{2^n - 1} \sum_{\widetilde{m''} \in \mathbb{F}_2^n \setminus \{\widetilde{m''} \xcancel{\oplus m \oplus m}\}} (x_{j'} - w_H(\widetilde{m''}))^2.
\end{aligned}
$$

Consequently, the last case is $\mathbb{E}(f_0 f_i f_j)$. We can subdivide it into two cases: $j = i + 2^n$ and $j \neq i + 2^n$. When $j = i + 2^n$, the permutation Φ is evaluated at the same ω in f_i and f_j. We denote by M' the R.V. $\Phi(\omega)$, where $\omega = j - 2$. Hence:

$$\mathbb{E}(f_0 f_i f_{j=i+2^n}) = \\ \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} (x_0 - w_H(S[t \oplus k] \oplus m))^2 \frac{1}{2^n} \sum_{m' \in \mathbb{F}_2^n} (x_i - w_H(m'))^2 (x_j - w_H(m' \oplus m))^2. \tag{31}$$

These terms (for all $j \in J$) correspond to the MVA$_{TR}$ attack published at CHES 2015 [7].

Eventually, there are the terms for $j \neq i - 2^n$. They are actually key dependent, hence must be kept. They are equal to:

$$\mathbb{E}(f_0 f_i f_{j \neq i+2^n}) = \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} (x_0 - w_H(S[t \oplus k] \oplus m))^2 \\ \times \frac{1}{2^n} \frac{1}{2^n - 1} \sum_{\substack{(m', m'') \in \mathbb{F}_2^n \times \mathbb{F}_2^n \\ \text{s.t. } m' \neq m''}} (x_i - w_H(m'))^2 (x_j - w_H(m'' \oplus m))^2.$$

Interestingly, without the constraint $m' \neq m''$, this quantity does not depend on the key. So, the leakage which is exploited here is due to the fact Φ is not a random function, but a bijection. As, in μ_3, we are only interested in non constant terms, we can rewrite:

$$\begin{aligned} \mathbb{E}(f_0 f_i f_{j \neq i+2^n}) &= \text{cst} - \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} (x_0 - w_H(S[t \oplus k] \oplus m))^2 \\ &\quad \times \frac{1}{2^n} \frac{1}{2^n - 1} \sum_{\substack{(m', m'') \in \mathbb{F}_2^n \times \mathbb{F}_2^n \\ \text{s.t. } m' = m''}} (x_i - w_H(m'))^2 (x_j - w_H(m'' \oplus m))^2 \\ &= \text{cst} - \frac{1}{2^n} \sum_{m \in \mathbb{F}_2^n} (x_0 - w_H(S[t \oplus k] \oplus m))^2 \\ &\quad \times \frac{1}{2^n - 1} \sum_{m' \in \mathbb{F}_2} (x_i - w_H(m'))^2 (x_j - w_H(m' \oplus m))^2. \end{aligned} \tag{32}$$

The non-constant term is similar to Eqn. (31) provided a scaling by $-(2^n - 1)/2^n$ is done.

So, for the purpose of the attack, we have:

$$\begin{aligned} \mu_3 &= \text{cst} + 3\mathbb{E}(f_0^2 f_1) + 3\mathbb{E}(f_0 f_1^2) + 3!\mathbb{E}(f_0 \times f_1) \left(\sum_{i \in I} \mathbb{E}(f_i) + \sum_{j \in J} \mathbb{E}(f_j) \right) \\ &\quad + 3! \sum_{i=2}^{2^n+1} \mathbb{E}(f_0 f_i f_{j=i+2^n}) + 3! \sum_{i=2}^{2^n+1} \sum_{j \in \{2+2^n, \ldots, 2^{n+1}+1\} \setminus \{i+2^n\}} \mathbb{E}(f_0 f_i f_j). \end{aligned} \tag{33}$$

B Complexity Proofs

B.1 Proof of Lemma 1

In order to prove Lemma 1 let us first introduce a preliminary result.

Lemma 2. *The quantity* $\binom{\varPi}{\ell}$ *is increasing if* $\ell < \lceil \varPi/2 \rceil$ *and its maximum is* $\binom{\varPi}{\lceil \frac{\varPi}{2} \rceil}$.

Proof.

$$\binom{\varPi}{\ell+1} = \frac{\varPi!}{(\varPi-\ell-1)!(\ell+1)!} = \frac{\varPi-\ell-1}{\ell+1}\binom{\varPi}{\ell},$$

and the factor $\frac{\varPi-\ell-1}{\ell+1}$ is strictly greater than 1. Indeed,

$$\frac{\varPi-\ell-1}{\ell+1} > 1 \iff \varPi > 2(\ell+1) \iff \ell < \lceil \varPi/2 \rceil .$$

□

Finally we can prove Lemma 1.

