Multi-variate High-Order Attacks of Shuffled Tables Recomputation

Nicolas Bruneau[1,2(✉)], Sylvain Guilley[1,3], Zakaria Najm[1], and Yannick Teglia[2]

[1] TELECOM-ParisTech, Crypto Group, Paris, France
nicolas.bruneau@telecom-paristech.fr
http://www.telecom-paristech.fr/en/eng/home.html,
http://www.comelec.enst.fr/recherche/sen.en
[2] STMicroelectronics, AST Division, Rousset, France
http://www.st.com/
[3] Secure-IC S.A.S., Rennes, France
http://www.Secure-IC.com/

Abstract. Masking schemes based on tables recomputation are classical countermeasures against high-order side-channel attacks. Still, they are known to be attackable at order d in the case the masking involves d shares. In this work, we mathematically show that an attack of order strictly greater than d can be more successful than an attack at order d. To do so, we leverage the idea presented by Tunstall, Whitnall and Oswald at FSE 2013: we exhibit attacks which exploit the multiple leakages linked to one mask during the recomputation of tables. Specifically, regarding first-order table recomputation, improved by a shuffled execution, we show that there is a window of opportunity, in terms of noise variance, where a novel highly multivariate third-order attack is more efficient than a classical bivariate second-order attack. Moreover, we show on the example of the high-order secure table computation presented by Coron at EUROCRYPT 2014 that the window of opportunity enlarges linearly with the security order d.

Keywords: Shuffled table recomputation · Highly multivariate high-order attacks · Signal-to-noise ratio

1 Introduction

For several years now Side-Channel Attacks (SCA [13]) have been a threat against cryptographic algorithms in embedded systems. To protect cryptographic implementations against these attacks several countermeasures and protection techniques have been developed. Data masking schemes [12] are widely used since their security can be formally grounded.

The rationale of masking schemes goes as follows: each sensitive variable is randomly splitted in d shares (using $d-1$ masks), in such a way that any tuple of $d-1$ shares manipulated during the masked algorithm is independent from any sensitive variable. Masking schemes are the target of higher-order SCA

T. Güneysu and H. Handschuh (Eds.): CHES 2015, LNCS 9293, pp. 475–494, 2015.
DOI: 10.1007/978-3-662-48324-4_24

[5,16,20,25]. A dth-order attack combines the leakages of d shares. A particular difficulty in the implementation of masking schemes is to compute non-linear parts of the algorithm, such as for example the S-Box of AES (a function from n bits to n bits). To solve this difficulty different methods have been proposed which can be classified in three categories [14].

- Algebraic methods [2,21]. The outputs of the S-Box will be computed using the algebraic representation of the S-box.
- Global Look-up Table [19,23] method. A table is precomputed off-line for each possible input and output masks.
- Table recomputation methods which precompute a masked S-Box stored in a table [1,5,15]; such tables can be recomputed only once per encryption to reach first-order security. More recently, Coron presented at EUROCRYPT 2014 [7] a table recomputation scheme secure against dth-order attacks. Since this countermeasure aims at high-order security ($d > 1$), it requires one table recomputation at each S-Box call.

These methods provide security against Differential Power Analysis [13] (DPA) or Higher-Order DPA (HODPA). Still, whatever the protection order, there is *at least one* leakage associated to each share: in practice, shares (typically masks) can leak *more than once*. For example attacks exploiting the multiplicity of leakages of the same mask during the table recomputation have been presented by Pan et al. in [18] and more recently by Tunstall et al. in [24]. Such attacks consist in guessing the mask in a first order horizontal Correlation Power Analysis [3] (CPA) and then conducting a first-order vertical CPA knowing the mask. Variants consist in using a machine learning technique to extract the mask [9]. Globally, we refer to these attacks as Horizontal-Vertical attacks (HV attacks).

Shuffling the table recomputation makes the HV attacks more difficult. Still shuffling can be bypassed if the random permutation is generated from a seed with low entropy, since both the mask and the shuffling seed can be guessed [24].

Our Contributions. Our first contribution is to describe a new HODPA tailored to target the table recomputation despite a highly entropic masking (unexploitable by exhaustive search). More precisely, we propose an innovative combination function, which has the specificity to be highly multivariate. We relate the combination function of HODPA attacks to their expected signal-to-noise ratio, which allows for a straightforward comparison the attacks based on their success rate. In particular, we compare the success rates of our highly multivariate HODPA (exploiting leakages in the table recomputation as well as in the masked algorithm, where the secret key is used) and of a state-of-the-art HODPA (exploiting only the leakages within the masked algorithm). Our analysis reveals that there is a window of opportunity, when the noise variance is smaller than a threshold, where our new HODPA is more successful than a straightforward HODPA, despite it is of higher-order.

For instance in this paper we attack a first-order masking scheme based on table recomputation with a $(2^{n+1}+1)$-variate third-order attack more efficiently

than with a classical bivariate second-order attack. In this case HV attacks could not be applied. This is the first time that a non minimal order attack is proved better (in terms of success rate) than the attack of minimal order. Actually, this non intuitive result arises from a relevant selection of leaking samples — this question is seldom addressed in the side-channel literature. We generalize our attack to a higher-order masking scheme based on tables recomputation (Coron, EUROCRYPT 2014), and prove that it remains better than a classical attack, with a window of opportunity that actually grows linearly with the masking order d.

Outline of the Paper. The rest of the paper is organized as follows. Sect. 2 introduce the notations used in this article. Sect. 3 provides a reminder on table recomputation algorithms and on the way to defeat and protect this algorithm using random permutations. In Sect. 4 we propose a new attack against the "protected" implementation of the table recomputation, prove theoretically the soundness of the attack and validate these results by simulation. In Sect. 5 we apply this attack on a higher-order masking scheme. Sect. 6 extends our results to the case where the leakage function is affine. Finally in Sect. 7 we validate our results on real traces.

2 Preliminary and Notations

In this article capital letters (e.g., U) denote random variables and lowercase letters denote their realizations (e.g., u).

Let $k^\star$ be the secret key of the cryptographic algorithm. T denotes the input or the ciphertext. We suppose that the computations are done on n-bit words which means that these words can be seen as elements of $\mathbb{F}_2^n$. As a consequence both $k^\star$ and T are expected to be elements of $\mathbb{F}_2^n$. Moreover as we study protected implementations of cryptographic algorithms these algorithms also take as input a set of uniform independent random variables (not known by an attacker). Let denote by $\mathcal{R}$ this set.

Let g be a mapping which maps the input data to a *sensitive variable*. A *sensitive variable* is an internal variable proceeded by the cryptographic algorithm which depends on a subset of the inputs not known by the attacker (e.g. the secret key but also the secret random value). A measured leakage could be defined by:

$$X = \Psi\left(g\left(k^\star, T, \mathcal{R}\right)\right) + N, \tag{1}$$

where $\Psi : \mathbb{F}_2^n \to \mathbb{R}$ denotes the leakage function. This leakage function is a specific characteristic of the target device. The leakage function could be for example the Hamming Weight (denoted by HW in this article). The random variable N denotes an independent additive noise. In order to conduct a dth-order attack an attacker should combine the leakages of d shares. To combine these leakages an attacker will use a *combination function* [5,16,17]. The degree of this combination function must be at least d for the attack to succeed.

