Boosting Higher-Order Correlation Attacks by Dimensionality Reduction

Nicolas BRUNEAU[1,2], Jean-Luc DANGER[1,3], Sylvain GUILLEY[1,3],
Annelie HEUSER[⋆1], Yannick TEGLIA[2]

[1] TELECOM-ParisTech, Crypto Group, Paris, FRANCE
[2] STMicroelectronics, AST division, Rousset, FRANCE
[3] Secure-IC S.A.S., Rennes, FRANCE

Abstract. Multi-variate side-channel attacks allow to break higher-order masking protections by combining several leakage samples. But how to optimally extract all the information contained in all possible d-tuples of points? In this article, we introduce preprocessing tools that answer this question. We first show that maximizing the higher-order CPA coefficient is equivalent to finding the maximum of the covariance. We apply this equivalence to the problem of trace dimensionality reduction by linear combination of its samples. Then we establish the link between this problem and the Principal Component Analysis. In a second step we present the optimal solution for the problem of maximizing the covariance. We also theoretically and empirically compare these methods. We finally apply them on real measurements, publicly available under the DPA Contest v4, to evaluate how the proposed techniques improve the second-order CPA (2O-CPA).

Keywords: Bi-variate attacks, second-order correlation power analysis (2O-CPA), principal component analysis, interclass variance, covariance vector

1 Introduction

For more than a decade now Side-Channel Attacks (SCA [6]) have been an important threat against embedded systems. As a consequence protection techniques and countermeasures have been an important research topic. Data masking [9] is one of the most popular protection technique. These schemes have in turn been the target of higher-order SCA [24,18].

In some particular masking implementations, the two shares [16] depending on the same mask leak at different moments (e.g., in software). Second-order attacks that combine two different time samples are called *bi-variate SCA*. When the masking scheme uses d shares, *multi-variate SCA* are still able to reveal the secret key by combining leakage samples corresponding to each of the d shares. Note that, depending on the implementation and the measurement setup each share may leak in multiple samples.

[⋆] Annelie Heuser is a Google European fellow in the field of privacy and is partially founded by this fellowship.

To enhance the results of SCA several preprocessing tools can be used. In the case of *bi-variate SCA* it is particularly interesting to take into account all the information spread over the time. Indeed, the number of possible pairs increases quadratically in the number of leakage samples. For example, if the first share leaks over T_1 samples and the second share over T_2 samples, we could perform a *bi-variate SCA* on $T_1 \times T_2$ possible combined points. So, taking into account all these leaks may undoubtedly increase the efficiency of an attack.

More generally, to break $(d-1)$-order masking schemes the attacker needs to combine d samples corresponding to d shares. So, if T_i is the number of samples which leak the i-th share then the attacker could perform *multi-variate SCA* on $\prod_{1 \leq i \leq d} T_i$ different d-tuples. In other words, the number of possible d-tuples to perform *multi-variate SCA* is in $O(T^d)$ where T is the number of samples each share leaks (and assuming that each share is leaking the same number of samples, i.e., $\forall i \in [\![1, d]\!], T_i = T$).

Many methods have been presented in the area of SCA to combine the information spread over time: the Principal Component Analysis (PCA) for dimensionality reduction [1] for Template attacks [7] but also as a preprocessing tool [2] for DPA [13]. Recently in [11] Hajra and Mukhopadhyay present an approach based on match filters to find the optimal preprocessing. Other methods have been designed to combine samples from different acquisitions ([22,20]). Additionally, PCA has also been used as a distinguisher in [21]. Some other methods could be applied like the Canonical Correlation Analysis [17] to improve CPA [6]. Interestingly, all these methods lead to a dimensionality reduction.

Another approach to improve the efficiency of SCA is to find the optimal model. A *linear-regression* approach may be used. In [17] Oswald and Paar introduce optimization algorithms to determine numerically the optimal linear combination before CPA. By choosing a different objective we can give a formal expression for the result of the optimization problem, and then have an optimal method without any utilization of sophisticated optimization algorithms that would require “parameter settings”, which could be costly in time. Still, we notice that the approach in [17] and our could be advantageously combined.

Our contributions. In this paper we tackle the question *how to optimally combine the information spread over multiple time samples, for HO-CPA attacks of arbitrary order?* We present the optimal preprocessing method and express it as a generic synthetic formula. By linking the PCA to the problem of maximizing the result of the CPA we are able to evaluate the presented method. We compare these two methods theoretically and prove that they are optimal under some assumptions. We then compare these methods empirically as preprocessing tools to boost 2O-CPA attacks on a first-order masking scheme. In particular, we test these methods on real measurements (DPA contest v4 [23]). In summary, we show that taking into account all possible pairs of leakage points will significantly improve the effectiveness of 2O-CPA, in one attack.

Outline of the paper. The rest of the paper is organized as follows. In Sect. 2 we present our case study and a theoretical comparison between PCA and the covariance method as a method to obtain the optimal preprocessing for second-order CPA. The attacks are then applied on a real masked implementation in Sect. 3. Sect. 4 provides another case study to apply these methods as preprocessing tools. Finally, conclusions and perspectives are drawn in Sect. 5.

2 Theoretical optimal preprocessing function

2.1 Case study

Let us assume that each measurement trace can be seen as a vector of points. So the leakage of the measurements can be defined as: $L = (L_t)_{t \in T}$ where $L_t = S_t + N_t$, S_t being the part of the leakage which is linked to the internal operation processed on the target component and N_t being the noise that assumed to be independent of L_T. It can be noted that, we simply refer to interval $[\![1, T]\!]$ as T, whenever there is no risk of confusion. It can also be assumed that these traces are centered and also reduced, i.e., $\mathbb{E}\left[L_t\right] = 0 \ \forall t$ and $\mathsf{Var}\left[L_t\right] = 1 \ \forall t$. Note that, the attacker is always able to center by removing the empirical mean and reduce by dividing the empirical standard deviation.

Let Z be the internal variable (depending on the sensitive variable) manipulated during the algorithm and let f define the leakage model. In the case of CPA, a transformation of the initial data (preprocessing) may increase the correlation coefficient. To consider all information contained in L an option would be to use a linear transformation as a prepossessing. Note that, combining all points by a weighted sum leads to a dimensionality reduction. More precisely,

$$\max_{\alpha} \left|\rho\left[L \cdot \alpha, f(Z)\right]\right|, \tag{1}$$

where ρ is the Pearson coefficient, α is a vector in $\mathbb{R}^T$ and $\cdot$ the scalar product.

Remark 1. The solution of $\max_{\alpha} \left|\rho\left[L \cdot \alpha, f(Z)\right]\right|$ is also a solution of $\max_{\alpha} \rho\left[L \cdot \alpha, f(Z)\right]^2$.

