Robust and One-Pass Parallel Computation of Correlation-Based Attacks at Arbitrary Order

Tobias Schneider[1], Amir Moradi[1], and Tim Güneysu[2]

[1]Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Germany
{tobias.schneider-a7a, amir.moradi}@rub.de
[2]University of Bremen, Germany
{tim.gueneysu}@uni-bremen.de

Abstract. The protection of cryptographic implementations against higher-order attacks has risen to an important topic in the side-channel community after the advent of enhanced measurement equipment that enables the capture of millions of power traces in reasonably short time. However, the preprocessing of multi-million traces for such an attack is still challenging, in particular when in the case of (multivariate) higher-order attacks all traces need to be parsed at least two times. Even worse, partitioning the captured traces into smaller groups to parallelize computations is hardly possible with current techniques.

In this work we introduce procedures that allow iterative computation of correlation in a side-channel analysis attack at any arbitrary order in both univariate and multivariate settings. The advantages of our proposed solutions are manifold: i) they provide stable results, i.e., by increasing the number of used traces high accuracy of the estimations is still maintained, ii) each trace needs to be processed only once and at any time the result of the attack can be obtained (without requiring to reparse the whole trace pool when adding more traces), iii) the computations can be efficiently parallelized, e.g., by splitting the trace pool into smaller subsets and processing each by a single thread on a multi-threading or cloud-computing platform, and iv) the computations can be run in parallel to the measurement phase. In short, our constructions allow efficiently performing higher-order side-channel analysis attacks (e.g., on hundreds of million traces) which is of crucial importance when practical evaluation of the masking schemes need to be performed.

1 Introduction

Side-channel analysis (SCA) poses a major threat for security-sensitive applications. This becomes particularly critical when the cryptographic device – particularly in pervasive applications – is delivered to the end user, where it is operated in a hostile environment (cf. [17, 22]). For such a case the integration of appropriate countermeasures against SCA attacks has become essential in the design of the device. In this context, *masking* as a countermeasure obtained the most attraction from both academia and industry due to its sound theoretical

basis as well as its practical efficiency to mitigate the attacks. Masking countermeasures are based on the principle of *secret sharing* for which many different forms including Boolean, arithmetic, multiplicative, polynomial base, etc. have been proposed [5, 6, 19].

Since the efficiency of a masking schemes strongly depends on its implementation, a practical evaluation of the final product (or a prototype) is inevitable. For this situation, techniques such as the *test vector leakage assessment* [9] (known as t-test) have been developed to practically examine the vulnerability of a cryptographic design. However, such an evaluation scheme can only report the *existence* of a leakage in a product, but it does not provide any indication whether this leakage is indeed *exploitable* by an attack. In reply to the question if a leakage is in fact exploitable for key recovery, one needs to mount different SCA attacks and examine their success. Depending on the definition and settings of the masking scheme, it can provide security against SCA attacks up to a certain order d. Consequently, all tests and attacks need to take all particular orders ranging from 1 up to $d + 1$ into account.

The most common SCA attack, Correlation Power Analysis (CPA) [4], is based on a hypothetical leakage model and the estimation of correlation (commonly by Pearson's correlation coefficient) between the hypothetical leakages and the SCA traces. In its simplest setting, the attack runs independently at each sample point of the SCA traces. This univariate first-order CPA can be extended to higher orders $d > 1$ by introducing a preprocessing stage for the traces at each sample point. This preprocessing involves the computation of mean-free values which are then squared (for a univariate $d = 2$nd-order CPA), cubed (for a univariate $d = 3$rd-order CPA), or any corresponding power for larger d. Prior to the attack d different sample points of each trace are combined into a centered product for the multivariate case at order $d > 1$. In other words, first mean-free representations are calculated of which d sample points of each trace are multiplied. It is noteworthy that finding such d points of interest is another challenging task which has been well studied in [8, 18].

By increasing the order of the underlying masking scheme the corresponding higher-order CPA becomes more susceptible to noise. Indeed the number of required traces to mount a successful attack increases exponentially in d with respect to the noise standard deviation. Therefore, a higher-order attack typically requires several (hundreds of) millions of traces to be successful [1, 13]. The conventional strategy for preprocessing (known as "three-pass") parses all traces three times to i) obtain the means, ii) combine the desired points by their mean-free product, and iii) estimate the correlation[1]. This procedure has many shortcomings as by adding more traces to the trace pool, the entire last two steps need to be repeated. Hence, it is not easily possible to parallelize the computations by splitting the trace pool into smaller sets. We should emphasize that, in case of univariate attacks, the parallelization can be trivially done by splitting each trace into smaller subtraces with a lower number of sample points.

[1] In some particular cases, e.g., univariate, the last two steps can be combined.

Alternatively as shown in [3] for first-order and second-order CPA, the formulas for preprocessing and the estimation of the correlation can be combined by following the displacement law. This procedure (so-called "Raw-Moment") solves all the shortcomings of the three-pass approach. In fact:

- When increasing the trace pool, the estimated raw moments are easily updated by only processing the given new traces.
- The attack can be started before the measurement phase is completed. This helps to further increase the performance of the attacks.
- The result of the attack can be obtained without introducing any overhead to the process of the further traces at any time during the measurement phase.
- The trace pool can be easily split into smaller sets and each set can be processed independently by different threads. Due to the nature of the raw moments, the result of different threads (at any time) can be easily combined to derive the result of the attack.

Note, however, that this procedure was only presented for first-order and bivariate second-order CPA using 10,000,000 traces and may suffer from numerical instabilities as the raw moments become pretty large values by increasing the number of traces. Hence, it can lead to serious accuracy loss due to the limited fraction significand of floating point formats (e.g., IEEE 754). This issue becomes extremely problematic for higher-order ($d > 2$) attacks.

The instability in formulas that are based on raw moments has been previously studied to come up for appropriate solutions. For example, in [15] robust iterative formulas for centralized and standardized moments at any arbitrary order as well as for correlation are given that avoid such instabilities by increasing the number of samples. Furthermore, iterative formulas for the t-test at any arbitrary order are given in [20].

Our Contribution: In this work, we present an approach based on centralized and standardized moments to cover univariate as well as multivariate CPA attacks at any arbitrary order. Our solution benefits from all the aforementioned advantages of the raw-moment approach while it maintains the accuracy (as for the three-pass approach) regardless of the order of the attack and the number of traces. This work not only covers CPA attacks but also Moments-Correlating DPA [14] where moments are correlated to the (preprocessed) traces with the goal of avoiding the necessity of a hypothetical leakage model (that is unavoidable in CPA attacks).