The *combination function* will then be applied both on the measured leakages and on the model (this is the optimal HODPA). As a consequence, an HODPA is completely defined by the *combination function* used.

In the rest of the paper the SNR is given by the following definition:

Definition 1 (Signal to Noise Ratio). *The Signal to Noise Ratio of a leakage denoted by a random variable L depending on informative part denoted I is given by:*

$$\mathrm{SNR}\left[L, I\right] = \frac{Var\left[\mathbb{E}\left[L|I\right]\right]}{\mathbb{E}\left[Var\left[L|I\right]\right]}. \tag{2}$$

An attack is said *sound* when it allows to recover the key $k^\star$ with success probability which tends to one when the number of measurements tends to the infinity.

3 Masking Scheme with Table Recomputation

3.1 Algorithm

In this article we consider Boolean masking schemes. In particular, we focus on schemes based on table recomputation where the masked S-Box is stored in a table and recomputed each time.

This algorithm begins by a key addition phase where one word of the plaintext t, one word the key k and a random mask word m, are Xored together.

Then, these values are passed through a non linear part stored in a table. The output of this operation could be masked by a different mask m'. Some linear operations could be done after the non linear part. Of course, in the whole algorithm, all the data are masked (exclusive-ored) with a random mask, to ensure the protection against first order attacks.

Masking the linear parts is straightforward but passing through the non linear one is less obvious. To realize this operation the table is recomputed. For all the elements of $\mathbb{F}_2^n$ the input mask is removed and then the output is masked by the output mask. In this step the key is never manipulated so all the leakages concern the mask. It can also be noticed that a new table S' of size $2^n \times n$ bits, is required for this step.

3.2 Classical Attacks

As the other masking schemes, masking schemes based on table recomputation can be defeated without the leakage of the table recomputation. Indeed an attacker can use:

- Second order attacks [5,16] such as second-order CPA (2O-CPA). It can be noticed that for such attacks, the adversary can also exploit the leakage of the mask during the table recomputation.
- Collisions attacks. If several S-Boxes are masked by the same mask the Collisions attacks may be practicable [6].

However these attacks do not take into account all the leakages due to table recomputation stage. An approach to exploit these leakages is to combine all of them with a leakage depending on the key. This method has been presented in [24] where an "horizontal" attack is performed on the table recomputation to recover the mask.

In such "horizontal" attacks two different steps can be targeted:

- An attacker could try to recover the output masks. In this case he should first recover the address in the table. In this case it is not necessary to recover the input mask but only the address value.
- An attacker could also try to recover the input masks.

The second step consists in a vertical attack which recover the key. In this second step the mask is now a known value. It can be noticed that the exact knowledge of the mask is not required to recover the key. Indeed if the probability to recover the mask is higher than $\frac{1}{2^n}$ then a first order attack is possible.

Recently, the optimal distinguisher in the case of masking has been studied in [4]: it is applied to the precomputation phase of masked table without shuffling in Sect. 5. This attack can be extended to the case of shuffled table recomputation but would require an enumeration of all shuffles, which is computationally unfeasible.

3.3 Classical Countermeasure

The strategy to protect the table recomputation against HV attacks and the distinguisher presented in [4] is to shuffle the recomputation, i.e. do the recomputation in a random order Algorithm 1.

Different methods to randomize the order are presented in [24]. One of the methods presented is based on a random permutation on a subset of $\mathbb{F}_2^n$.

If the random permutation over $\mathbb{F}_2^n$ is randomly drawn from a set of permutation $S \subset S_{2^n}$, where $card(S) \ll card(S_{2^n})$, it is still possible to take advantage of the table recomputation. Indeed as it is shown in [24] attacks could be built by including all the possible permutations in the key hypothesis. If the permutation is drawn over all S_{2^n} the number of added hypothesis is $2^n!$ which can be too much for attacks.

By generating permutation, such as defined in [24] or any pseudo random permutation generator (RC4 key scheduler...), a designer could protect table recomputation against HV attacks. Indeed using for example five or six bytes of entropy as seed for the permutation generator could be enough to prevent an attacker to guess all the possible permutations.

4 Totally Random Permutation and Attack

In this section we present a new attack against shuffled table recomputation. The success of this attack will not be impacted by the entropy used to generate the shuffle. As a consequence this attack will succeed when the HV attacks will failed

Algorithm 1. Shuffled Table recomputation

output: Mask SubBytes
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** $\varphi(\omega) \in \{\varphi(0), \varphi(1), \ldots, \varphi(2^n - 1)\}$ **do** // S-Box masking
4 $\quad z \leftarrow \varphi(\omega) \oplus m$ // Masked input ;
5 $\quad z' \leftarrow S[\varphi(\omega)] \oplus m'$ // Masked output ;
6 $\quad S'[z] = z'$ // Creating the masked S-Box entry ;
7 **end**
8 **return** S'

because of the quantity of entropy used to generate the shuffle. We then express the condition where this attack will outperform the state of the art second order attack.

4.1 Defeating the Countermeasure

As the permutation φ is completely random, the value of the current index in the loop for (line 3 to line 7) is unknown. But it can be noticed that this current index is manipulated twice at each step of the loop (line 4, line 5):

$$z \leftarrow \boldsymbol{\varphi(\omega)} \oplus m, \tag{3}$$
$$z' \leftarrow S[\boldsymbol{\varphi(\omega)}] \oplus m'. \tag{4}$$

It can be noticed that [20] if U is a random variable uniformly drawn over $\mathbb{F}_2^n$ and $m \in \mathbb{F}_2^n$ then:

$$\mathbb{E}\left[(\mathsf{HW}[U] - \mathbb{E}\left[\mathsf{HW}[U]\right]) \times (\mathsf{HW}[U \oplus m] - \mathbb{E}\left[\mathsf{HW}[U \oplus m]\right])\right] = -\frac{\mathsf{HW}[m]}{2} + \frac{n}{4}. \tag{5}$$

As a consequence, it may be possible for an attacker to exploit the leakage depending on the two (3, 4) manipulations of the current random index in the loop. Indeed, at each of the 2^n steps of the loop of the table recomputation, the leakage of the $\varphi(\omega)$ in Eqs. 3 and 4 which plays the role of U in Eq. 5 will be combined (by a centered product) to recover a variable depending on the mask. Then these 2^n variables will be combined together (by a sum) before being combined (again by a centered product) with a leakage depending on the key. This gives us a rough idea of the attack, also illustrated in Fig. 1.