Remark 2 (EIS (Equal Images under the Same key) assumption [19]). The only part of the correlation that allows to distinguish the key is the covariance.

After the preprocessing we do not need to normalize by the variance of the traces, because we compare key guesses between each other for a given time sample not on a direct scale. So, as seen in Remark 2 the normalization by the variance does not impact the way we distinguish the key. Thus, we can simply focus on maximizing the following equation:

$$\max_{\|\alpha\|=1} \mathsf{Cov}\left[L \cdot \alpha, f(Z)\right]^2. \tag{2}$$

As the covariance is not bounded we introduce the constraint $\|\alpha\| = 1$ where $\|\cdot\|$ is the Euclidean norm, namely $\|\alpha\| = \sqrt{\alpha \cdot \alpha}$.

In this section we assume that the attacker has a "learning device" with a fixed key on which he is unrestricted in the number of acquisitions, which is typically more than the required number to successfully perform the attack. As a consequence we can reasonably assume that the attacker knows the key on the learning device and he is able to identify the zones of interest in $[\![1, T]\!]$ where the internal variable leaks. Moreover, he is able to estimate the weights of the linear combination (see Eq. (2)) on the learning device. In the rest of this study we call this step the "learning phase". In the final step the attacker targets another device that is expected to leak in a similar way as the learning one. However, on the device under attack he is no longer able to acquire an unlimited amount of traces. In particular, in this "attack phase" his main goal is to retrieve the secret key using only the minimum number of traces.

2.2 Principal component analysis

A classical way to recombine information with linear combinations is to apply PCA [12]. Let us define X as a set of data that is composed of N vectors of size T. Accordingly, we write X as an $N \times T$ matrix.

Definition 1. *The* PCA *is an orthonormal linear projection of the data, which maximizes the variance of the projected subspace of dimension $T' \leq T$. More formally, we search the projection which maximizes the variance of the projected data. For the first dimension of the subspace this leads to:*

$$\max_{\|u_1\|=1} \mathsf{Var}\left[X u_1\right] = \max_{\|u_1\|=1} {}^t u_1 {}^t X X u_1.$$

For the second dimension, as we want an orthonormal projection, this yields:

$$\max_{\substack{\|u_2\|=1 \\ u_2 \cdot u_1 = 0}} {}^t u_2 {}^t X X u_2.$$

The process is iterated for each dimension $T' \leq T$.

Remark 3. In general, most of the variance lays within a few dimensions (i.e., much less than T).

Proposition 1. *The solution of the problem in Def. 1 is the T' eigenvectors of X associated to the T' maximal eigenvalues.*

Proof. The proof can be found in [12]. □

As the problem of maximizing the covariance depends on the expected leakage model the preprocessing is defined such that it takes f into account. This implies that the given preprocessing methods are model-dependent. We can explicit the Proposition 1:

Proposition 2. *If we link our measurements L to their conditional expectations $\mathbb{E}\left[L|f(Z)\right]$ knowing a model $f(Z)$, then the PCA yields the principal direction:*

$$\max_{\|\alpha\|=1} \mathsf{Var}\left[\mathbb{E}\left[L|f(Z)\right]\cdot\alpha\right].$$

This result means that the eigenvector of largest eigenvalue is the projection that maximizes the inter-class variance.

Proof. Let $f_1, f_2, \ldots, f_N$ the values that $f(Z)$ can take. Then, the lines of matrix X are $\mathbb{E}\left[L|f(Z)=f_1\right]$, $\mathbb{E}\left[L|f(Z)=f_2\right]$, $\ldots$, $\mathbb{E}\left[L|f(Z)=f_N\right]$. Apply Proposition 1. □

2.3 Preprocessing on modulated side channel traces

Definition 2. *Let us now define a* modulated trace *as a trace in which each time sample can be expressed as a modulation of a model (static in time) plus an independent noisy part:*

$$L = \left(\beta_t f(Z) + N_t\right)_{t\in T} = f(Z)\beta + \left(N_t\right)_{t\in T}\,, \tag{3}$$

where β is a vector in $\mathbb{R}^T$ and each N_t is drawn from an independent identical distribution $\mathcal{N}$. In specific, the variance of the noise does not depend on the time sample $t\in T$.

This notion is illustrated in Fig. 1. In this figure, the pink curve represents the *deterministic part* of the signal, which is a damped oscillation due to the mismatch between the sensor and the leakage source. The pinkish area extending over the pink line represents the *noise noise*. This figure 1 clearly shows that there are many samples which can be advantageously combined, in a view to increase the signal-to-noise ratio.

Theorem 1. *In the case of modulated traces the solution of PCA is equivalent to maximizing the covariance (Eqn. (2)). More precisely, if $L = \left(\beta_t f(Z) + N_t\right)_{t\in T}$ then*

$$\alpha \in \operatorname*{argmax}_{\|\alpha\|=1} \mathsf{Cov}\left[L\cdot\alpha, f(Z)\right]^2 \iff \alpha \in \operatorname*{argmax}_{\|\alpha\|=1} \mathsf{Var}\left[\mathbb{E}\left[L|f(Z)\right]\cdot\alpha\right].$$

Proof. The proof is given in Appendix A. □

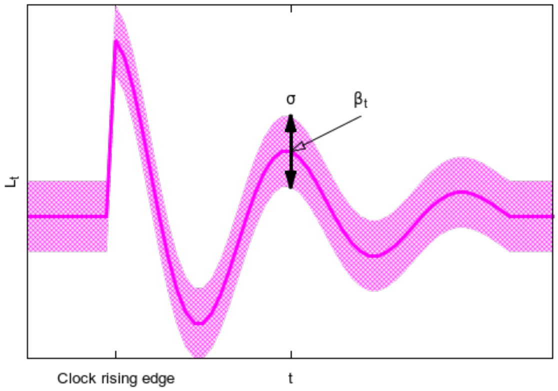


Fig. 1: Example of a modulated trace

Notice that, in Theorem 1, we consider that many vectors α can maximize the covariance: so, the return value of the argmax operator is a set.

In a particular case of Theorem 1 we can explicitly describe α.

Lemma 1. *If α and β are linearly dependent, we have:*

$$\frac{\beta}{\|\beta\|} \in \underset{\|\alpha\|=1}{\operatorname{argmax}} \mathsf{Cov}\left[L \cdot \alpha, f(Z)\right]^2 \quad . \tag{4}$$

Proof. The proof is given in Appendix B. □

After projection into the new reduced space the covariance matrix will be zero everywhere except at $(0, 0)$. Moreover, all the variance should be contained in the first principal direction, thus, we do not need to take the other eigenvectors into consideration.