Proof. Let us first assume that one dimension leaks at most one element of the permutation. We can thus develop the expression of μ_ℓ, and we denote the complexity under the braces.

$$\begin{aligned} \mu_\ell &= \mathbb{E}_R \left(\|x - y(t,k,R)\|^{2\ell} \right) \\ &= \underbrace{\sum_{k_1+\ldots+k_D=\ell}}_{\binom{D+\ell-1}{\ell}} \frac{\ell!}{\prod_{d=1}^{D} k_d!} \underbrace{\mathbb{E}_R}_{2^{(\varOmega-1)n}\binom{\varPi}{\lceil \frac{\varPi}{2} \rceil}} \underbrace{\left(\prod_{d=1}^{D} f_d^{k_d} \right)}_{\min(D,\ell)} \end{aligned}$$

As $k_1 + \ldots + k_D = \ell$ there are at most D indices $k_d, 1 \leq d \leq D$ such that $k_d \neq 0$. Hence there are at most $\min(D,\ell)$ elements in the product.

Each dimensions which leaks an element of the permutation can also leaks the masks. The worst case in terms of complexity is when all the permutation leakages depend also on the masks. Let us denote by i such that $1 \leq i \leq \min(D,\ell)$ the number of those terms. Then the expectation is computed over $2^{(\varOmega-1)n}\frac{\varPi!}{(\varPi-i)!}$. Nevertheless by taking into account the commutativity properties of the product one can only compute $2^{(\varOmega-1)n}\binom{\varPi}{i}$.

By Lemma 2 we have that is value $\binom{\varPi}{i}$ is maximum with $\binom{\varPi}{\ell}$ when $\ell \leq \lceil \frac{\varPi}{2} \rceil$. When $\ell > \frac{\varPi}{2} + 1$ the maximum is $\binom{\varPi}{\lceil \frac{\varPi}{2} \rceil}$.

Finally as there are $\binom{D+\ell-1}{\ell}$ elements in the sum.

The complexity of μ_ℓ is lower than $\mathcal{O}\left(\binom{D+\ell-1}{\ell} 2^{(\varOmega-1)n} \binom{\varPi}{\min(\lceil \frac{\varPi}{2} \rceil, \ell)}\right)$. □

B.2 Proof of Proposition 6

In order to prove Lemma 6 let us first introduce a preliminary result.

Lemma 3. *The quantity $\binom{D-1+\ell}{\ell}$ is increasing with ℓ if $D > 1$.*

Proof. We have that :

$$\binom{D-1+\ell+1}{\ell+1} = \frac{D+\ell}{\ell+1}\binom{D-1+\ell}{\ell},$$

where $\forall \ell$, $\frac{D+\ell}{\ell+1} > 1$ provided $D > 1$. □

Finally let us prove Prop. 6.

Proof.

Complexity of OPT*:* Following Eq. (2) we have that the computation for a key guess of OPT is:

$$\underbrace{\sum_{q=1}^{Q}}_{Q} \log \underbrace{\mathbb{E}}_{\Pi!2^{n(\Omega-1)}} \exp \underbrace{\frac{-\|x-y(t,k,R)\|^2}{2\sigma^2}}_{D}. \tag{34}$$

We assume that the computation of the log and the exp is constant. As a consequence the complexity of the optimal distinguisher is $\mathcal{O}\left(Q \cdot (2^n)^{\Omega-1} \cdot \Pi! \cdot D\right)$

Complexity of ROPT$_L$*:* The computation of ROPT$_L$ involves the computation of the μ_ℓ with $\ell \leq L$ (Eq. (2) and Eq. (1)). By Lemma 1 and Lemma 3 all these terms have a complexity lower than $\mathcal{O}\left(\binom{D+L-1}{L} \cdot 2^{(\Omega-1)n} \cdot \binom{\Pi}{\min(\lceil\frac{\Pi}{2}\rceil, L)}\right)$ (Eq. (16)).