An attacker could want to perform the attack on the output of the S-Box. But depending on the implementation of the masking scheme the output masks can be different for each value of the S-Box (see for example the masking scheme of Coron [7]). To avoid loss of generality we focus our study on the S-Box input mask of the recomputation. Indeed by design of the table recomputation masking scheme, the input mask is the same for each value of the S-Box: the attacker can thus exploit it multiple times. Moreover an attacker can still take advantage of

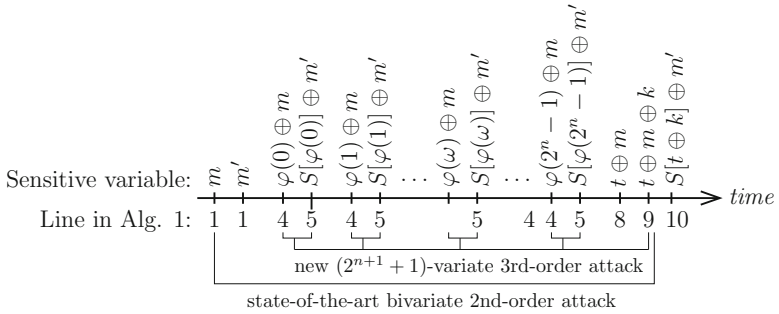


Fig. 1. State-of-the-art attack and new attack investigated in this article

the confusion of the S-Box [11] to better discriminate the various key candidates. Indeed he can target the input the of SubBytes operation of the last round.

4.2 Multivariate Attacks Against Table Recomputation

In the previous section, it is shown that at each turn of the loop of the table recomputation, it is possible to extract a value depending on the mask. As a consequence it is possible to use all of these values to perform a multivariate attack. In this subsection we give the formal formula of this new attack. Let us define the leakages of the table recomputation. The leakage of the masked random index in the loop is given by: $\mathsf{HW}[\Phi(\omega) \oplus M] + N_\omega^{(1)}$. The leakage of the random index is given by: $\mathsf{HW}[\Phi(\omega)] + N_\omega^{(2)}$.

Depending on the knowledge about the model, the leakage could be centered by the "true" expectation or by the estimation of this expectation. We assume this expectation is a known value given by: $\mathbb{E}\left[\mathsf{HW}[\Phi(\omega) \oplus m] + N_\omega^1\right] = \mathbb{E}\left[\mathsf{HW}[\Phi(\omega)] + N_\omega^2\right] = \frac{n}{2}$. Then let us denote by:

$$X_\omega^{(1)} = \mathsf{HW}[\Phi(\omega) \oplus M] + N_\omega^{(1)} - \frac{n}{2}, \tag{6}$$

$$X_\omega^{(2)} = \mathsf{HW}[\Phi(\omega)] + N_\omega^{(2)} - \frac{n}{2}. \tag{7}$$

Let us denote the leakage of the masked AddRoundKey:

$$X^\star = \mathsf{HW}[T \oplus M \oplus k^\star] + N - \frac{n}{2}. \tag{8}$$

To use all the leakages of the table recomputation an original combination function could be defined.

Definition 2. *The combination function exploiting the leakage of the table recomputation C_{tr} is given by:*

$$\begin{aligned} C_{tr}: \quad \mathbb{R}^{2^{n+1}} \times \mathbb{R} &\longrightarrow \mathbb{R} \\ \left(\left(X_\omega^{(1)}, X_\omega^{(2)}\right)_\omega, X^\star\right) &\longmapsto \left(-2 \times \tfrac{1}{2^n} \textstyle\sum_{\omega=0}^{2^n-1} X_\omega^{(1)} \times X_\omega^{(2)}\right) \times X^\star. \end{aligned}$$

Following the Fig. 1 it can be noticed that C_{tr} is in fact the combination of two sub-combination functions. Indeed first the leakages of the table recomputation are combined, the results of this combination is the following value:

$$X_{tr} = -2 \times \frac{1}{2^n} \sum_{\omega=0}^{2^n-1} X_{\omega}^{(1)} \times X_{\omega}^{(2)}. \tag{9}$$

Then this value is combined with $X^\star$.

Based on the combination function C_{tr}, a multivariate attack can be built.

Definition 3. *The MultiVariate Attack exploiting the leakage of the table recomputation is given by the function:*

$$\begin{aligned} MVA_{tr}: \quad \mathbb{R}^{2^{n+1}} \times \mathbb{R} \times \mathbb{R} &\longrightarrow \mathbb{F}_2^n \\ \left(\left(X_{\omega}^{(1)}, X_{\omega}^{(2)}\right)_{\omega}, X^\star, Y\right) &\longmapsto \underset{K \in \mathbb{F}_2^n}{\operatorname{argmax}}\, \rho\left[C_{tr}\left(\left(X_{\omega}^{(1)}, X_{\omega}^{(2)}\right)_{\omega}, X^\star\right), Y\right], \end{aligned}$$

where $Y = \mathbb{E}\left[\left(\mathsf{HW}[T \oplus M \oplus K] - \frac{n}{2}\right) \cdot \left(\mathsf{HW}[M] - \frac{n}{2}\right) | T, K\right]$ *and* ρ *the Pearson coefficient.*

Proposition 1. *MVA_{tr} is sound.*

Remark 1. *The attack presented in Definition 3 is a $2^{n+1}+1$ multivariate third order attack.*

Let us denote the leakage of the mask by:

$$X^{(3)} = \mathsf{HW}[M] + N^{(3)} - \frac{n}{2}. \tag{10}$$

In the rest of the paper we denote by 2O-CPA the CPA using the centered product as combination function.

$$\begin{aligned} \text{2O-CPA}: \quad \mathbb{R} \times \mathbb{R} \times \mathbb{R} &\longrightarrow \mathbb{F}_2^n \\ \left(X^{(3)}, X^\star, Y\right) &\longmapsto \underset{K \in \mathbb{F}_2^n}{\operatorname{argmax}}\, \rho\left[X^{(3)} \times X^\star, Y\right]. \end{aligned}$$

Using the Definitions 2, 3 and Eq. 9, it can be noticed that the only difference between the MVA_{tr} and the 2O-CPA is the use of X_{tr} instead of $X^{(3)}$. X_{tr} will act as the leakage of the mask. Let us call X_{tr} the *second order leakage.*

Remark 2. *The informative part of the second order leakage is the same as the informative part of the leakage mask i.e.*

$$\mathbb{E}\left[X_{tr} | M = m\right] = \mathbb{E}\left[X^{(3)} | M = m\right].$$

Proof. Straightforward application of the results of [20] □

4.3 Leakage Analysis

By using the formula of the theoretical success rate we show that as the same operations are targeted by the MVA_{tr} and the 2O-CPA then it is equivalent to compare the SNR and compare the SR of the attacks. Based on this fact we can theoretically establish the conditions in which the MVA_{tr} outperforms the 2O-CPA. This conditions are given in Theorem 2.