Prior to the description of our solution we define two terms *iterative* and *incremental* which are frequently used in the rest of the paper. Suppose that after finishing all the required processes on the trace pool $\mathcal{Q}$, a new trace y is added to the trace pool $\mathcal{Q}' = \mathcal{Q} \cup \{y\}$. We provide *incremental* formulas that allow updating the previously computed terms by only processing the new trace y. In addition to that, we suppose that the trace pool $\mathcal{Q}$ is divided into two groups as $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$, and each group is independently processed using the given incremental formulas. We provide (two-pair) *iterative* formulas that enable

the combination of results computed over each group $\mathcal{Q}_1$ and $\mathcal{Q}_2$ to derive the result of the full trace pool $\mathcal{Q}$.

2 Notations

We use capital letters for random variables, and lower-case letters for their realizations. Vectors are denoted with bold notations, functions with sans serif fonts, and sets with calligraphic ones.

Suppose that in a side-channel attack, with respect to n queries with associated data (e.g., plaintext or ciphertext) $\boldsymbol{d}_{i\in\{1,\ldots,n\}}$, n side-channel measurements (so-called traces) are collected. Let us denote each trace by $\boldsymbol{t}_{i\in\{1,\ldots,n\}}$ containing m sample points $\{t_i^{(1)},\ldots,t_i^{(m)}\}$.

Following the divide-and-conquer principle, one objective of a side-channel attack is to recover a part k of the secret key $\boldsymbol{k}$, which contributed to the processing of the entire associated data $\boldsymbol{d}_{i\in\{1,\ldots,n\}}$. Prior to the attack an intermediate value V is selected, which given the associated data and a key guess k is predictable, i.e., $v_i = \mathsf{F}(\boldsymbol{d}_i, k)$. In a CPA attack a hypothetical leakage model $\widetilde{\mathsf{L}}(.)$ is applied on the chosen intermediate value which should be (sufficiently) linearly proportional to the actual leakage of the target device, i.e., $\mathsf{L}(.)$. As a common and straightforward example, the Hamming weight of an Sbox output during the first round of an encryption function is employed when attacking an exemplary micro-processor based implementation, i.e., $l_i = \widetilde{\mathsf{L}}(v_i) = HW(\mathsf{S}(d_i \oplus k))$, where d_i denotes a necessary part of $\boldsymbol{d}_i$ to predict v_i.

Let us denote the dth-order raw statistical moment of a random variable X by $M_d = \mathsf{E}(X^d)$, with $\mu = M_1$ the mean and $\mathsf{E}(.)$ the expectation operator. We also denote the dth-order ($d > 1$) central moment by $CM_d = \mathsf{E}\left((X-\mu)^d\right)$, with $s^2 = CM_2$ the variance. Finally, the dth-order ($d > 2$) standardized moment is denoted by $SM_d = \mathsf{E}\left(\left(\frac{X-\mu}{s}\right)^d\right)$, with SM_3 the skewness and SM_4 the kurtosis.

3 Univariate CPA

For a *univariate* CPA attack the correlation between the traces $\boldsymbol{T}$ and the hypothetical leakage values L is estimated. Due to the *univariate* nature of the attack, such a process is performed at each sample point $(1,\ldots,m)$ independently. Therefore, below – for simplicity – we omit the upper index of the sample points and denote a sample point of the ith trace by t_i.

The estimation of the correlation with Pearson correlation coefficient (as the normalized covariance) is defined as

$$\rho = \frac{\mathrm{cov}(T,L)}{s_t\, s_l} = \frac{\mathsf{E}\big((T-\mu_t)(L-\mu_l)\big)}{s_t\, s_l}, \tag{1}$$

where μ_t (resp. μ_l) denotes the estimated mean of the traces (resp. of the hypothetical leakages). s_t (resp. s_l) also stands for standard deviation.

In the discrete domain we can write

$$\rho = \frac{\frac{1}{n}\sum_{i=1}^{n}(t_i-\mu_t)(l_i-\mu_l)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(t_i-\mu_t)^2\,\frac{1}{n}\sum_{i=1}^{n}(l_i-\mu_l)^2}} \tag{2}$$

Based on the way followed in [3] one can write

$$\rho = \frac{\frac{1}{n}\sum_{i=1}^{n}t_i\,l_i-\mu_t\,\mu_l}{\sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}{t_i}^2-{\mu_t}^2\right)\left(\frac{1}{n}\sum_{i=1}^{n}{l_i}^2-{\mu_l}^2\right)}} = \frac{M_{1,\mathcal{T}\cdot\mathcal{L}}-M_{1,\mathcal{T}}\,M_{1,\mathcal{L}}}{\sqrt{\left(M_{2,\mathcal{T}}-{M_{1,\mathcal{T}}}^2\right)\left(M_{2,\mathcal{L}}-{M_{1,\mathcal{L}}}^2\right)}}, \tag{3}$$

which are based on dth-order raw moments, i.e., $M_{d,\mathcal{X}} = \frac{1}{n}\sum_{i=1}^{n}{x_i}^d$. However, as stated in [20], such constructions can lead to numerically unstable situations [10]. During the computation of the raw moments the intermediate values tend to become very large which can lead to a loss in accuracy. Further, M_2 and ${M_1}^2$ can be large values, and the result of $M_2-{M_1}^2$ can also lead to a significant accuracy loss due to the limited fraction significand of floating point formats (e.g., IEEE 754).

Iterative. We can alternatively write

$$\rho = \frac{\frac{1}{n}\sum_{i=1}^{n}(t_i-\mu_t)(l_i-\mu_l)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(t_i-\mu_t)^2\,\frac{1}{n}\sum_{i=1}^{n}(l_i-\mu_l)^2}} = \frac{\frac{1}{n}\,ACS_1}{\sqrt{\frac{1}{n}\,CS_{2,\mathcal{T}}\,\frac{1}{n}\,CS_{2,\mathcal{L}}}}, \tag{4}$$

with $CS_{d,\mathcal{X}} = \sum_{i=1}^{n}(x_i-\mu_x)^d$ as the definition of dth-order *centralized sum* given in [20]. Further, we define ACS_1 as the first-order *adjusted centralized sum*.

Suppose that $M_{1,\mathcal{Q}_1}$ (resp. $M_{1,\mathcal{Q}_2}$) denotes the first raw moment (sample mean) of the given set $\mathcal{Q}_1$ (resp. $\mathcal{Q}_2$) with cardinality $n_1=|\mathcal{Q}_1|$ and $n_2=|\mathcal{Q}_2|$. $M_{1,\mathcal{Q}}$ as the first raw moment of $\mathcal{Q}=\mathcal{Q}_1\cup\mathcal{Q}_2$ can be written as [15]

$$M_{1,\mathcal{Q}} = \frac{n_1\,M_{1,\mathcal{Q}_1}+n_2\,M_{1,\mathcal{Q}_2}}{n}, \tag{5}$$

with $n=n_1+n_2$ as the cardinality of $\mathcal{Q}$.