As a consequence the complexity of ROPT$_L$ is lower than

$$\mathcal{O}\left(Q \cdot L\binom{D+L-1}{L} \cdot 2^{(\Omega-1)n} \cdot \binom{\Pi}{\min\left(\lceil\frac{\Pi}{2}\rceil, L\right)}\right). \tag{35}$$

□

B.3 Proof of Proposition 7

Proof. Let us develop all the product in the term μ_ℓ in order to compute the expectation in the minimum number of values.

$$\begin{aligned}
\mu_\ell &= \mathbb{E}_M\Big(\big(\sum_{d=1}^{D}(x_{d,q}-y_d)^2\big)^\ell\Big) \\
&= \sum_{\substack{\ell_1,\ell_2,\ldots,\ell_D \\ \sum_{d=1}^{D}=\ell}} \frac{\ell!}{\prod_{d=1}^{D}\ell_d!}\mathbb{E}_M\big((x_1-y_1)^{2\ell_1}\cdots(x_D-y_D)^{2\ell_D}\big).
\end{aligned}$$

Moreover $(x_d - y_d(t,k,M))^{2\ell_d} = \sum_{i=0}^{2\ell_d} \binom{2\ell_d}{i} x_d^{2\ell_d - i} y_d(t,k,M)^i$

$$\mu_\ell = \sum_{\substack{\ell_1,\ell_2,\ldots,\ell_D \\ \sum_{d=1}^{D} \ell_d = \ell}} \frac{\ell!}{\prod_{d=1}^{D} \ell_d!} \mathbb{E}_M \left(\prod_{d=1}^{D} \left(\sum_{i=0}^{2\ell_d} \binom{2\ell_d}{i} x_d^{2\ell_1 - i} y_d(t,k,M)^i \right) \right)$$

$$= \sum_{\substack{\ell_1,\ell_2,\ldots,\ell_D \\ \sum_{d=1}^{D} \ell_d = \ell}} \frac{\ell!}{\prod_{d=1}^{D} \ell_d!} \sum_{\substack{i_1 \le 2\ell_1 \\ \vdots \\ i_D \le 2\ell_D}} \prod_{d=1}^{D} \left(\binom{2\ell_d}{i_d} x_d^{2\ell_d - i_d} \right) \underbrace{\mathbb{E}_M \left(\prod_{d=1}^{D} y_d(t,k,M)^{i_d} \right)}_{\text{can be precomputed}} .$$

□

B.4 Proof of Proposition 8

Proof. In our case study the size of the permutation is $\Pi = 2^n$.

Then the complexity of OPT is given by a straightforward application of Eq. (17).

From Eq. (14) we have that for ROPT$_2$ the computation for one key guess and one trace is given by $\mathbb{E}(f_0 \times f_1)$. In this equation the expectation is computed over 2^n values (Eq. (28)).

From Eq. (15) we have that for ROPT$_3$ the computation for one key guess and one trace is given by $\mu_2^{(q)}(1 + \gamma\mu_1^{(q)}) - \gamma\frac{\mu_3^{(q)}}{3}$. It can be seen in Eq. (23), Eq. (24), Eq. (25) and Eq. (27) that the expectation of μ_1 is computed over 2^n values. The dominant term in μ_3 (Eq. (33)) is :

$$\underbrace{\sum_{i=2}^{2^n+1} \sum_{j \in \{2+2^n,\ldots,2^{n+1}+1\} \backslash \{i+2^n\}}}_{2^{2n}} \underbrace{\mathbb{E}}_{2^{2n}}(f_0 f_i f_j).$$

The expectation in this term is computed over 2^{2n} values (Eq. (32)). The sum is computed on less than 2^{2n}. □

B.5 Time and complexity

The times of the section are expressed in seconds. All the attacks have been run on Intel Xeon X5660 running at 2.67 GHz. All the implementations are mono-thread. The model of the simulations is the one describe in Sec. 6. For each distinguisher the attacks are computed 1000 times on 1000 traces.

C Analysis of the DPAcontest.

Recently an open implementation of a masking scheme with shuffling has been presented in the DPA contest v4.2 [35]. In this implementation the execution of the different states is performed in an random order.

Attack	Dimension	Time (in seconds)	Computational Complexity
2O-CPA	2	39	$\mathcal{O}\left(Q\right)$
ROPT$_2$	2	295	$\mathcal{O}\left(Q\right)$
OPT$_{2O}$	2	9473	$\mathcal{O}\left(Q\cdot\left(2^n\right)\right)$
MVA$_{TR}$	$2^{n+1}+1$	130	$\mathcal{O}\left(Q\cdot 2^n\right)$
ROPT$_3$	$2^{n+1}+2$	2495	$\mathcal{O}\left(Q\cdot 2^{2n}\right)$
OPT	$2^{n+1}+2$	Not computable	$\mathcal{O}\left(Q\cdot\left(2^n\right)\cdot 2^n!\cdot\left(2^{n+1}+2\right)\right)$

Table 1: Time and complexity

An attacker can target the integrated leakages of the different states in order to counter the shuffling [9, 31].