Recently A.A Ding et al. [10, Sect. 3.4] give the following formula to establish the Success Rate (SR) of second-order attacks:

$$\text{SR} = \Phi_{N_k-1}\left(\frac{\sqrt{b}\delta_0\delta_1}{4}K^{-1/2}\kappa\right).$$

In this formula:

- δ_0 denotes the SNR of the first share and δ_1 denotes the SNR of the second one;
- Φ_{N_k-1} denotes the cumulative distribution function of (N_k-1)-dimensional standard Gaussian distribution; as underlined by the authors in [10], if the noise distribution is not multi-variate Gaussian, then Φ_{N_k} is to be understood as its cumulative distribution function.
- N_k denotes the number of key candidates.
- K denotes the confusion matrix and κ the confusion coefficient.
- b denotes the number of traces.

This formula allows to establish the link between the SNR and SR of second order attacks against Boolean masking schemes.

Let us apply the A.A Ding et al. formula in the case of our two attacks:

$$\text{SR}_{\text{2O-CPA}} = \Phi_{2^n-1}\left(\sqrt{b}\frac{\text{SNR}\left[X^{(3)}, M\right]\text{SNR}\left[X^\star, (T, M)\right]}{4}K^{-1/2}\kappa\right),$$

$$\text{SR}_{\text{MVA}_{tr}} = \Phi_{2^n-1}\left(\sqrt{b}\frac{\text{SNR}\left[X_{tr}, M\right]\text{SNR}\left[X^\star, (T, M)\right]}{4}K^{-1/2}\kappa\right).$$

We target the same operation for the share that leaks the secret key ($X^\star$). Moreover by Remark 2 the informative parts of the leakages depending on the mask (X_{tr} and $X^{(3)}$) is the same in the two leakages. As a consequence the K and κ are the same in the two attacks.

It can be noticed that the only difference in the formulas of the success rate is the use of $\text{SNR}\left[X_{tr}, M\right]$ instead of $\text{SNR}\left[X^{(3)}, M\right]$. Then it is equivalent to compare these values and compare the SR of the attacks.

Theorem 2. *The* SNR *of the "second-order leakage" is greater than the* SNR *of the leakage of the mask if and only if*

$$\sigma^2 \leqslant 2^{n-2} - \frac{n}{2},$$

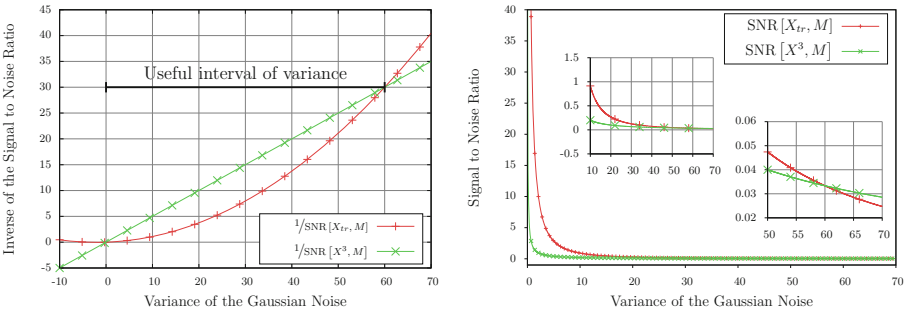
where σ denotes the standard deviation of the Gaussian noise.

As a consequence MVA_{tr} will be better than 2O-CPA in this interval

Theorem 2 gives us the cases where exploiting the second-order leakage will give better results than exploiting the classical leakage of the mask. For example if $n = 8$ (the case of AES) the second-order leakage is better until $\sigma^2 \leqslant 60$.

Figure 2 shows when the SNR of X_{tr} is greater than the SNR of X^3. In order to have a better representation of this interval $1/\text{SNR}$ is also plotted in Fig. 2a.

It is easy to observe that the largest difference in Fig. 2a occurs at $\frac{1}{2}\left(2^{n-2} - \frac{n}{2}\right)$, i.e., in the middle of the *useful interval of variance*.



(a) $1/\text{SNR}$ (b) Signal to noise ratio

Fig. 2. Comparison between the variance of the noise for the classical leakage and the second-order and the impact of these noises on the SNR.

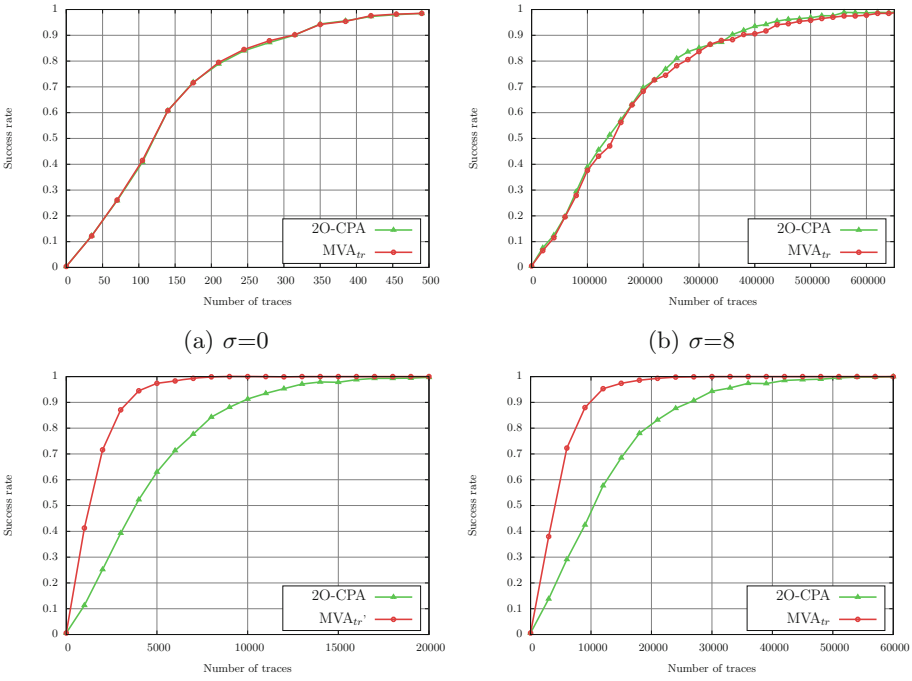
4.4 Simulation Results

In order to validate empirically the results of the Sect. 4, we test the method presented on simulated data. The target is a first order protected AES with table recomputation. To simulate the leakages we assume that each value leaks its Hamming weight with a Gaussian noise of standard deviation σ. The 512 leakages of the table recomputation are those given in Subsect. 4.2.

1000 attacks are realized to compute the success rate of each experiment. In this part, the comparisons are done on the number of traces needed to reach 80 % of success.

It can be seen in Fig. 3a and b that the difference between the two attacks is null for $\sigma = 0$ and $\sigma = 8$. It confirms the bound of the interval shown in Fig. 2. This also confirms that comparing the SNR is equivalent to comparing the SR.

It can be seen in Fig. 3 that in presence of noise the MVA_{tr} outperforms the 2O-CPA. The highest difference between the MVA_{tr} and 2O-CPA is reached when $\sigma = 3$. In this case, the MVA_{tr} needs 2500 traces to mount the attack while the 2O-CPA needs 7500 traces. This represents a gain of 200 %. The gain decreases to 122 % when $\sigma = 4$ Fig. 3d.