As β does not depend on a particular model we also maximize the covariance for wrong keys in the same proportion as the covariance for the good key. Thus we do not change the way we distinguish the good key from the wrong ones (the relative distinguishing margin is unchanged [25]). However, the dimensionality reduction leads to an improvement of the attack by increasing the signal-to-noise ratio (SNR). We define the SNR as the variance of the signal divided by the variance of the noise. This definition of SNR coincides with the Normalized Inter-Class Variance (NICV [5,4]).

Lemma 2. *If the noise N_t is identically distributed (i.d.) for all t, then the noise is unchanged by any linear combination of unitary norm.*

Proof. By hypothesis, $\mathsf{Var}\left[(N_t)_{t\in T} \cdot \alpha\right] = \|\alpha\|^2 \mathsf{Var}\left[\mathcal{N}\right] = \mathsf{Var}\left[\mathcal{N}\right]$. □

Proposition 3. *If the noise N_t is i.d. for all t, then the signal-to-noise ratio is maximum after the projection:*

$$\frac{\max_{t\in T} \mathsf{Var}\left[\beta_t f(Z)\right]}{\mathsf{Var}\left[\mathcal{N}\right]} \leq \frac{\max_{\|\alpha\|=1} \mathsf{Var}\left[\mathbb{E}\left[L|f(Z)\right]\cdot\alpha\right]}{\mathsf{Var}\left[\mathcal{N}\right]}.$$

Proof. By definition of α we have $\max_{t\in T} \mathsf{Var}\left[\beta_t f(Z)\right] \leq \max_{\|\alpha\|=1} \mathsf{Var}\left[\mathbb{E}\left[L|f(Z)\right]\cdot\alpha\right]$. Besides, by lemma 2, the numerator of the SNR does not depend on our preprocessing, since is satisfies $\|\alpha\| = 1$. □

Remark 4. In the case of modulated traces the PCA gives the solution of a matched-filter [14].

2.4 Covariance vector as a preprocessing method

In the general case when the model is not known or in the presence of noise, the variance may not only be contained in the first eigenvector [2]. Therefore, it may be useful to also take the other directions of the PCA into account. Note that, we still obtain an optimal function to reduce the dimensionality before conducting a CPA under the same leakage model assumption.

Proposition 4 (Covariance method).

$$\left(\frac{\mathsf{Cov}\left[L_t; f(Z)\right]}{\|(\mathsf{Cov}\left[L_t; f(Z)\right])_{t\in T}\|}\right)_{t\in T} \in \operatorname*{argmax}_{\|\alpha\|=1} \mathsf{Cov}\left[L\cdot\alpha, f(Z)\right]^2$$

Proof. The proof is given in Appendix C. □

So, the normalized covariance is the optimal preprocessing method to maximize the value of the covariance when using linear combinations of traces points. In the rest of this study we call this method the "covariance method" and the result the "covariance vector".

Remark 5. Note that, the model of the actual leakage of the traces is not used in the proof of Appendix C. The results are therefore applicable for any leakage model such as the one presented in [10].

2.5 Discussion

The previous subsection shows that the projection of the differential traces on the covariance vector gives a solution to the problem of maximizing the covariance after dimensionality reduction (i.e., after having learned the best linear form). This method is better than the state-of-the-art, where each tuple of samples is processed on its own (see the *big picture* in Fig. 2); it can be seen as a generalization to higher-order attacks of [11]. Some other preprocessing tools have been proposed to reduce the dimensionality and enhance the quality of the CPA. The PCA [2] is a known way to preprocess the data to reduce the dimension and increase the efficiency of attacks. As defined in Sect. 2.3, PCA is directly linked to the maximization problem, which is also underlined by our empirical results given in Sect. 3.

Oswald and Paar showed in [17] that the best linear combination ("best" in the sense of separating the highest peaks from the nearest rival) can be approached by numerical resolutions. The model presented in [11] is not totally applicable to our study case. If we are in the case of modulated traces, the expectation over each sample of the combined traces could be null. In this case the method presented is not directly suitable.

The point of this study is not to exhibit a better method for dimensionality reduction but to show that we can solve this problem in an easier way by using the vector of covariance.

Other preprocessing methods can be used before any dimensionality reduction such as reduction filtering using a Fourier or a Hartley transform [3]. However, when the transformation is linear and invertible, the covariance method applies in a strictly equivalent way. The next subsection clarifies this point on the example of the Fourier transform.

2.6 Time vs Frequency domains

Let L a signal in time domain, i.e., $L = (L_t)_{t \in T}$. The representation of L in the frequency domain is the discrete Fourier transform $\mathcal{F}(L)$.

Definition 3 (Discrete Fourier transform). *Let $\imath$ be a square root of -1 in $\mathbb{C}$. The discrete Fourier transform of L is a vector $\mathcal{F}(L)$ of same length, defined as $\mathcal{F}(L)_f = \sum_{t \in [\![1,T]\!]} L_t \cdot e^{-2\pi \imath f t/T}$, for all f in the interval $[\![1,F]\!]$ (where $F = T$).*

Proposition 4 can also be applied on $\mathcal{F}(L)$ instead of L. We then have the following Corollary.

Corollary 1 (Covariance method in the frequency domain). *The covariance method in frequency domain yields covariance vectors equal to the Fourier transform of the covariance vectors in the time domain.*

Proof. We have $\mathcal{F}(L) \cdot \alpha = L \cdot \mathcal{F}(\alpha)$, by interversion of the sums on f and t. Besides, Parseval's theorem states that $||\mathcal{F}(\alpha)||^2 = ||\alpha||^2$. Thus, the application of Proposition 4 on $\mathcal{F}(L)$ instead of L yields $\mathcal{F}(\alpha)$, where α are the covariance vectors obtained in the time domain. □