A better approach is to take into account the possible leakages of the permutation. In this case the optimal distinguisher will be not computable as it involves an expectation over 16! values. In this case the rounded optimal attack will reduced this complexity.

Let us defined the leakages of such implementations.

- $X_0 = y_0\left(t,k,R\right)+N_0$ with $y_0\left(t,k,R\right)=w_H(M)$,
- $X_1 = y_1\left(t,k,R\right)+N_1$ with $y_1\left(t,k,R\right)=w_H(S[\pi\left(T\oplus k\right)]\oplus M)$,
- $X_i = y_i\left(t,k,R\right)+N_i$, for $i=2,\ldots,18$ with $y_i\left(t,k,R\right)=w_H(\Phi(i-2))$,

Then similarly to the Appendix A we have that:

$$\mu_1 = \mathbb{E}(f_0)+\mathbb{E}(f_1)+\sum_{i\in I}\mathbb{E}(f_i), \tag{36}$$

$$\mu_2 = \mathbb{E}(f_0\times f_1)+\text{cst}. \tag{37}$$

Additionally as it is a low entropy masking scheme the secret key can leaked in an univariate high order attack. Depending on the number of masks involve in the masking scheme it could be at order 2, 3 or more. For simplicity let us assume it is at order 3. In such cases

$$\mu_3 = \mathbb{E}(f_1^3)+3\mathbb{E}(f_0^2 f_1)+3\mathbb{E}(f_0 f_1^2)+3!\sum_{i=2}^{2^n+1}\mathbb{E}(f_0 f_1 f_i)+\text{cst}. \tag{38}$$

Of course an attacker can additionally exploit all the leakages of the different states in order to increase the success of the attacks.

In some particular low entropy masking schemes the same masks are reused several time or are linked by deterministic relations (e.g the first version of the DPAcontest). In this context it could be interesting to combine the leakages of different states [4]. In this case our method could benefit of the multiple possible points combinations.

References

1. Mehdi-Laurent Akkar and Christophe Giraud. An Implementation of DES and AES Secure against Some Attacks. In LNCS, editor, *Proceedings of CHES'01*, volume 2162 of *LNCS*, pages 309–318. Springer, May 2001. Paris, France.
2. Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual Information Analysis: a Comprehensive Study. *J. Cryptology*, 24(2):269–291, 2011.
3. Johannes Blömer, Jorge Guajardo, and Volker Krummel. Provably Secure Masking of AES. In Helena Handschuh and M. Anwar Hasan, editors, *Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 69–83. Springer, 2004.
4. Nicolas Bruneau, Jean-Luc Danger, Sylvain Guilley, Annelie Heuser, and Yannick Teglia. Boosting Higher-Order Correlation Attacks by Dimensionality Reduction. In Rajat Subhra Chakraborty, Vashek Matyas, and Patrick Schaumont, editors, *Security, Privacy, and Applied Cryptography Engineering - 4th International Conference, SPACE 2014, Pune, India, October 18-22, 2014. Proceedings*, volume 8804 of *Lecture Notes in Computer Science*, pages 183–200. Springer, 2014.
5. Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Less is More – Dimensionality Reduction from a Theoretical Perspective. In Handschuh and Güneysu [13].
6. Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, and Olivier Rioul. Masks Will Fall Off – Higher-Order Optimal Distinguishers. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 344–365. Springer, 2014.
7. Nicolas Bruneau, Sylvain Guilley, Zakaria Najm, and Yannick Teglia. Multi-variate High-Order Attacks of Shuffled Tables Recomputation. In Handschuh and Güneysu [13].
8. 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.
9. Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Çetin Kaya Koç and Christof Paar, editors, *CHES*, volume 1965 of *Lecture Notes in Computer Science*, pages 252–263. Springer, 2000.
10. Jean-Sébastien Coron. Higher Order Masking of Look-Up Tables. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 441–458. Springer, 2014.
11. A. Adam Ding, Liwei Zhang, Yunsi Fei, and Pei Luo. A statistical model for higher order DPA on masked devices. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 147–169. Springer, 2014.
12. Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings,*

Part I, volume 9056 of *Lecture Notes in Computer Science*, pages 401–429. Springer, 2015.