In the same way, such a formula can be written for the centralized sum $CS_{d,\mathcal{Q}}$ at any arbitrary order $d > 1$ as [15]

$$CS_{d,\mathcal{Q}} = CS_{d,\mathcal{Q}_1} + CS_{d,\mathcal{Q}_2} + \sum_{p=1}^{d-2} \binom{d}{p} \left[\left(\frac{-n_2}{n}\right)^p CS_{d-p,\mathcal{Q}_1} + \left(\frac{n_1}{n}\right)^p CS_{d-p,\mathcal{Q}_2} \right] \Delta^p + \left(\frac{n_1\, n_2}{n}\, \Delta\right)^d \left[\left(\frac{1}{n_2}\right)^{d-1} - \left(\frac{-1}{n_1}\right)^{d-1} \right], \tag{6}$$

with $\Delta = M_{1,\mathcal{Q}_2} - M_{1,\mathcal{Q}_1}$. It is noteworthy that the calculation of $CS_{d,\mathcal{Q}}$ additionally requires $CS_{p,\mathcal{Q}_1}$ and $CS_{p,\mathcal{Q}_2}$ for $1 < p \leq d$.

The remaining part is the first-order adjusted centralized sum ACS_1. Suppose that $\mathcal{Q}_1$ and $\mathcal{Q}_2$ denote sets of doubles (t, l) with first-order adjusted centralized sum $ACS_{1,\mathcal{Q}_1}$ and $ACS_{1,\mathcal{Q}_2}$ respectively. The first-order adjusted centralized sum of $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$ can be written as

$$ACS_{1,\mathcal{Q}} = ACS_{1,\mathcal{Q}_1} + ACS_{1,\mathcal{Q}_2} + \frac{n_1\, n_2}{n}\, \Delta_t\, \Delta_l, \tag{7}$$

with $\Delta_t = \mu_{t,\mathcal{Q}_2} - \mu_{t,\mathcal{Q}_1}$ and $\Delta_l = \mu_{l,\mathcal{Q}_2} - \mu_{l,\mathcal{Q}_1}$. For simplicity, we denote $M_{1,\mathcal{T}_1}$ by $\mu_{t,\mathcal{Q}_1}$ and $M_{1,\mathcal{L}_1}$ by $\mu_{l,\mathcal{Q}_1}$. The sets $\mathcal{T}_1$ and $\mathcal{L}_1$ are formed respectively from the first and second elements of the doubles in $\mathcal{Q}_1$ (the same holds for $\mathcal{Q}_2$, $\mu_{t,\mathcal{Q}_2}$, and $\mu_{l,\mathcal{Q}_2}$).

Incremental, $n_2 = 1$. We now optimize the computations of each set. It is indeed enough to suppose that $\mathcal{Q}_2$ consists of only one element y. Hence the update formula for the first raw moment can be written as

$$M_{1,\mathcal{Q}} = M_{1,\mathcal{Q}_1} + \frac{\Delta}{n},$$

with $\Delta = y - M_{1,\mathcal{Q}_1}$. Note that $\mathcal{Q}_1$ and $M_{1,\mathcal{Q}_1}$ are initialized with $\emptyset$ and respectively zero. Similarly, we can write the same for the dth-order centralized sum

$$CS_{d,\mathcal{Q}} = CS_{d,\mathcal{Q}_1} + \sum_{p=1}^{d-2} \binom{d}{p} CS_{d-p,\mathcal{Q}_1} \left(\frac{-\Delta}{n}\right)^p + \left(\frac{n-1}{n}\Delta\right)^d \left[1 - \left(\frac{-1}{n-1}\right)^{d-1}\right], \tag{8}$$

where $\Delta = y - M_{1,\mathcal{Q}_1}$. For the first-order adjusted centralized sum we can also write

$$ACS_{1,\mathcal{Q}} = ACS_{1,\mathcal{Q}_1} + \frac{n-1}{n}\, \Delta_t\, \Delta_l, \tag{9}$$

with $\Delta_t = t_n - \mu_{t,\mathcal{Q}_1}$ and $\Delta_l = l_n - \mu_{l,\mathcal{Q}_1}$, where $\mathcal{Q}_2 = \{(t_n, l_n)\}$.

Based on these formulas the correlation can be computed efficiently in one pass. Furthermore, since the intermediate results of the central sums are mean-free, they do not become significantly large which helps preventing the numerical instabilities.

3.1 Univariate Higher-Order CPA

Higher-order attacks require that the sample traces are preprocessed. For the second-order univariate CPA the preprocessing consists of making each sample point mean-free squared:

$$t'_i = (t_i - \mu_t)^2 .$$

For higher orders $d > 2$ the traces are usually additionally standardized as $\frac{t'_i}{{s_t}^d}$, where s_t denotes the standard deviation. Therefore, the Pearson correlation can be written as

$$\rho = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{t'_i}{{s_t}^d} - \frac{\mu_{t'}}{{s_t}^d}\right)(l_i - \mu_l)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{t'_i}{{s_t}^d} - \frac{\mu_{t'}}{{s_t}^d}\right)^2 \frac{1}{n}\sum_{i=1}^{n}(l_i - \mu_l)^2}} = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(t'_i(l_i - \mu_l)\right)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(t'_i - \mu_{t'})^2 \frac{1}{n}\sum_{i=1}^{n}(l_i - \mu_l)^2}} \tag{10}$$

The straightforward way is to first preprocess the entire trace set $t_{i\in\{1,\ldots,n\}}$. Hence the measurement phase has to be completed before the preprocessing can be started. Another drawback is the reduced efficiency as each of the preprocessing and the estimation of the correlation steps needs at least one pass over the whole trace set.

In [3], the authors propose iterative formulas for first- and second-order CPA. Their approach is based on raw moments which can lead to numerical instability if the values get too large [20]. Alternatively, we propose an iterative method which is based on the centralized moments. These values are mean-free which leads to smaller values and better accuracy for a large number of measurements. This approach can be run in parallel to the measurements (and can be also split into smaller threads) as the result is incrementally updated for each new measurement. Therefore, it needs only one pass over the whole trace set. In the following, we present all necessary iterative formulas to perform a univariate CPA at any arbitrary order with sufficient accuracy. We divide the expressions by the numerator and denominator of Equation (10).

3.2 Numerator

Note that even though the numerator looks similar to a raw-moment approach, it operates with centralized (mean-free) values. Therefore, numerical instabilities are avoided. The numerator for the d-th order correlation can be written as

$$\frac{1}{n}\sum_{i=1}^{n}\left(t'_i(l_i - \mu_l)\right) = \frac{1}{n}\sum_{i=1}^{n}(t_i - \mu_t)^d (l_i - \mu_l) = \frac{1}{n} ACS_d, \tag{11}$$

with ACS_d which we refer to as the dth-order *adjusted centralized sum*.