(a) σ=0 (b) σ=8

(c) σ=3 (d) σ=4

Fig. 3. Comparison between 2O-CPA and MVA$_{tr}$.

5 An Example on High-Order Countermeasure

The result of the previous section could be extended to any masking scheme based on table recomputation. In particular the MVA$_{tr}$ could be extended to High-Order masking schemes.

5.1 Coron Masking Scheme Attack and Countermeasure

The use of table recomputation could be extended to High-Order masking schemes. An approach has been proposed by Schramm and Paar [22]. However this masking scheme can be defeated by a third order attack [8]. To avoid this vulnerability Coron recently presented [7] a new method based on table recomputation. This method provides a high-order masking (see Algorithm 2). The core idea of this method is to mask each output of the S-Box by different mask and refresh the set of masks between each shift of the table. HV attacks are still a threat against such schemes. Indeed iteratively an attacker will recover each input mask x_i. Afterwards he will be able to perform a first order attack on the AddRoundKey to recover the key. To prevent attacks based on the exploitation of the leakages of the input masks an approach based on the randomization of the

index of the loop is possible. It can be noticed that the entropy needed to build the permutation could be low compare to the entropy needed for the masking scheme.

Algorithm 2. Masked computation of $y = S(x)$

input : $x_1, \ldots, x_d$, such that $x = x_1 \oplus \ldots \oplus x_d$
output: $y_1, \ldots, y_d$, such that $y = y_1 \oplus \ldots \oplus y_d = S(x)$

1 **for** $\omega \in \mathbb{F}_2^n$ **do**
2 $\quad T(\omega) \leftarrow (S(\omega), 0, \ldots, 0) \in (\mathbb{F}_2^n)^d$ // $\oplus(T(\omega)) = S(u)$
3 **end**
4 **for** $i = 1$ *to* $i = d-1$ **do** // $\oplus(T(\varphi(\omega))) = S(\varphi(\omega) \oplus x_1, \ldots, \oplus x_{d-1}) \, \forall \omega \in \mathbb{F}_2^n$
5 $\quad$ **for** $\omega \in \mathbb{F}_2^n$ **do**
6 $\quad\quad$ **for** $j = 1$ **to** d **do**
7 $\quad\quad\quad T'(\varphi(\omega))[j] \leftarrow T(\varphi(\omega) \oplus x_i)[j]$ // $T'(\varphi(\omega)) \leftarrow T(\varphi(\omega) \oplus x_i)$
8 $\quad\quad$ **end**
9 $\quad$ **end**
10 $\quad$ **for** $\omega \in \mathbb{F}_2^n$ **do**
11 $\quad\quad T(\varphi(\omega)) \leftarrow \mathsf{RefreshMasks}(T(\varphi(\omega)))$
$\quad\quad$ // $\oplus(T(\varphi(\omega))) = S(\varphi(\omega) \oplus x_1, \ldots, \oplus x_i)$
12 $\quad$ **end**
13 **end**
14 $(y_1, \ldots, y_d) \leftarrow \mathsf{RefreshMasks}(T(x_n))$ // $\oplus(T(x_d)) = S(x)$
15 **return** $y_1, \ldots, y_n$

5.2 Attack on the Countermeasure

Similarly to the definitions in Subsect. 4.2 let us define the leakages of the table recomputation of the masking scheme of Coron where the order of the masking is $d-1$: $X^{(1)}_{(\omega,i,j)} = \mathsf{HW}[\Phi_i(\omega) \oplus M_i] + N^{(1)}_{(\omega,i,j)} - \frac{n}{2}$ and $X^{(2)}_{(\omega,i,j)} = \mathsf{HW}[\Phi_i(\omega)] + N^{(2)}_{(\omega,i,j)} - \frac{n}{2}$. where $i \in [\![1, d-1]\!]$ will index the $d-1$ masks. The d-th share is the masked sensitive value. And $j \in [\![1, d]\!]$ denotes the index of the loop from lines 6 to lines 9 of the Algorithm 2.

The leakage of the masks is given by $X^{(3)}_i = \mathsf{HW}[M_i] + N^{(3)}_i - \frac{n}{2}$

And let us denote by: $X^\star = \mathsf{HW}[\bigoplus_{i=0}^{d-1}(M_i) \oplus k^\star \oplus T] + N - \frac{n}{2}$ the leakages of the masked value.

Definition 4. *The combination function exploiting the leakage of the table recomputation C_{tr} is given by:*

$$C^d_{cs}: \quad \mathbb{R}^{d\times(d-1)\times 2^{n+1}} \times \mathbb{R} \quad \to \quad \mathbb{R}$$

$$\left(\left(X^{(1)}_{(\omega,i,j)}, X^{(2)}_{(\omega,i,j)}\right)_{\substack{\omega\in\mathbb{F}_2{}^n \\ i\in[\![1,d-1]\!] \\ j\in[\![1,d]\!]}}, X^\star\right) \mapsto \prod_{i=1}^{d-1}\left(\frac{-2}{d2^n}\sum_{\substack{\omega\in\mathbb{F}_2{}^n \\ j\in[\![1,d]\!]}} X^{(1)}_{(\omega,i,j)} \times X^{(2)}_{(\omega,i,j)}\right) \times X^\star.$$

Similarly to Subsection 4.3 we could define:

$$X_{CS_i}(d) = \frac{-2}{d2^n} \sum_{\substack{\omega \in \mathbb{F}_{2^n} \\ j \in [\![1,d]\!]}} X^{(1)}_{(\omega,i,j)} \times X^{(2)}_{(\omega,i,j)}.$$

This value is the combination of all the leaking values of the table recomputation depending of one share. Based on the combination function a multivariate attack can be built.

Definition 5. *The MultiVariate Attack exploiting the leakage of the table recomputation of the $d-1$ order Coron masking Scheme is given by:*

$$MVA^d_{cs}: \begin{array}{ccc} \mathbb{R}^{d\times(d-1)\times 2^{n+1}} \times \mathbb{R} \times \mathbb{R} & \to & \mathbb{F}_2^n \\ \left(\left(X^{(1)}_{(\omega,i,j)}, X^{(2)}_{(\omega,i,j)}\right)_{\substack{\omega\in\mathbb{F}_{2^n} \\ i\in[\![1,d-1]\!] \\ j\in[\![1,d]\!]}}, X^\star, Y\right) & \mapsto & \underset{K\in\mathbb{F}_2^n}{\operatorname{argmax}}\, \rho\left[\prod_{i=1}^{d-1}\left(X_{CS_i}(d)\right) \times X^\star, Y\right], \end{array}$$

where $Y = \mathsf{HW}[T \oplus K] - \frac{n}{2}$

Proposition 3. *MVA_{cs} is sound.*

Remark 3. *The attack presented in Definition 3 is a $d \times (d-1) \times 2^{n+1} + 1$ multivariate $2 \times (d-1) + 1$ order attack.*

HOCPA can be built by combining the d shares using the centered product combination function. In the rest of this article we denote such attacks by "classical" dO-CPA.

$$d\text{O-CPA}: \begin{array}{ccc} \mathbb{R}^{d-1} \times \mathbb{R} \times \mathbb{R} & \longrightarrow & \mathbb{F}_2^n \\ \left(\left(X_i^{(3)}\right)_{i\in[\![1,d-1]\!]}, X^\star, Y\right) & \longmapsto & \underset{K\in\mathbb{F}_2^n}{\operatorname{argmax}}\, \rho\left[X^\star \times \prod_{i=1}^{d-1} X_i^{(3)}, Y\right]. \end{array}$$

5.3 Leakage Analysis

The difference between the two attacks is the use of $X_{CS_i}(d)$ instead of $X_i^{(3)}$ as the leakage of the $d-1$ shares which do not leak the secret key. A.A Ding et al. also provides a formula to compute the SR of HOCPA [10, Sect. 3.4].