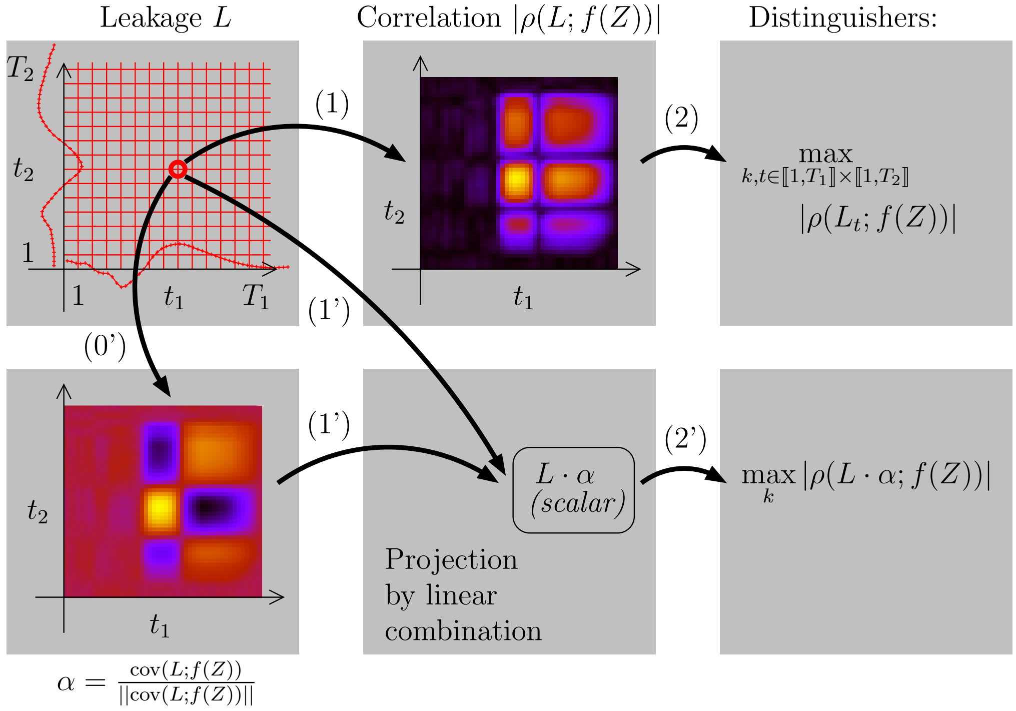

Leakage L
Correlation $|\rho(L; f(Z))|$
Distinguishers:
T_2
t_2
1
1
t_1
T_1
(1)
(2)
$\max_{k,t\in[\![1,T_1]\!]\times[\![1,T_2]\!]} |\rho(L_t; f(Z))|$
(1')
(0')
(1')
(2')
$L \cdot \alpha$
(scalar)
$\max_k |\rho(L \cdot \alpha; f(Z))|$
Projection
by linear
combination
$\alpha = \frac{\mathrm{cov}(L;f(Z))}{||\mathrm{cov}(L;f(Z))||}$

Fig. 2: Big picture of the "covariance method". The usual 2O-CPA computes a correlation for each pair (t_1, t_2) of leakage (step (1)), and then searches for a maximum over the keys and the time instances (step (2)). Our new method obtains a "covariance vector" (termed α) on a "learning device" (step (0')), and then first projects the leakage L on α (step (1')), before looking for the best key only while maximizing the distinguisher (step (2')). Notice that the model $f(Z)$ depends implicitly on the key guess k.

3 Empirical results

In Sect. 2 we defined two preprocessing methods (the PCA and the "covariance method"). They were described in general, but can also apply to second-order CPA; the only difference is that the interval $[\![1, T]\!]$ where samples live is replaced by the Cartesian product $[\![1, T_1]\!] \times [\![1, T_2]\!]$, where T_1 and T_2 are the window lengths containing the leakage of the two shares. Accordingly, the leakage L is the suitable combination (e.g., the centered product [18]) of samples from each window, which is reflected in the model (See for instance Eqn. (5) and (6)). We will now compare these two methods on real measurements. These methods combine in one point the information spread over several points. The more samples to combine, the more the dimensionality reduction increases the success of the attacks.

3.1 Implementation of the masking scheme

To evaluate these two methods we use the publicly available traces of the DPA contest v4 [23], which uses a first order low-entropy masking protection applied on AES called Rotating S-box Masking (RSM). In RSM only sixteen Substitution boxes (S-boxes) are used and all the sixteen outputs of SubBytes are masked by a different mask. We take great in this paper to exploit second-order leakage (in particular, we avoid the first-order leakage identified by Moradi et al. [15]). Moreover, the same mask is used for the AddRoundKey operation where it is XORed to one plaintext byte P and in the SubBytes operation where it is XORed with the S-box output depending on another plaintext byte P'. As a consequence a bi-variate CPA can be built by combining these two leaks knowing P and P'. The leakage model in this case is given by:

$$f(Z) = \mathbb{E}\left[(w_H(P \oplus M) - 4) \cdot \big(w_H(\mathtt{Sbox}[P' \oplus K] \oplus M) - 4\big)\, | P, P', K\right], \tag{5}$$

where P, P', K are two bytes of the plaintext and a byte of the key respectively, together noted $Z = (P, P', K)$, and where $w_H(\,\cdot\,)$ is the Hamming weight function and the expectation is taken over K. We denote this combination as (XOR, S-Boxes).

Moreover, we also define another way to combine points in order to compensate the mask. As only sixteen different masks in RSM are used, also a link between the masks used at the output of the S-boxes exists. Accordingly, the combination of the outputs of two different S-boxes are not well balanced and we could consider an attack depending on two different S-Boxes which use two different masks. In this case the leakage model for the bi-variate CPA is:

$$f(Z) = \mathbb{E}\left[(w_H(\mathtt{Sbox}[P \oplus K] \oplus M) - 4) \cdot \big(w_H(\mathtt{Sbox}[P' \oplus K'] \oplus M') - 4\big)\, | P, P', K, K'\right]. \tag{6}$$

In this equation, which we denote as (S-Boxes, S-Boxes), P and K (resp. P' and K') are the plaintext and key bytes entering the first (resp. the second) S-Box, and Z is a shortcut for the quadruple (P, P', K, K'). We notice that there exists a deterministic link between M and M'; M and M' belong to some subset $\{m_0, m_1, \ldots, m_{15}\}$ of $\mathbb{F}_2^8$. We assume that M enters S-box $0 \leq i \leq 15$ and M' S-box $0 \leq i' \leq 15$. Then when $M = m_{\mathsf{offset}}$ for some $0 \leq \mathsf{offset} \leq 15$, we have that $M' = m_{\mathsf{offset}+i'-i \mod 16}$.

3.2 Leakage analysis

We assume that the adversary is able to identify the area where the two operations leak during the "learning phase". In order to analyze the leakage of the two operations, we first calculate the covariance of the traces when the mask is known using 25000 measurements.

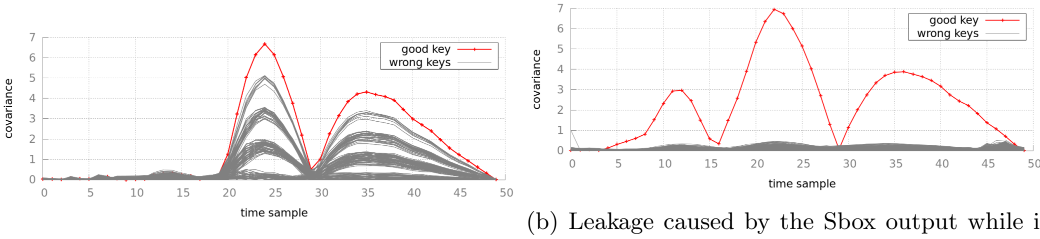
Figure 3a presents the absolute value of the covariance between the points where the mask is XORed with the plaintext and the leakage model $w_H(P \oplus M \oplus K) - 4$.