We start with a generic formula which merges the adjusted centralized sum of two sets $\mathcal{Q}_1 \cup \mathcal{Q}_2 = \mathcal{Q}$ with $|\mathcal{Q}_1| = n_1$, $|\mathcal{Q}_2| = n_2$ and $|\mathcal{Q}| = n$. The goal is to compute $ACS_{d,\mathcal{Q}}$ given only the adjusted and centralized sums of $\mathcal{Q}_1$ and $\mathcal{Q}_2$.

Theorem 1. *Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be given sets of doubles (t, l). Suppose also $\mathcal{T}_1$ and $\mathcal{L}_1$ as the sets of respectively the first and second elements of the doubles in $\mathcal{Q}_1$ (the same for $\mathcal{T}_2$ and $\mathcal{L}_2$). The dth-order adjusted centralized sum $ACS_{d,\mathcal{Q}}$ of the extended set $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$ with $\Delta_t = \mu_{t,\mathcal{Q}_2} - \mu_{t,\mathcal{Q}_1}$ and $\Delta_l = \mu_{l,\mathcal{Q}_2} - \mu_{l,\mathcal{Q}_1}$ can be written as*

$$\begin{aligned} ACS_{d,\mathcal{Q}} = {} & ACS_{d,\mathcal{Q}_1} + ACS_{d,\mathcal{Q}_2} + \frac{\Delta_l}{n}\left(n_1\, CS_{d,\mathcal{Q}_2} - n_2\, CS_{d,\mathcal{Q}_1}\right) \\ & + \sum_{p=1}^{d-1} \binom{d}{p} \left(\frac{\Delta_t}{n}\right)^p \Bigg[(-n_2)^p\ ACS_{d-p,\mathcal{Q}_1} + (n_1)^p\ ACS_{d-p,\mathcal{Q}_2} \\ & + \frac{\Delta_l}{n}\left((-n_2)^{p+1}\ CS_{d-p,\mathcal{Q}_1} + (n_1)^{p+1}\ CS_{d-p,\mathcal{Q}_2}\right)\Bigg] \\ & + \frac{\left(n_1\,(-n_2)^{d+1} + n_2\,(n_1)^{d+1}\right)}{n^{d+1}}\,(\Delta_t)^d\ \Delta_l \end{aligned} \tag{12}$$

The proof of Theorem 1 is omitted due to length restrictions.

Incremental, $n_2 = 1$. For the iterative formulas when $\mathcal{Q}_2 = \{(t_n, l_n)\}$ Equation (12) can be simplified to

$$\begin{aligned} ACS_{d,\mathcal{Q}} = {} & ACS_{d,\mathcal{Q}_1} + CS_{d,\mathcal{Q}_1}\left(-\frac{\Delta_l}{n}\right) \\ & + \sum_{p=1}^{d-1} \binom{d}{p} \left(-\frac{\Delta_t}{n}\right)^p \left[ACS_{d-p,\mathcal{Q}_1} + CS_{d-p,\mathcal{Q}_1}\left(-\frac{\Delta_l}{n}\right)\right] \\ & + \frac{(-1)^{d+1}\,(n-1) + (n-1)^{d+1}}{n^{d+1}}\,(\Delta_t)^d\,\Delta_l, \end{aligned} \tag{13}$$

with $\Delta_t = t_n - \mu_{t,\mathcal{Q}_1}$ and $\Delta_l = l_n - \mu_{l,\mathcal{Q}_1}$.

3.3 Denominator

The denominator of Equation (10) requires the computation of two centralized sums. For the second centralized sum $\sum_{i=1}^{n} (l_i - \mu_l)^2$ we already gave pair-wise iterative as well as incremental formulas for $CS_{2,\mathcal{Q}}$ in Equation (6) and Equation (8).

The first centralized sum $\sum_{i=1}^{n} (t'_i - \mu_{t'})^2$ relates to the preprocessed traces. For this, efficient formulas to compute the variance of the preprocessed traces are given in [20]. In order to estimate the variance (second centralized moment $CM_{2,\mathcal{T}'}$) of $\mathcal{T}' = \{t'_{i\in\{1,\ldots,n\}}\}$ as the set of preprocessed traces at any arbitrary order $d > 1$ we can write [20]

$$\frac{1}{n}\sum_{i=1}^{n} (t'_i - \mu_{t'})^2 = CM_{2,\mathcal{T}'} = CM_{2d,\mathcal{T}} - (CM_{d,\mathcal{T}})^2 = \frac{CS_{2d,\mathcal{T}}}{n} - \left(\frac{CS_{d,\mathcal{T}}}{n}\right)^2,$$

where $\mathcal{T}$ denotes the traces without preprocessing. Therefore, given the iterative and incremental formulas for $CS_{d,\mathcal{Q}}$ in Equation (6) and Equation (8) we can efficiently as well as in parallel estimate both centralized sums of the denominator of Equation (10). Further, having the formulas given in Section 3.2 the correlation of a univariate CPA at any arbitrary order d can be easily derived.

4 Multivariate CPA

In the following we give iterative formula for multivariate higher-order CPA with the optimum combination function, i.e., centered product [16, 21]. Given d sample point indices $\mathcal{J} = \{j_1, ..., j_d\}$ as the points to be combined and a set of sample vectors $\mathcal{Q} = \{\boldsymbol{V}_{i\in\{1,...,n\}}\}$ with $\boldsymbol{V}_i = \left(t_i^{(j)} \mid j \in \mathcal{J}\right)$, the centered product of the ith trace is defined as

$$c_i = \prod_{j\in\mathcal{J}} \left(t_i^{(j)} - \mu_{\mathcal{Q}}^{(j)}\right), \tag{14}$$

where $\mu_{\mathcal{Q}}^{(j)}$ denotes the mean at sample point j over set $\mathcal{Q}$.

The authors of [3] proposed an iterative formula for the Pearson correlation coefficient in the bivariate case, i.e., $d = 2$. However, during the computation they calculate the sum $\sum_{i=1}^{n} \left(t_i^{(j_1)}\, t_i^{(j_2)}\right)^2$ for the two point indices j_1 and j_2 (cf. s_{11} of Table 5 in [3]). Their method is basically equivalent to using the raw moments to derive higher-order statistical moments. Given a high number of traces this value can grow very large, and can cause numerical instability.