Similarly to Sect. 4 the only differences in the formula are the SNR of the shares which do not leak the key. Then by comparing the $\mathrm{SNR}\left[X_{CS_i}(d), M_i\right]$ and $\mathrm{SNR}\left[X_i^{(3)}, M_i\right]$ we compare the success rate of the attacks. It can be noticed that in our model the SNR does not depend on i.

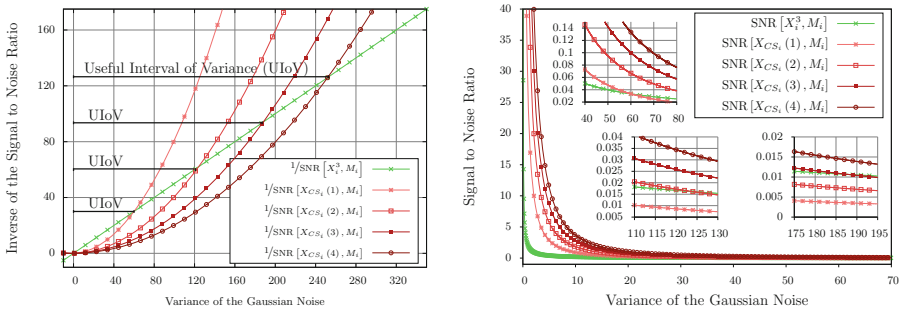
Theorem 4. *The* SNR *of the "second-order leakage" is greater than the* SNR *of the leakage of the mask if and only if*

$$\sigma^2 \leqslant d \times 2^{n-2} - \frac{n}{2}, \tag{11}$$

where σ denotes the standard deviation of the Gaussian noise.

As a consequence MVA_{tr} will be better than 2O-CPA when the noise is in this interval. We can immediately deduce that the size of the Useful Interval of Variance increases linearly with the order of the masking scheme.

Figure 4a and b show the impact of the order d of the attack on the interval of noise where the MVA^d_{CS} outperfoms dO-CPA (let us called this interval the Useful Interval of Variance). We can see that the size of these intervals increases with the order. For example for $d = 3$ the useful interval of variance is $[0, 124]$. It is almost impossible to perform a second order attack with a noise variance of 124.



(a) $1/\text{SNR}$ (b) Signal to noise ratio

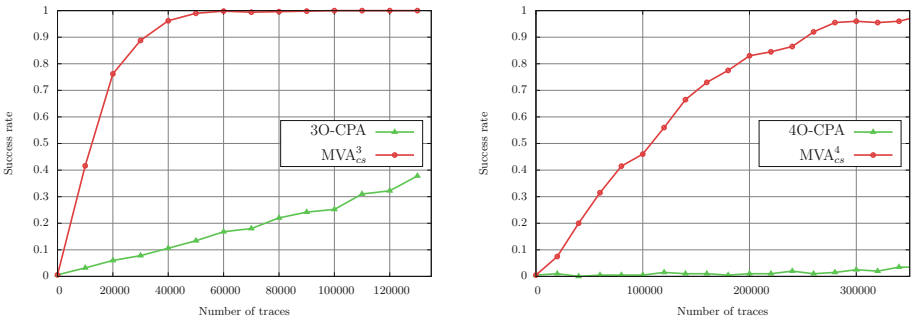
Fig. 4. Comparison between the signal to noise ratio of X_i^3 and signal to noise ratio of $X_{CS}\,(d-1)$ (where d is the order of the attack)

5.4 Simulation Results on Coron Masking Scheme

In order to validate the theoretical results of the Subsect. 5.3 the MVA_{CS} was tested on simulated data and compared to dO-CPA. The simulations have been done with the Hamming weight model and Gaussian noise such as the leakages defined in Subsect. 5.2. We test these attacks against a second and a third order masking scheme.

To compute the success rate the attacks are redone 500 times for the second order masking and 100 times for the third order masking.

In Fig. 5a it can be seen that MVA^3_{cs} reaches 80 % of success rate for less the 20000 traces while the 3O-CPA does not reach 30 % for 100000. In Fig. 5b it can be seen that MVA^4_{cs} reaches 80 % of success rate for less than 200000 traces while the 4O-CPA does not reach 5 %.



(a) σ=3 (b) σ=3

Fig. 5. Comparison between the 4O-CPA and the MVA$^4_{cs}$

6 A Note on Affine Model

In Sect. 4 the leakage function was expected to be the Hamming weight. Let us now study the impact of the leakage function on the MVA$_{tr}$. We suppose that the leakage function is affine.

6.1 Properties of the affine model

A leakage function is said affine if this function is a weighted sum of the bit of the leaking value. As a consequence the leakage function could be rewritten as $\Psi_\alpha(V) = \alpha \cdot V$, where V is the leaking value, α the weight of the leakage of each bit and $\cdot$ the inner product.

Assumption 1. *In order to compare the results in case of an affine model and the Hamming weight model let us assume that the variance is the same in the two cases i.e.* $Var[\mathcal{L}(\alpha, V)] = Var[\mathsf{HW}[V]]$ *this is equivalent to* $\|\alpha\|_2{}^2 = n$.

Let us also assume that all the values manipulated during the algorithm leak in the same way i.e. the weight vector α *of the sum is the same for all the variables* V *of the algorithm.*

Let us redefine the leakage of the table recomputation the (centered) leakage of the random index: $X^{(1)}_\omega = \alpha \cdot (\Phi(\omega) \oplus M) + N^{(1)}_\omega - \frac{1}{2}(\alpha \cdot \mathbf{1})$, the (centered) leakage of the mask random index: $X^{(2)}_\omega = \alpha \cdot (\Phi(\omega)) + N^{(2)}_\omega - \frac{1}{2}(\alpha \cdot \mathbf{1})$. And the (centered) leakage of the mask: $X^{(3)} = \alpha \cdot M - \frac{1}{2}(\alpha \cdot \mathbf{1})$. And let $X^\star$ be the leakage of a sensitive value depending on the key.

6.2 Theoretical Analysis

Similarly to the Subsection 4.3 let us study the impact of the affine model on the success of the MVA$_{tr}$ compared to the 2O-CPA.