The covariance is computed for all key guesses K, where the wrong keys are plotted in grey and the correct key in red. Note that, as we target an XOR operation the maximum of the absolute value of the covariance is reached for two key guesses, namely the correct one and its opposite. It is quite clear, in Fig. 3a, that the traces are reasonably modulated (as per Def. 2); consequently, the relative distinguishing margin is constant over all the whole trace (as underlined in Sec. 2.3). In the sequel, we use as leakage for the first share $w_H(P \oplus M) - 4$ instead of $w_H(P \oplus M \oplus K) - 4$. As the second share is key-dependent, this choice allows us to restrict ourselves to one key search instead of two.

Figure 3b presents the covariance between the points where the output of an S-box leaks and the leakage model $w_H(\mathtt{Sbox}[P' \oplus K] \oplus M) - 4$.

In both cases the mask leaks over several points; 50 samples represent less than 1 clock cycle. In this case the leakage is not uniformly spread over the points, thus it is reasonable to use a weighted sum to reduce the dimensionality of the data.

As the two leakages do not depend on the same operations their shapes are different. Interestingly, the distance between the correct key (red) and the next rival (grey) is much smaller in Figure 3a than in Figure 3b, Indeed the covariance plotted in Figure 3a is computed using a leakage depending on AddRoundKey, whereas the covariance plotted in Figure 3b is computed using a leakage caused by SubBytes. More precisely, the second plot corresponds to a time window when the value of the S-box output is moved during the ShiftRows operation that follows SubBytes.

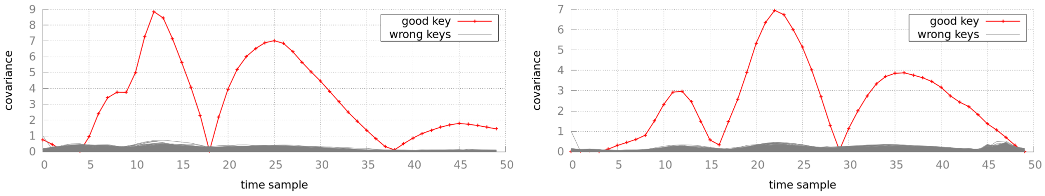


(a) Leakage caused by AddRoundKey

(b) Leakage caused by the Sbox output while in ShiftRows

Fig. 3: Covariance absolute value, for (a) XOR and (b) S-box

Figure 4a (resp. 4b) presents the covariance between the points where the output of an S-box leaks and the leakage model $w_H(\mathtt{Sbox}[P \oplus K] \oplus M) - 4$ (resp. $w_H(\mathtt{Sbox}[P' \oplus K'] \oplus M') - 4$). It can be noticed that the leakages of two different S-boxes indeed differ. The reason of this difference is that the two leakages are not due to the execution of the same operations. Figure 4b shows the covariance between the leakage of the S-box output due to the ShiftRows operation that follows and the corresponding model, whereas Figure 4a shows the covariance between the leakage due to the SubBytes operations and the corresponding model. As *looking-up* and *moving* a byte are different operations, they leak differently.



(a) Leakage caused by SubBytes

(b) Leakage caused by ShiftRows

Fig. 4: Covariance absolute value, for (a) S-box and (b) S-box+ShiftRows

3.3 Experimental protocol

In this experiment we select two windows of 50 points corresponding to the leakage of the two shares. Then all possible pairs of points have been combined using the centered product function [18]. In all the experiments, the preprocessing method and the 2O-CPA are applied on these “combined” traces. We compare 2O-CPA with and without preprocessing.

We used the 50000 first traces of the DPA contest v4 for the learning phase and the remaining for the attack phase. To compute the success rate we repeated the experiment as many times as we could due to the restricted amount of traces.

Note that, several attacks using profiling or semi-profiling have been published in the Hall of Fame of the DPA contest v4. Most of these attacks are specially adapted to the vulnerabilities of the provided implementation or the particularities of RSM. However, our proposed preprocessing tools do not particularly target RSM, moreover, they are generic and could be applied to any masking scheme leaking two shares.

3.4 Comparison of the two preprocessing methods and classical second-order CPA

First of all, for the (XOR, S-Boxes) combination we see in Fig. 5 that the preprocessing improves the efficiency of the attacks. We need less than 200 measurements to reach 80% of success with the covariance or PCA preprocessing while we need more than 275 measurements for the classical 2O-CPA, which gives an improvement of 30%.

Figure 6 shows a 3-D representations of the vectors using the PCA (which returns the first eigenvector) and the covariance method (which returns the covariance

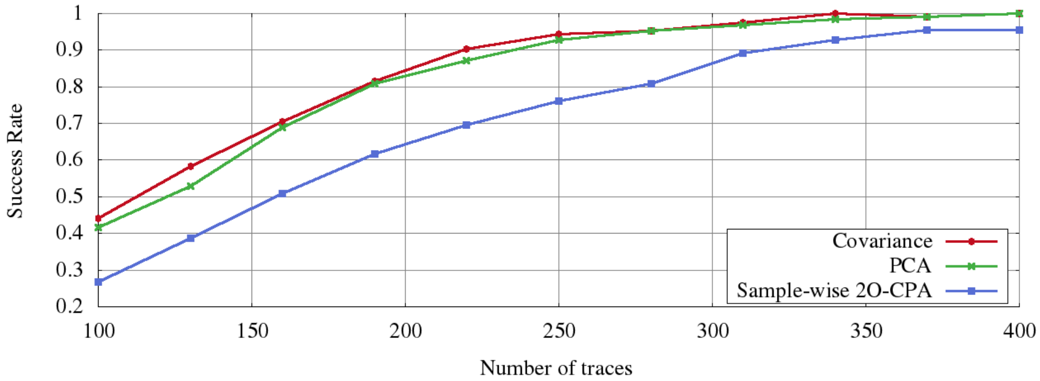


Fig. 5: Comparison between the classical second-order CPA and second-order CPA with preprocessing using (XOR, S-Boxes)