We instead provide iterative formulas based on mean-free values. In our approach, the formula for the multivariate Pearson correlation coefficient is first simplified using Equation (10) to

$$\rho = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(c_i - \mu_c\right)\left(l_i - \mu_l\right)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(c_i - \mu_c\right)^2 \frac{1}{n}\sum_{i=1}^{n}\left(l_i - \mu_l\right)^2}} = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(c_i\left(l_i - \mu_l\right)\right)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(c_i - \mu_c\right)^2 \frac{1}{n}\sum_{i=1}^{n}\left(l_i - \mu_l\right)^2}}. \tag{15}$$

4.1 Numerator

The way of computing the numerator of Equation (15)

$$\frac{1}{n}\sum_{i=1}^{n}\left(c_i\left(l_i - \mu_l\right)\right) = \frac{1}{n}\sum_{i=1}^{n}\Big(\prod_{j\in\mathcal{J}}\left(t_i^{(j)} - \mu_{\mathcal{Q}}^{(j)}\right)\left(l_i - \mu_l\right)\Big) \tag{16}$$

is similar to the iterative computation of the first parameter for the multivariate t-test as presented in [20]. We indeed can write Equation (16) as

$$\frac{1}{n}\sum_{i=1}^{n}\left(c_i\left(l_i - \mu_l\right)\right) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j\in\mathcal{J}'}\left(t_i^{(j)} - \mu_{\mathcal{Q}}^{(j)}\right), \tag{17}$$

with $\mathcal{J}' = \mathcal{J} \cup \{j^*\}$, $t_i^{(j*)} = l_i$ and $\mu_{\mathcal{Q}}^{(j*)} = \mu_l$. With this, we define the term *sum of centered products* as

$$SCP_{d+1,\mathcal{Q},\mathcal{J}'} = \sum_{\boldsymbol{V}_i \in \mathcal{Q}} \prod_{j \in \mathcal{J}'} \left(t_i^{(j)} - \mu_{\mathcal{Q}}^{(j)} \right). \tag{18}$$

In addition, we define the b-th order power set of $\mathcal{J}'$ as

$$\mathcal{P}_b = \{\mathcal{S} \mid \mathcal{S} \in \mathbb{P}(\mathcal{J}'), |\mathcal{S}| = b\}, \tag{19}$$

where $\mathbb{P}(\mathcal{J}')$ refers to the power set of the indices of the points of interest $\mathcal{J}'$. The given formulas in [20] are for the incremental case when set $\mathcal{Q}_2$ has a cardinality of 1. Hence, the sum of the centered products $SCP_{d+1,\mathcal{Q},\mathcal{J}'}$ of the extended set $\mathcal{Q} = \mathcal{Q}_1 \cup \left\{ (t_n^{(j_1)}, ..., t_n^{(j_d)}, t_n^{(j^*)}) \right\}$ with $t_n^{(j^*)} = l_n$ and $|\mathcal{Q}| = n$ can be computed as [20]

$$\begin{aligned} SCP_{d+1,\mathcal{Q},\mathcal{J}'} = SCP_{d+1,\mathcal{Q}_1,\mathcal{J}'} &+ \left(\sum_{b=2}^{d} \sum_{\mathcal{S} \in \mathcal{P}_b} SCP_{b,\mathcal{Q}_1,\mathcal{S}} \prod_{j \in \mathcal{J}' \setminus \mathcal{S}} \left(\frac{\Delta^{(j)}}{-n} \right) \right) \\ &+ \left(\frac{(-1)^{d+1}(n-1) + (n-1)^{d+1}}{n^{d+1}} \prod_{j \in \mathcal{J}'} \Delta^{(j)} \right), \end{aligned} \tag{20}$$

where $\Delta^{(j \in \mathcal{J}')} = t_n^{(j)} - \mu_{\mathcal{Q}_1}^{(j)}$. Below we present a generalization of this method to arbitrary sized $\mathcal{Q}_2$.

Generalization of [20]

Theorem 2. *Let $\mathcal{J}'$ be a given set of indices (of $d+1$ points of interest) and two sets of sample vectors $\mathcal{Q}_1 = \{\boldsymbol{V}_{i \in \{1,...,n_1\}}\}$, $\mathcal{Q}_2 = \{\boldsymbol{V}_{i \in \{1,...,n_2\}}\}$ with $\boldsymbol{V}_i = \left(t_i^{(j)} \mid j \in \mathcal{J}' \right)$. The sum of the centered products $SCP_{d+1,\mathcal{Q},\mathcal{J}'}$ of the extended set $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$ with $\Delta^{(j \in \mathcal{J}')} = \mu_{\mathcal{Q}_2}^{(j)} - \mu_{\mathcal{Q}_1}^{(j)}$ and $|\mathcal{Q}| = n$ can be computed as:*

$$\begin{aligned} SCP_{d+1,\mathcal{Q},\mathcal{J}'} = SCP_{d+1,\mathcal{Q}_1,\mathcal{J}'} &+ SCP_{d+1,\mathcal{Q}_2,\mathcal{J}'} \\ &+ \sum_{b=2}^{d} \sum_{\mathcal{S} \in \mathcal{P}_b} \left((-n_2)^{d+1-b} SCP_{b,\mathcal{Q}_1,\mathcal{S}} + n_1^{d+1-b} SCP_{b,\mathcal{Q}_2,\mathcal{S}} \right) \prod_{j \in \mathcal{J}' \setminus \mathcal{S}} \frac{\Delta^{(j)}}{n} \\ &+ \frac{(-n_2)^{d+1} n_1 + n_1^{d+1} n_2}{n^{d+1}} \prod_{j \in \mathcal{J}'} \Delta^{(j)}. \end{aligned} \tag{21}$$

The proof of Theorem 2 is omitted due to length restrictions.

4.2 Denominator

Similar to the expressions given in Section 3.3 the denominator of Equation (15) consists of two centralized sums. The second one $\sum_{i=1}^{n} (l_i - \mu_l)^2$ is the same as that of the univariate CPA and Equation (6) and Equation (8) are still valid.

For the first centralized sum $\sum_{i=1}^{n} \left(c_i - \mu_c\right)^2$ we recall the formulas given in [20] which deal with the estimation of the variance of the preprocessed traces in a multivariate setting. It means that we can write

$$\sum_{i=1}^{n} \left(c_i - \mu_c\right)^2 = \sum_{\boldsymbol{V} \in \mathcal{Q}} \left(\prod_{j \in \mathcal{J}} \left(t^{(j)} - \mu_{\mathcal{Q}}^{(j)}\right) - \frac{SCP_{d,\mathcal{Q},\mathcal{J}}}{n} \right)^2$$
$$= SCP_{2d,\mathcal{Q},\mathcal{J}''} - \frac{\left(SCP_{d,\mathcal{Q},\mathcal{J}}\right)^2}{n}, \tag{22}$$

with multiset $\mathcal{J}'' = \{j_1, ..., j_d, j_1, ..., j_d\}$. It is noteworthy that in contrast to the computation of the numerator, where the set $\mathcal{J}'$ with $d+1$ indices is used, here for the denominator the set $\mathcal{J}$ and its extension $\mathcal{J}''$ with respectively d and $2d$ indices are applied.