As motivated in Sect. 4.1, we can modify the MVA$_{tr}$ in order to target the last round S-Box input: $X^\star = \alpha \cdot \left(\mathtt{Sbox}^{-1}[T \oplus k^\star] \oplus M\right) + N - \frac{1}{2}(\alpha \cdot \mathbf{1})$.

Theorem 5. *The* SNR *of the "second-order leakage" is greater than the* SNR *of the leakage of the mask if and only if*

$$\sigma^2 \leqslant \|\alpha\|_4{}^4 \times \frac{2^{n-2}}{n} - \frac{n}{2},$$

where σ denotes the standard deviation of the Gaussian noise.

As a consequence MVA_{tr} will be better than 2O-CPA when the noise is in this interval.

Corollary 6. *The $\min_{\|\alpha\|_2{}^2=n} \|\alpha\|_4{}^4$ is reached when all the component of α are equal. This means that the worst case for the MVA_{tr} compare to the 2O-CPA is when the leakage is in Hamming Weight.*

6.3 Simulation Results

In order to validate the results of the theoretical study of the previous section some simulations have been done.

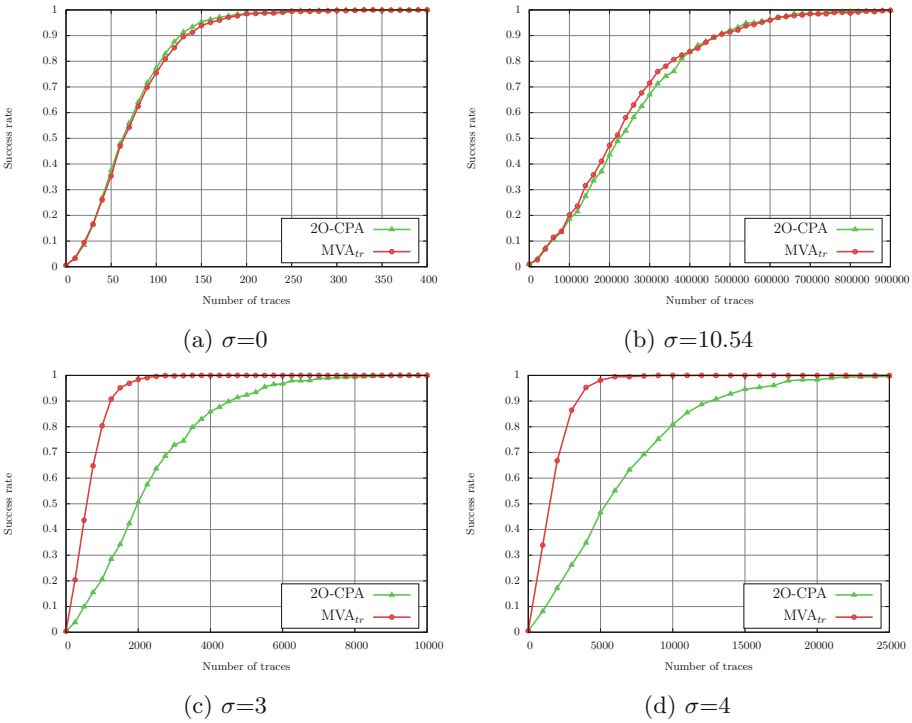
The target considered is the input of the S-Box of the last round; as a consequence we consider $X^\star = \alpha \cdot \left(\texttt{Sbox}^{-1}[T \oplus k^\star] \oplus M\right) + N - \frac{1}{2}(\alpha \cdot \mathbf{1})$.

The mask M and the plain text T are randomly drawn from $\mathbb{F}_2^8$. The noises are drawn from a Gaussian distribution with different variance σ^2. The results of the attacks are expressed using the Success rate. To compute the success rates the experiments have been redone 1000 times. For each experiment the secret key $k^\star$ are randomly drawn over $\mathbb{F}_2^8$. To compare the efficiency of the two attacks we compare the number of traces needed to reach 80 % of success.

For the first experiment let us choice α such as $\alpha_i = \sqrt{\left(1 + (-1)^{i \mod 2} \times \varepsilon\right)}$ with $\varepsilon = 0,9$. In this case $\|\alpha\|_4{}^4 = 14.480$ and following the result of the Theorem 5, the MVA_{tr} should outperform the classical success rate in the interval: $[0, 111]$. It can be seen in Fig. 6a and b that in such case when $\sigma^2 = 0$ or when $\sigma^2 = 111$ the MVA_{tr} and the 2O-CPA need the same number of traces to reach 80 % of success. This confirms first of all the soundness of our model and also that in case of affine model when the target is proceeded in a non linear part of the cryptographic algorithm, the main difference between the two attacks is the SNR. When the standard deviation of the Gaussian noise $\sigma = 3$ the 2O-CPA needs around 3800 traces to reach 80 % of success whereas the MVA_{tr} needs around 1000 traces (Fig. 6c). This represents a gain of 280 %. Compared to the gain observed in case of the Hamming weight model this confirm that the MVA_{tr} performs better compare to the 2O-CPA in case of an affine model. It can be seen in Fig. 6d, when the $\sigma = 4$, the number of traces needed to reach 80 % of success is around 2500 for the MVA_{tr} and around 10000 for the 2O-CPA; this represents a gain of 300 %.

7 Practical Validation

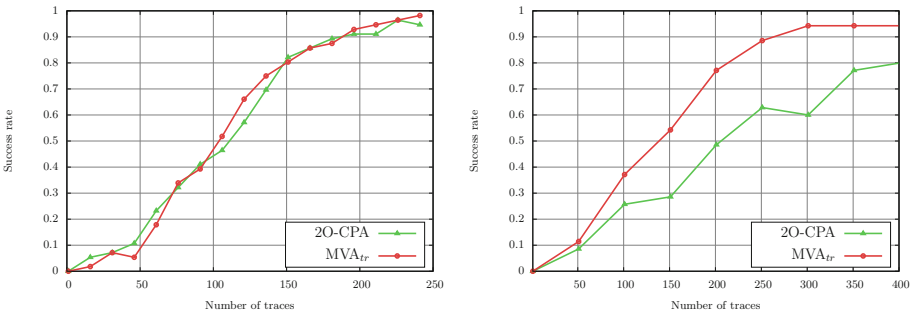
This section presents the results of the multivariate attack exploiting the table recomputation stage on true traces. The traces are electromagnetic leakages of



(a) σ=0 (b) σ=10.54

(c) σ=3 (d) σ=4

Fig. 6. Comparison between 2O-CPA and MVA$_{tr}$ for $\varepsilon = 0.9$.



(a) Comparison on raw traces (b) Comparison with noise addition

Fig. 7. Comparison of the SR of the MVA$_{tr}$ and the 2O-CPA

the execution of an AES with table recomputation executed on an ATMega163 8-bit smartcard which is known to be leaky. To build our experiments 13000 traces have been acquired.

Let us first study the results of the attack. They are expressed using the success rate. The leakage function as been recovered using a linear regression. Both

the MVA_{tr} and the 2O-CPA target: $\texttt{Sbox}[T \oplus k^\star] \oplus M$ as in our implementation the input and output masks are the same.