13. Helena Handschuh and Tim Güneysu, editors. *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015. Proceedings*, volume 9293 of *Lecture Notes in Computer Science*. Springer, 2015.
14. Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. An AES Smart Card Implementation Resistant to Power Analysis Attacks. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *ACNS*, volume 3989 of *Lecture Notes in Computer Science*, pages 239–252, 2006.
15. Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, August 17–21 2003. Santa Barbara, California, USA.
16. Kerstin Lemke-Rust and Christof Paar. Analyzing side channel leakage of masked implementations with stochastic methods. In Joachim Biskup and Javier Lopez, editors, *Computer Security - ESORICS 2007, 12th European Symposium On Research In Computer Security, Dresden, Germany, September 24-26, 2007, Proceedings*, volume 4734 of *Lecture Notes in Computer Science*, pages 454–468. Springer, 2007.
17. Kerstin Lemke-Rust and Christof Paar. Gaussian Mixture Models for Higher-Order Side Channel Analysis. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES*, volume 4727 of *LNCS*, pages 14–27. Springer, 2007.
18. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.
19. Thomas S. Messerges. Securing the AES Finalists Against Power Analysis Attacks. In *Fast Software Encryption'00*, pages 150–164. Springer-Verlag, April 2000. New York.
20. Thomas S. Messerges. Using second-Order Power Analysis to Attack DPA resistant Software. In *CHES*, volume 1965 of *LNCS*, pages 71–77. Springer, August 17-18 2000. Worcester, MA, USA.
21. Amir Moradi. Statistical tools flavor side-channel collision attacks. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 428–445. Springer, 2012.
22. Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. *IACR Cryptology ePrint Archive*, 2014:409, June 2 2014.
23. Amir Moradi and Alexander Wild. Assessment of hiding the higher-order leakages in hardware - what are the achievements versus overheads? In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 453–474. Springer, 2015.
24. Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. *J. Cryptology*, 24(2):292–321, 2011.
25. Elisabeth Oswald and Stefan Mangard. Template Attacks on Masking — Resistance Is Futile. In Masayuki Abe, editor, *CT-RSA*, volume 4377 of *Lecture Notes in Computer Science*, pages 243–256. Springer, 2007.
26. Jing Pan, Jerry I. den Hartog, and Jiqiang Lu. You cannot hide behind the mask: Power analysis on a provably secure S-box implementation. In Heung Youl Youm and Moti Yung, editors, *Information Security Applications, 10th International*

Workshop, WISA 2009, Busan, Korea, August 25-27, 2009, Revised Selected Papers, volume 5932 of *Lecture Notes in Computer Science*, pages 178–192. Springer, 2009.

27. Éric Peeters, François-Xavier Standaert, Nicolas Donckers, and Jean-Jacques Quisquater. Improved Higher-Order Side-Channel Attacks with FPGA Experiments. In *CHES*, volume 3659 of *LNCS*, pages 309–323. Springer, 2005.
28. Emmanuel Prouff and Matthieu Rivain. A Generic Method for Secure SBox Implementation. In Sehun Kim, Moti Yung, and Hyung-Woo Lee, editors, *WISA*, volume 4867 of *Lecture Notes in Computer Science*, pages 227–244. Springer, 2007.
29. Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.
30. Matthieu Rivain and Emmanuel Prouff. Provably Secure Higher-Order Masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *LNCS*, pages 413–427. Springer, 2010.
31. Matthieu Rivain, Emmanuel Prouff, and Julien Doget. Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers. In *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 171–188. Springer, September 6-9 2009. Lausanne, Switzerland.
32. François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, volume 5479 of *LNCS*, pages 443–461. Springer, April 26-30 2009. Cologne, Germany.
33. François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.
34. Alan Stuart and Keith Ord. *Kendall's Advanced Theory of Statistics: Distribution Theory.* Wiley-Blackwell, June 2 1994. 6th Edition. ISBN-10: 0470665300; ISBN-13: 978-0470665305.
35. TELECOM ParisTech SEN research group. DPA Contest (4^{th} edition), 2013–2014. `http://www.DPAcontest.org/v4/`.
36. Michael Tunstall, Carolyn Whitnall, and Elisabeth Oswald. Masking Tables - An Underestimated Security Risk. In Shiho Moriai, editor, *FSE*, volume 8424 of *Lecture Notes in Computer Science*, pages 425–444. Springer, 2013.
37. Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against Side-Channel Attacks: A Comprehensive Study with Cautionary Note. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 740–757. Springer, 2012.
38. Jason Waddle and David Wagner. Towards Efficient Second-Order Power Analysis. In *CHES*, volume 3156 of *LNCS*, pages 1–15. Springer, 2004. Cambridge, MA, USA.
39. Eric W Weisstein. Cumulant. From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/Cumulant.html`.