5 Moments-Correlating DPA

Moments-Correlating DPA (MC-DPA) [14] as a successor of Correlation-Enhanced Power Analysis Collision Attack [12] solves its shortcomings and is based on correlating the moments to the traces [7, 8, 11]. It relaxes the necessity of a hypothetical leakage model which is essential in the case of a CPA.

The most general form of MC-DPA is Moments-Correlating Profiling DPA (MCP-DPA). In such a scenario, the traces used to build the model $\boldsymbol{t}^{(M)}_{i \in \{1,...,n^{(M)}\}}$ (and trivially their number $n^{(M)}$) are not necessarily the same as the traces used in the attack $\boldsymbol{t}_{i \in \{1,...,n\}}$. An MC-DPA in a multivariate settings uses two sets of sample point indices $\mathcal{J}_M$ and $\mathcal{J}_t$ related to the sample points of the model and the attack respectively. Such sample points are taken based on the time instances when a certain function (e.g., an Sbox) operates on an intermediate value $v^{(M)}_{i \in \{1,...,n^{(M)}\}}$ to form the model and on another intermediate value $v^{(t)}_{i \in \{1,...,n\}}$ to perform the attack. In a simple scenario, such intermediate values can be different Sbox inputs. Optionally a leakage function can be considered as $\widetilde{\mathsf{L}}(.)$ over the targeted intermediate values. Note that in the most general form such a leakage function can be the identity mapping, i.e., $\widetilde{\mathsf{L}}(v) = v$. Following the original MC-DPA scheme [14], $v_i^{(M)} = d_i^{(M)} \oplus k^{(M)}$ and $v_i^{(t)} = d_i^{(t)} \oplus k^{(t)}$ with $d^{(M)}$ and $d^{(t)}$ e.g., plaintext portions (bytes) respectively of the model and the attack. Hence, due to the linear relations such a setting turns into a linear collision attack [2] with $\widetilde{\mathsf{L}}(v_i^{(M)}) = d_i^{(M)}$ and $\widetilde{\mathsf{L}}(v_i^{(t)}) = d_i^{(t)} \oplus \Delta k$, which is referred to as Moments-Correlating Collision DPA (MCC-DPA), where the traces for the model and the

attack are the same and $n^{(M)} = n$. However, in the following expressions we consider the profiling one which can be easily simplified to the collision one.

Let us denote $\mathcal{L}$ as a set of all possible outputs of the leakage function with cardinality of $n_{\mathcal{L}}$ is defined as

$$\mathcal{L} = \{l^{(1)}, \ldots, l^{(n_{\mathcal{L}})}\} = \{l \,|\, \exists v,\, \widetilde{\mathsf{L}}(v) = l\}. \tag{23}$$

Correspondingly we define $n_{\mathcal{L}}$ subsets $\mathcal{I}^{(M)}_{l^{(a\in\{1,\ldots,n_{\mathcal{L}}\})}}$

$$\mathcal{I}^{(M)}_{l^{(a)}} = \{i \in \{1, \ldots, n^{(M)}\} \,|\widetilde{\mathsf{L}}(v_i^{(M)}) = l^{(a)}\} \tag{24}$$

as the trace indices with particular leakage value $l^{(a)}$ on the model's intermediate values $v_i^{(M)}$ with cardinality of $n^{(M)}_{l^{(a)}}$. The same subsets are also defined with respect to the attack's intermediate values $v_i^{(t)}$as

$$\mathcal{I}^{(t)}_{l^{(a)}} = \{i \in \{1, \ldots, n\} \,|\widetilde{\mathsf{L}}(v_i^{(t)}) = l^{(a)}\}, \tag{25}$$

with $|\mathcal{I}^{(t)}_{l^{(a)}}| = n^{(t)}_{l^{(a)}}$.

Depending on the type of the attack (univariate vs. multivariate) the sample points at $\mathcal{J}_M$ are first combined using a combining function, e.g., centered product, split into the subsets depending the leakage model $\widetilde{\mathsf{L}}(.)$ and then used to estimate the statistical moments of a given order d. Depending on the order of the attack, prior preprocessing is also necessary. We denote these moments as the model by

$$\forall l^{(a)} \in \mathcal{L},\; M_{l^{(a)}} \xleftarrow[d\text{th-order moment}]{\substack{\text{preprocessing,}\\ \text{(centralized/standardized)}}} \{\boldsymbol{t}_i^{(M)},\, i \in \mathcal{I}^{(M)}_{l^{(a)}}, \mathcal{J}_M\}. \tag{26}$$

On the other hand, the traces at the sample points $\mathcal{J}_t$ need also to be preprocessed according to the variate of the attack (univariate vs. multivariate) as well as the given order d.

The correlation between the moments $M_{l^{(a\in\{1,\ldots,n_{\mathcal{L}}\})}}$ and the preprocessed traces $t'_{i\in\{1,\ldots,n\}}$ is defined as

$$\rho = \frac{\dfrac{1}{n}\sum\limits_{i=1}^{n}(t'_i - \mu_{t'})(M_{l_i} - \mu_M)}{\sqrt{\dfrac{1}{n}\sum\limits_{i=1}^{n}(t'_i - \mu_{t'})^2 \dfrac{1}{n}\sum\limits_{i=1}^{n}(M_{l_i} - \mu_M)^2}}, \tag{27}$$

where $M_{l_{i\in\{1,\ldots,n\}}} = M_{l^{(a)}}$, $l^{(a)} = \widetilde{\mathsf{L}}(v_i^{(t)}) \in \mathcal{L}$.

5.1 Numerator

To compute the numerator of Equation (27) it is first simplified to

$$\frac{1}{n}\sum_{i=1}^{n}(t'_i - \mu_{t'})(M_{l_i} - \mu_M) = \sum_{a=1}^{n_{\mathcal{L}}}(M_{l^{(a)}} - \mu_M)\,\frac{1}{n}\sum_{i\in\mathcal{I}^{(t)}_{l^{(a)}}} t'_i. \tag{28}$$