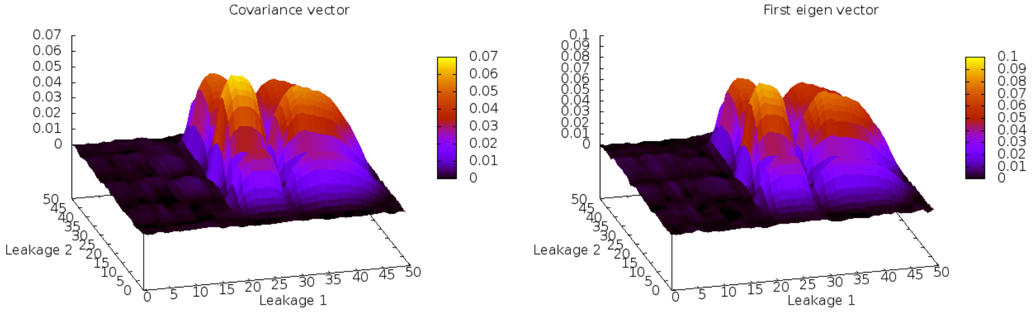


Fig. 6: Comparison between covariance vector and the first eigenvector

vector). The larger the value on the z-axis of Fig. 6 and 8, the higher the contribution (weight) of this point. The axes "leakage 1" and "leakage 2" represent the part depending on the two leakages of XOR (Fig. 3a) and S-box (Fig. 3b) operations in the combined traces. We can see in Figure 6 that the two methods highlight the same points of the combined traces and have the same shape (approximately the same values). Thus, the two methods give similar results in terms of success rate, which is confirmed by Figure 5.

As can be seen in Figure 7, in case of the (S-Boxes, S-Boxes) combination we need around 275 traces to reach 80% of success for the 2O-CPA after the two preprocessing methods, while the raw 2O-CPA needs around 550 traces to succeed. So, using the preprocessing method decreases the number traces to perform the attack by 50%. It can be seen that the two methods yield apparently exactly the same results, which means that we are precisely in the framework of Theorem 1: the display traces that are almost perfectly modulated by one static leakage model.

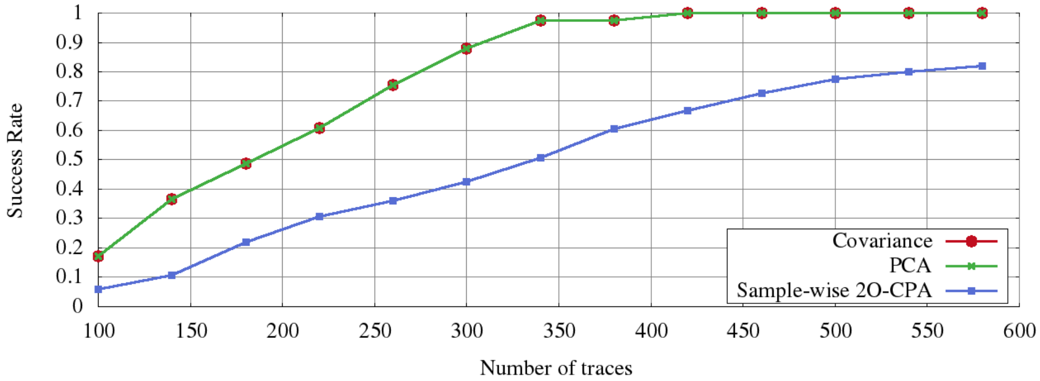


Fig. 7: Comparison between the classical second-order CPA and second-order CPA with preprocessing using (S-boxes, S-Boxes)

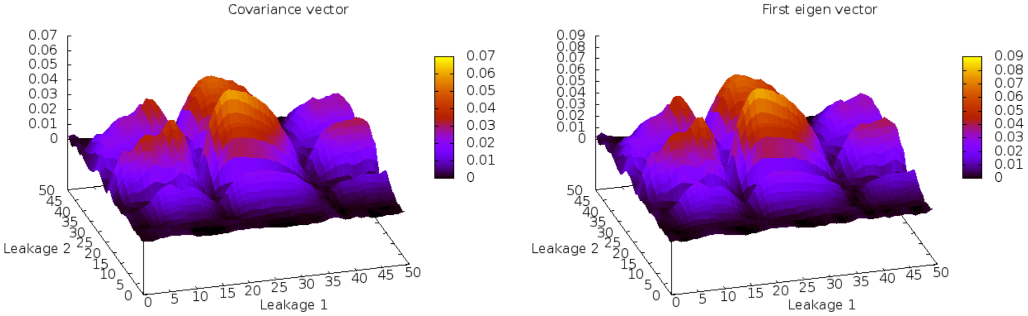
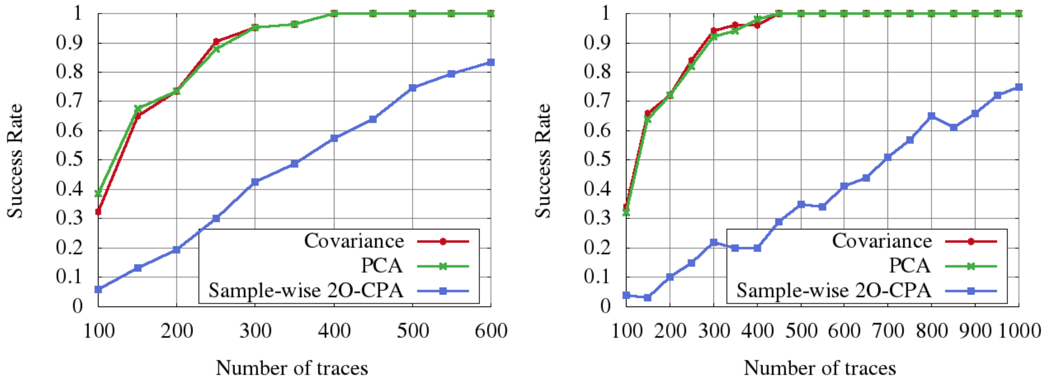


Fig. 8: Comparison between the covariance vector and the first eigenvector

One explanation of the effectiveness of the preprocessing can be found in Figure 8. There are much more leaking points in the same window size when we combine two S-boxes. It can be seen in Sect. 3.5 that another explanation can be the fact that when we apply these preprocessing methods the attacks are less sensitive to the noise.



(a) Standard deviation of 3

(b) Standard deviation of 5

Fig. 9: Comparison between 2O-CPA with preprocessing method and without in presence of Gaussian noise, with a standard deviation of 3 for (a) with a standard deviation of 5 for (b)

3.5 How is the preprocessing linked to the noise?

We have theoretically shown in Proposition 3 that the presented preprocessing methods improve the SNR. We now empirically verify this results. In each point we add Gaussian noise to mimic real noisy measurements. We perform this experiment on the same points and with the same model as used for Figure 5.

Figure 9a shows that using preprocessing methods improves second-order CPA in presence of noise. In this case we added Gaussian noise with a standard deviation of 3. The attacks after preprocessing need around 225 measurements to reach 80% of success whereas the 2O-CPA needs more than 550 measurements. Thus, preprocessing leads to a gain over 50%. As shown in Figure 5 the gain was close to 30% without noise.