It can be seen in Fig. 7a that the results of the two attacks are similar. Both attacks perform similarly because the curves are not noisy.

Indeed the average values of the SNR of the 256 leakages of the masked random index $(\Phi(\omega) \oplus M)$ and the SNR of the 256 leakages of the random index $(\Phi(\omega))$ is 5. If we assume that the variance of the signal is equal to two (such as HW on 8 bit CPU) then the variance of the noise is less than 0.5. The mask (M) and the key-dependent share $\texttt{Sbox}[T \oplus k^\star] \oplus M)$ leak with a SNR of 14 which corresponds to a noise variance of 0.1, which is very low (compared to the upper bound of the useful interval of variance given in Theorem 2, namely 60).

This two results are specific to the implementation and a clear disadvantage for the MVA_{tr}. But even in this case the MVA_{tr} works as well as the 2O-CPA, this shows that there is (generally) a gain to use the MVA_{tr}.

In order to confirm these results let us verify that when the noise increases the MVA_{tr} outperforms the 2O-CPA. Let us add a Gaussian noise with a standard deviation of 0.040. Then it can be seen in Fig. 7b that in this case the MVA_{tr} outperforms the 2O-CPA. This confirms that the gain is in the SNR.

8 Conclusions and Perspectives

The table recomputation is a known weakness of masking schemes. We have recalled that practical countermeasures could be built to protect the table recomputation. In this article, we have presented a new multivariate attack exploiting the leakage of the protected table that outperformed classical HODPA even if a large amount of entropy is used to generate the countermeasure. This multivariate attack gives an example of an HOSCA of non-minimal order which is more efficient than the corresponding minimal order HODPA. We have theoretically expressed the bound of noise in which this attack outperforms HOCPA using the SNR. Then we have empirically validated this bound. Moreover, we have shown that the gain to use the multivariate attack grows linearly with the order of the masking schemes. This result highlights the fact that the study of masking scheme should take into account as second parameter the number of variables exploitable by theses attacks. Indeed we have shown in this article that when the number of variables used to perform the attacks increases, the *order* does not alone provide a criterion to evaluate the security of the countermeasure, and that the *SNR* is a better security metric to consider.

In future works we will investigate how to protect table recomputation against such attacks and investigate the cost of such countermeasures, evaluate the threat of such attacks on high-order masking schemes implemented on real component. We will also investigate how multivariate attacks could be applied on other masking schemes and protection techniques. And then, we will quantify the impact of these attacks.

References

1. Akkar, M.-L., Giraud, C.: An implementation of DES and AES, secure against some attacks. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 309–318. Springer, Heidelberg (2001)
2. Blömer, J., Guajardo, J., Krummel, V.: Provably secure masking of AES. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 69–83. Springer, Heidelberg (2004)
3. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)
4. Bruneau, N., Guilley, S., Heuser, A., Rioul, O.: Masks will fall off. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 344–365. Springer, Heidelberg (2014)
5. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, p. 398. Springer, Heidelberg (1999)
6. Clavier, C., Feix, B., Gagnerot, G., Roussellet, M., Verneuil, V.: Improved collision-correlation power analysis on first order protected AES. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 49–62. Springer, Heidelberg (2011)
7. Coron, J.-S.: Higher order masking of look-up tables. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 441–458. Springer, Heidelberg (2014)
8. Coron, J.-S., Prouff, E., Rivain, M.: Side channel cryptanalysis of a higher order masking scheme. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 28–44. Springer, Heidelberg (2007)
9. DeTrano, A., Guilley, S., Guo, X., Karimi, N., Karri, R.: Exploiting small leakages in masks to turn a second-order attack into a first-order attack. In: Proceedings of the Fourth Workshop on Hardware and Architectural Support for Security and Privacy, HASP 2015, pp. 7:1–7:5. ACM, New York (2015)
10. Ding, A.A., Zhang, L., Fei, Y., Luo, P.: A statistical model for higher order DPA on masked devices. In: Batina, L., Robshaw, M. (eds.) CHES 2014. LNCS, vol. 8731, pp. 147–169. Springer, Heidelberg (2014)
11. Fei, Y., Luo, Q., Ding, A.A.: A statistical model for DPA with novel algorithmic confusion analysis. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 233–250. Springer, Heidelberg (2012)
12. Goubin, L., Patarin, J.: DES and differential power analysis the "Duplication" method. In: Koç, Ç.K., Paar, C. (eds.) CHES 1999. LNCS, vol. 1717, pp. 158–172. Springer, Heidelberg (1999)
13. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
14. Maghrebi, H., Prouff, E., Guilley, S., Danger, J.-L.: A first-order leak-free masking countermeasure. Cryptology ePrint Archive, Report 2012/028 (2012). http://dblp.uni-trier.de/rec/bibtex/conf/ctrsa/MaghrebiPGD12
15. Messerges, T.S.: Securing the AES finalists against power analysis attacks. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 150–164. Springer, Heidelberg (2000)
16. Messerges, T.S.: Using second-order power analysis to attack DPA resistant software. In: Paar, C., Koç, Ç.K. (eds.) CHES 2000. LNCS, vol. 1965, pp. 238–251. Springer, Heidelberg (2000)

17. Oswald, E., Mangard, S.: Template attacks on masking—resistance is futile. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 243–256. Springer, Heidelberg (2006)
18. Pan, J., den Hartog, J.I., Lu, J.: You cannot hide behind the mask: power analysis on a provably secure s-box implementation. In: Youm, H.Y., Yung, M. (eds.) WISA 2009. LNCS, vol. 5932, pp. 178–192. Springer, Heidelberg (2009)
19. Prouff, E., Rivain, M.: A generic method for secure sbox implementation. In: Kim, S., Yung, M., Lee, H.-W. (eds.) WISA 2007. LNCS, vol. 4867, pp. 227–244. Springer, Heidelberg (2008)
20. Prouff, E., Rivain, M., Bevan, R.: Statistical analysis of second order differential power analysis. IEEE Trans. Comput. **58**(6), 799–811 (2009)
21. Rivain, M., Prouff, E.: Provably secure higher-order masking of AES. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 413–427. Springer, Heidelberg (2010)
22. Schramm, K., Paar, C.: Higher order masking of the AES. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 208–225. Springer, Heidelberg (2006)
23. Standaert, F.-X., Veyrat-Charvillon, N., Oswald, E., Gierlichs, B., Medwed, M., Kasper, M., Mangard, S.: The world is not enough: another look on second-order DPA. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 112–129. Springer, Heidelberg (2010)
24. Tunstall, M., Whitnall, C., Oswald, E.: Masking tables - an underestimated security risk. IACR Cryptology ePrint Archive 2013: 735 (2013). http://dblp.uni-trier.de/rec/bibtex/conf/fse/TunstallWO13
25. Waddle, J., Wagner, D.: Towards efficient second-order power analysis. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 1–15. Springer, Heidelberg (2004)