The preprocessing of the MC-DPA requires the sum of Equation (28) $SUM_{\mathcal{I}_{l^{(a)}}^{(t)}} = \sum_{i \in \mathcal{I}_{l^{(a)}}^{(t)}} t'_i$ to be processed independently. Otherwise, it is not trivially possible to provide iterative formulas as the mean and variance of subgroup of the traces $\in \mathcal{I}_{l^{(a)}}^{(t)}$ change. Since $n_{\mathcal{L}}$ is limited, we store a sum for each value of set $\mathcal{L}$ and merge them only at the end when the value of the estimated correlation is desired. In the multivariate higher-order $d > 1$ scenario, we store $n_{\mathcal{L}}$ sums of the traces as

$$SUM_{\mathcal{I}_{l^{(a)}}^{(t)}} = \sum_{i \in \mathcal{I}_{l^{(a)}}^{(t)}} t'_i = \sum_{i \in \mathcal{I}_{l^{(a)}}^{(t)}} \prod_{j \in \mathcal{J}_t} \left(t_i^{(j)} - \mu_{\mathcal{I}_{l^{(a)}}^{(t)}}^{(j)} \right) = SCP_{d, \mathcal{I}_{l^{(a)}}^{(t)}, \mathcal{J}_t}, \tag{29}$$

and in case of the univariate higher-order $d > 2$ as

$$SUM_{\mathcal{I}_{l^{(a)}}^{(t)}} = \sum_{i \in \mathcal{I}_{l^{(a)}}^{(t)}} t'_i = \frac{1}{\left(s_{\mathcal{I}_{l^{(a)}}^{(t)}}\right)^d} \sum_{i \in \mathcal{I}_{l^{(a)}}^{(t)}} \left(t_i - \mu_{\mathcal{I}_{l^{(i)}}^{(t)}} \right)^d = \frac{1}{\left(s_{\mathcal{I}_{l^{(i)}}^{(t)}}\right)^d} CS_{d, \mathcal{I}_{l^{(i)}}^{(t)}} \,. \tag{30}$$

Note that for $d = 2$ the denominator of Equation (30) is omitted. For a univariate first-order attack the means are used to derive the latter term of Equation (28) as

$$\frac{1}{n} SUM_{\mathcal{I}_{l^{(a)}}^{(t)}} = \frac{1}{n} \sum_{i \in \mathcal{I}_{l^{(a)}}^{(t)}} t_i = \frac{n_{l^{(a)}}^{(t)}}{n} \mu_{\mathcal{I}_{l^{(a)}}^{(t)}} \,. \tag{31}$$

We should here emphasize that – in contrast to the methods of the prior sections – in case of MC-DPA when a new trace is added to the set of traces following the incremental formulas only the sum and the moments which correspond to the leakage value $l^{(a)}$ related to the new trace are updated.

In order to calculate the whole numerator it is necessary to store the moments $M_{l^{(a)}}, \forall l^{(a)} \in \mathcal{L}$. This procedure is similar to before, and for the multivariate higher-order case it can be done by computing

$$M_{l^{(a)}} = \frac{1}{n_{l^{(a)}}^{(M)}} \sum_{i \in \mathcal{I}_{l^{(a)}}^{(M)}} \prod_{j \in \mathcal{J}_M} \left(t_i^{(j)} - \mu_{\mathcal{I}_{l^{(a)}}^{(M)}}^{(j)} \right) = \frac{SCP_{d, \mathcal{I}_{l^{(a)}}^{(M)}, \mathcal{J}_M}}{n_{l^{(a)}}^{(M)}}. \tag{32}$$

For the univariate case Equation (32) changes analog Equation (30). In a univariate first-order attack there is no preprocessing, and $M_{l^{(a)}}$ simply represents the mean $\mu_{\mathcal{I}_{l^{(a)}}^{(M)}}$.

The mean μ_M in Equation (27) is

$$\mu_M = \frac{1}{n} \sum_{a=1}^{n_{\mathcal{L}}} n_{l^{(a)}}^{(t)} \, M_{l^{(a)}}, \tag{33}$$

and as an example in case of a multivariate higher-order attack can be written as

$$\mu_M = \frac{1}{n}\sum_{a=1}^{n_{\mathcal{L}}} SCP_{d,\mathcal{I}^{(t)}_{l^{(i)}},\mathcal{J}_M}. \tag{34}$$

Since the iterative formulas (for both pair-wise and incremental cases) to compute $SCP_{d,\ldots}$ and $CS_{d,\ldots}$ as well as other necessary moments are given in previous sections, the numerator of Equation (27) can be easily derived.

5.2 Denominator

The first part of the denominator can be written as

$$\frac{1}{n}\sum_{i=1}^{n}\left(t'_i - \mu_{t'}\right)^2 = \frac{1}{n}\sum_{i=1}^{n} t'^2_i - \left(\mu_{t'}\right)^2 = \frac{1}{n}\sum_{a=1}^{n_{\mathcal{L}}}\left(\sum_{i\in\mathcal{I}^{(t)}_{l^{(a)}}} t'^2_i\right) - \left(\mu_{t'}\right)^2. \tag{35}$$

Therefore, we additionally need to compute the sums of the squared preprocessed traces $SUM^2_{\mathcal{I}^{(t)}_{l^{(a)}}} = \sum_{i\in\mathcal{I}^{(t)}_{l^{(a)}}} t'^2_i$. For a multivariate higher-order case, this can be written as $SCP_{2d,\mathcal{I}^{(t)}_{l^{(a)}},\{\mathcal{J}_t,\mathcal{J}_t\}}$ similar to Equation (29) or similar to Equation (30) and Equation (31) for the univariate cases. Further, the sums $SUM_{\mathcal{I}^{(t)}_{l^{(a)}}}$ computed by Equation (29), Equation (30), or Equation (31) can be used to derive $\mu_{t'}$ following the same principle of Equation (33).

The second part of the denominator of Equation (27) can be obtained from the values that are already used to compute the numerator:

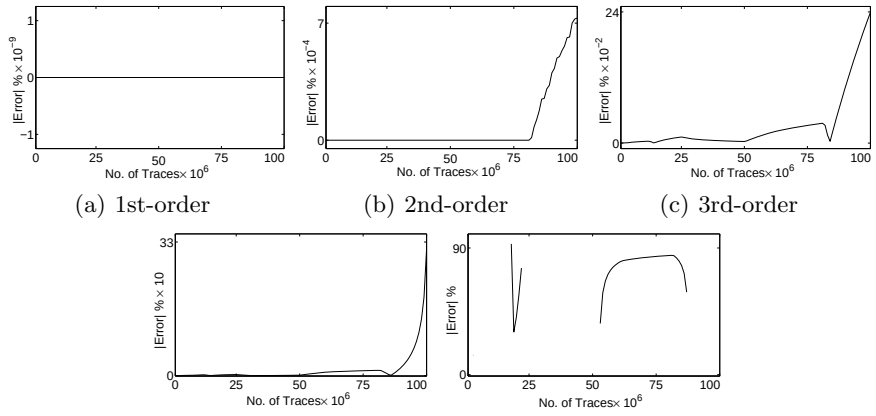
$$\frac{1}{n}\sum_{i=1}^{n}\left(M_{l_i} - \mu_M\right)^2 = \frac{1}{n}\sum_{a=1}^{n_{\mathcal{L}}} n^{(t)}_{l^{(a)}}\left(M_{l^{(a)}} - \mu_M\right)^2. \tag{36}$$

Since $n_{\mathcal{L}}$ is limited, the above expression can be computed at the end when all traces are processed to estimate the correlation.