In Figure 9b we can see that for Gaussian noise with a standard deviation of 5 the gain is more than 75%. Indeed the 2O-CPA after preprocessing needs around 250 traces reach 80% of success rate whereas for 2O-CPA 1000 measurements are not sufficient.

So this kind of preprocessing by dimensionality reduction is well designed against noisy implementation where the noise is not correlated with the time or the data.

4 On the fly preprocessing

We have defined a case study when the attacker owns a “learning device”. As a consequence he is able to acquire a sufficient number of measurements to well estimate the covariance matrix for the PCA and the covariance vectors. However, the attacker might not always have this powerful tool.

As seen in Subsect. 3.4 even for a small number of traces for the learning phase we have a significant improvement when we use preprocessing methods. We therefore evaluate these tools also as "on the fly" preprocessing methods.

4.1 Case study

We now model a less powerful attacker who does not have a "learning device" and estimates the value of the coefficient of the linear transformation directly on the traces used for the attack. Because the key is unknown the preprocessing method has to be computed for each key hypothesis. Finally, the adversary applies the covariance between the transformed data and the model depending on the key hypothesis. In this experiment we use the 10000 first traces of the DPA contest to compute the success rate which results in 25 repetitions.

4.2 Empirical results

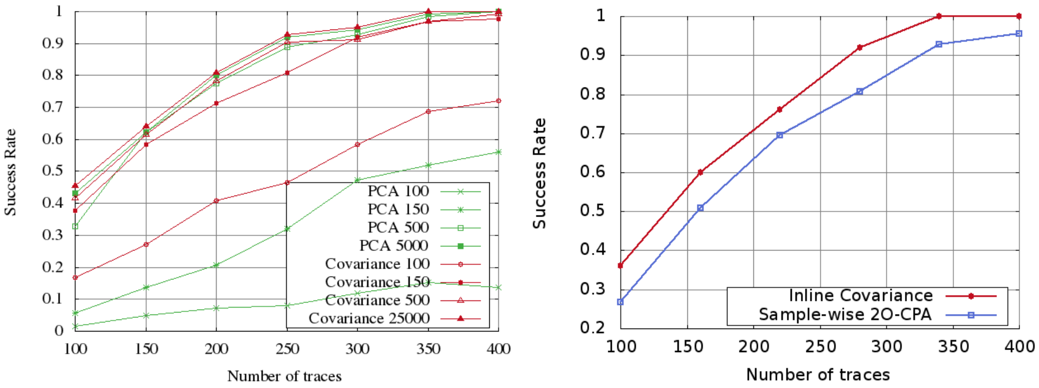
Figure 10a illustrates the success rate after preprocessing for different sizes of the learning set for PCA (green) and the covariance vector (red). One can observe that the covariance method performs better than PCA when a low number of traces is used during the learning phase, accordingly, this method is a good choice as a "on the fly" preprocessing method. The reason why the PCA method needs more measurements for the learning than the covariance method to reach its maximum efficiency during the attack phase could be the fact that the covariance matrix (see the term ${}^tXX$ in Def. 1) needs more traces to be well estimated.

Figure 10b shows that with the "on the fly" preprocessing we can perform 2O-CPA using 225 measurements. This represents a gain of 18% compared to raw (sample-wise) 2O-CPA.

5 Conclusions and Perspectives

In this article we presented the covariance method as an optimal preprocessing method for second-order CPA. By using all possible leakage points our method improves the efficiency of the attacks and as the number of combined leakage points grow quadratically, thus our preprocessing method is well adapted for *bi-variate CPA*. We further theoretically linked the PCA to the problem of maximization of the covariance. We demonstrated theoretically the result of the covariance method to be the optimal linear combination for maximizing the covariance and underlined empirically that this method improves the result of *bi-variate CPA*.

Compared to 2O-CPA, the attack based on the optimal covariance method is significantly improved, particularly in presence of noise and when the number of



(a) Comparison between covariance and PCA depending on the size of the learning base

(b) Comparison between covariance in line preprocessing and 2O-CPA

Fig. 10: Evaluation of inline preprocessing methods

leaking points is important. This is partly explained by the fact the optimal covariance considers all the relevant sampling points, whereas the 2O-CPA considers only the best pair of samples and does not exploit the other interesting pairs.

We have also shown that the optimal covariance method is more efficient than PCA when the learning phase is performed on the fly. All the results have been validated by experiences on real traces corresponding to masking implementation of the DPA contest v4. As a consequence dimensionality reduction by linear combination is well adapted to the case of *multi-variate CPA*. Moreover, the higher the order of masking, the more efficient the attack after preprocessing.

In our future work we will extend the previous results on other implementations which are less favorable to attacker, e.g., with more noise. Also we plan to compare the method presented in this article and the method presented in [11] in these cases. We will additionally apply these methods on different masking scheme especially on higher-order masking schemes.

References

1. Cédric Archambeau, Éric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template Attacks in Principal Subspaces. In *CHES*, volume 4249 of *LNCS*, pages 1–14. Springer, October 10-13 2006. Yokohama, Japan.
2. Lejla Batina, Jip Hogenboom, and Jasper G. J. van Woudenberg. Getting more from pca: First results of using principal component analysis for extensive power analysis. In Dunkelman [8], pages 383–397.
3. Pierre Belgarric, Shivam Bhasin, Nicolas Bruneau, Jean-Luc Danger, Nicolas Debande, Sylvain Guilley, Annelie Heuser, Zakaria Najm, and Olivier Rioul. Time-Frequency Analysis for Second-Order Attacks. In Aurélien Francillon and Pankaj Rohatgi, editors, *CARDIS*, volume 8419 of *Lecture Notes in Computer Science*, pages 108–122. Springer, 2013.

4. Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, and Zakaria Najm. NICV: Normalized Inter-Class Variance for Detection of Side-Channel Leakage. In *International Symposium on Electromagnetic Compatibility" (EMC '14 / Tokyo)*. IEEE, May 12-16 2014. Session OS09: EM Information Leakage. Hitotsubashi Hall (National Center of Sciences), Chiyoda, Tokyo, Japan.
5. Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, and Zakaria Najm. Side-channel Leakage and Trace Compression Using Normalized Inter-class Variance. In *Proceedings of the Third Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '14, pages 7:1–7:9, New York, NY, USA, 2014. ACM.
6. Éric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *CHES*, volume 3156 of *LNCS*, pages 16–29. Springer, August 11–13 2004. Cambridge, MA, USA.
7. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In *CHES*, volume 2523 of *LNCS*, pages 13–28. Springer, August 2002. San Francisco Bay (Redwood City), USA.
8. Orr Dunkelman, editor. *Topics in Cryptology - CT-RSA 2012 - The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27 - March 2, 2012. Proceedings*, volume 7178 of *Lecture Notes in Computer Science*. Springer, 2012.
9. Louis Goubin and Jacques Patarin. DES and Differential Power Analysis. The "Duplication" Method. In *CHES*, LNCS, pages 158–172. Springer, Aug 1999. Worcester, MA, USA.
10. Suvadeep Hajra and Debdeep Mukhopadhyay. Pushing the limit of non-profiling dpa using multivariate leakage model. *IACR Cryptology ePrint Archive*, 2013:849, 2013.
11. Suvadeep Hajra and Debdeep Mukhopadhyay. On the Optimal Pre-processing for Non-profiling Differential Power Analysis. In *COSADE*, Lecture Notes in Computer Science. Springer, April 14-15 2014. Paris, France.
12. Ian T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics, 2002. ISBN: 0387954422.
13. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Proceedings of CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer-Verlag, 1999.
14. Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX — Smartcard'99*, pages 151–162, May 10–11 1999. Chicago, Illinois, USA.
15. Amir Moradi, Sylvain Guilley, and Annelie Heuser. Detecting Hidden Leakages. In Ioana Boureanu, Philippe Owesarski, and Serge Vaudenay, editors, *ACNS*, volume 8479. Springer, June 10-13 2014. 12th International Conference on Applied Cryptography and Network Security, Lausanne, Switzerland.
16. 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.
17. David Oswald and Christof Paar. Improving side-channel analysis with optimal linear transforms. In Stefan Mangard, editor, *CARDIS*, volume 7771 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2012.
18. Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.
19. Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In LNCS, editor, *CHES*, volume 3659 of *LNCS*, pages 30–46. Springer, Sept 2005. Edinburgh, Scotland, UK.
20. Youssef Souissi, Shivam Bhasin, Sylvain Guilley, Maxime Nassar, and Jean-Luc Danger. Towards Different Flavors of Combined Side Channel Attacks. In Dunkelman [8], pages 245–259.
21. Youssef Souissi, Maxime Nassar, Sylvain Guilley, Jean-Luc Danger, and Florent Flament. First Principal Components Analysis: A New Side Channel Distinguisher. In Kyung Hyune Rhee and DaeHun Nyang, editors, *ICISC*, volume 6829 of *Lecture Notes in Computer Science*, pages 407–419. Springer, 2010.
22. François-Xavier Standaert and Cédric Archambeau. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In *CHES*, volume

5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, August 10–13 2008. Washington, D.C., USA.

23. TELECOM ParisTech SEN research group. DPA Contest (4th edition), 2013–2014. http://www.DPAcontest.org/v4/.
24. 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.
25. Carolyn Whitnall and Elisabeth Oswald. A Fair Evaluation Framework for Comparing Side-Channel Distinguishers. *J. Cryptographic Engineering*, 1(2):145–160, 2011.

A Proof of Theorem 1

Proof. On the one side we have

$$\begin{aligned}\mathsf{Cov}\left[L\cdot\alpha, f(Z)\right] &= \left(\mathsf{Cov}\left[S_t+\mathcal{N}_t, f(Z)\right]\right)_{t\in T}\cdot\alpha\\ &= \left(\mathsf{Cov}\left[S_t, f(Z)\right]\right)_{t\in T}\cdot\alpha\\ &= \left(\mathbb{E}\left[S_t f(Z)\right]\right)_{t\in T}\cdot\alpha\ .\end{aligned}$$

The other side yields $\mathsf{Var}\left[\mathbb{E}\left[L|f(Z)\right]\cdot\alpha\right] = \mathsf{Var}\left[(S_t)_{t\in T}\cdot\alpha\right]$. Now if $S_t = \beta_t f(Z)$, then we have for both sides

$$\begin{cases}\mathsf{Cov}\left[L\cdot\alpha; f(Z)\right]^2 & = (\alpha\cdot\beta)^2\,\mathbb{E}\left[f(Z)^2\right]^2,\\ \mathsf{Var}\left[\mathbb{E}\left[L|f(Z)\right]\cdot\alpha\right] & = \mathsf{Var}\left[(\alpha\cdot\beta)\,f(Z)\right] = (\alpha\cdot\beta)^2\,\mathbb{E}\left[f(Z)^2\right],\end{cases}$$

which proves equivalence. □

B Proof of Lemma 1

Proof.

$$\begin{aligned}\underset{\|\alpha\|=1}{\operatorname{argmax}}\,\mathsf{Cov}\left[L\cdot\alpha, f(Z)\right]^2 &= \underset{\|\alpha\|=1}{\operatorname{argmax}}\,(\alpha\cdot\beta)^2\,\mathbb{E}\left[f(Z)^2\right]^2\\ &= \underset{\|\alpha\|=1}{\operatorname{argmax}}\,(\alpha\cdot\beta)^2,\ \text{because}\ \mathbb{E}\left[f(Z)^2\right]^2 > 0.\end{aligned}$$

By the Cauchy-Schwarz theorem, we have: $(\alpha\cdot\beta)^2 \leqslant \|\alpha\|^2\times\|\beta\|^2$,where equality holds if and only if α and β are linearly dependent, i.e., $\alpha = \lambda\beta$. Accordingly, if $\|\alpha\| = 1$ we have $\lambda = \frac{1}{\|\beta\|}$, which gives us the required solution. □

C Proof of Proposition 4

Proof. We have

$$\mathsf{Cov}\left[L\cdot\alpha, f(Z)\right] = \left(\mathsf{Cov}\left[L_t; f(Z)\right]\right)_{t\in T}\cdot\alpha\ .$$

Similar to the proof of Lemma 1, we use the Cauchy-Schwarz inequality. In particular,

$$\left((\mathsf{Cov}\,[L_t; f(Z)])_{t\in T}\cdot\alpha\right)^2 \leqslant \|\alpha\|^2 \times \|(\mathsf{Cov}\,[L_t; f(Z)])_{t\in T}\|^2.$$

We have the equality,

$$\left((\mathsf{Cov}\,[L_t; f(Z)])_{t\in T}\cdot\alpha\right)^2 = \|\alpha\|^2 \times \|(\mathsf{Cov}\,[L_t; f(Z)])_{t\in T}\|^2,$$

if and only if $\alpha = \lambda\,(\mathsf{Cov}\,[L_t; f(Z)])_{t\in T}$.

So, if $\|\alpha\| = 1$ we have $\lambda = \frac{1}{\|(\mathsf{Cov}[L_t; f(Z)])_{t\in T}\|}$. □