In the aforementioned approach the sums $SUM_{\mathcal{I}^{(t)}_{l^{(a)}}}$ are grouped based on the output of the leakage function, i.e., $l^{(a)}$, which is also key dependent. Hence, the traces have to be regrouped for each key candidate as well as for each selected leakage function $\widetilde{\mathsf{L}}(.)$.

6 Evaluation

We evaluate the accuracy (convergence) of our presented approaches, and compare it to the corresponding results of the raw-moment and three-pass approaches. To this end, we generate 100 million simulated leakages by $\sim\mathcal{N}(100+\mathrm{HW}(x),3)$, where x is drawn uniformly from $\{0,1\}^4$. Hence, the correlation between the leakages and $\mathrm{HW}(x)$ is estimated. Following the concept of higher-order attacks, the leakages are also preprocessed (up to fifth order) to allow an



(a) 1st-order (b) 2nd-order (c) 3rd-order

(d) 4th-order (e) 5th-order

Fig. 1. Difference between the result of correlation estimations (raw-moment versus three-pass)

emulation of a higher-order univariate CPA. Note that the performance results are still valid in the multivariate case given additional leakage points with a similar leakage structure and the normalized product as combination function. This can be easily seen as both type of attacks require the estimation of centralized values up to a power of $2d$ (with an additional standardization for univariate higher-order attacks). The results based on our incremental approaches are exactly the same to the three-pass ones, i.e., with absolute 0 difference. As [3] only includes the formulas for first-order and second-order bivariate CPA, we further had to derive the necessary formulas for the univariate correlation up to the fifth order. The formulas can be found in Appendix A.

With these formulas we computed the correlation up to the fifth order on an Intel Xeon X5670 using a single thread, and examined the differences with respect to the results of the three-pass approach. Figure 1 presents the corresponding results. As expected, in the first-order setting the results are exactly the same, but the differences start to be obvious at higher orders particularly for higher number of traces. It is noteworthy that in the cases where no difference is shown for the fifth-order correlation, one of the variances of the denominator in the raw-moment approach turned to a negative value which indicates the instability of such formulas. With respect to the execution time of each approach, although it depends on the optimization level of the underlying computer code, we report 43 s, 17.8 s, and 11.6 s for three-pass, our incremental, and raw-moment approach respectively to estimate all five correlations at the same time on 100 million leakage points. Obviously, the raw-moment approach is faster than the others due to its lower amount of computations compared to our incremental one.

Acknowledgment

The research in this work was supported in part by the DFG Research Training Group GRK 1817/1.

References

1. B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen. Higher-Order Threshold Implementations. In *ASIACRYPT 2014*, volume 8874 of *LNCS*, pages 326–343. Springer, 2014.
2. A. Bogdanov. Multiple-Differential Side-Channel Collision Attacks on AES. In *CHES 2008*, volume 5154 of *LNCS*, pages 30–44. Springer, 2008.
3. P. Bottinelli and J. W. Bos. Computational Aspects of Correlation Power Analysis. Cryptology ePrint Archive, Report 2015/260, 2015. `http://eprint.iacr.org/`.
4. E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In *CHES 2004*, volume 3156 of *LNCS*, pages 16–29. Springer, 2004.
5. S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *CRYPTO 1999*, volume 1666 of *LNCS*, pages 398–412. Springer, 1999.
6. A. Duc, S. Dziembowski, and S. Faust. Unifying Leakage Models: From Probing Attacks to Noisy Leakage. In *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 423–440. Springer, 2014.
7. A. Duc, S. Faust, and F. Standaert. Making Masking Security Proofs Concrete - Or How to Evaluate the Security of Any Leaking Device. In *EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 401–429. Springer, 2015.
8. F. Durvaux, F.-X. Standaert, N. Veyrat-Charvillon, J.-B. Mairy, and Y. Deville. Efficient Selection of Time Samples for Higher-Order DPA with Projection Pursuits. In *COSADE 2015*, volume 9064 of *LNCS*, pages 30–50. Springer, 2015.
9. G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi. A testing methodology for side channel resistance validation. In *NIST non-invasive attack testing workshop*, 2011. `http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf`.
10. N. J. Higham. *Accuracy and Stability of Numerical Algorithms (2. ed.)*. SIAM, 2002.
11. A. Moradi and V. Immler. Early Propagation and Imbalanced Routing, How to Diminish in FPGAs. In *CHES 2014*, volume 8731 of *LNCS*, pages 598–615. Springer, 2014.
12. A. Moradi, O. Mischke, and T. Eisenbarth. Correlation-Enhanced Power Analysis Collision Attack. In *CHES 2010*, volume 6225 of *LNCS*, pages 125–139. Springer, 2010.
13. A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang. Pushing the Limits: A Very Compact and a Threshold Implementation of AES. In *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 69–88. Springer, 2011.
14. A. Moradi and F. Standaert. Moments-Correlating DPA. Cryptology ePrint Archive, Report 2014/409, 2014. `http://eprint.iacr.org/`.
15. P. Pébay. Formulas for Robust, One-Pass Parallel Computation of Covariances and Arbitrary-Order Statistical Moments. *Sandia Report SAND2008-6212, Sandia National Laboratories*, 2008.
16. E. Prouff, M. Rivain, and R. Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.
17. J. R. Rao, P. Rohatgi, H. Scherzer, and S. Tinguely. Partitioning Attacks: Or How to Rapidly Clone Some GSM Cards. In *IEEE Symposium on Security and Privacy 2002*, pages 31–41. IEEE Computer Society, 2002.
18. O. Reparaz, B. Gierlichs, and I. Verbauwhede. Selecting Time Samples for Multivariate DPA Attacks. In *CHES 2012*, volume 7428 of *LNCS*, pages 155–174. Springer, 2012.

19. M. Rivain and E. Prouff. Provably Secure Higher-Order Masking of AES. In *CHES 2010*, volume 6225 of *LNCS*, pages 413–427. Springer, 2010.
20. T. Schneider and A. Moradi. Leakage Assessment Methodology - a clear roadmap for side-channel evaluations. In *CHES 2015*, volume 9293 of *LNCS*, pages 495–513. Springer, 2015.
21. F. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, and S. Mangard. The World Is Not Enough: Another Look on Second-Order DPA. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 112–129. Springer, 2010.
22. Y. Zhou, Y. Yu, F. Standaert, and J. Quisquater. On the Need of Physical Security for Small Embedded Devices: A Case Study with COMP128-1 Implementations in SIM Cards. In *Financial Cryptography 2013*, volume 7859 of *LNCS*, pages 230–238. Springer, 2013.

A Correlation from the Raw Moments

As [3] only includes the formulas for first-order and second-order bivariate CPA, we first transform the bivariate formulas to the univariate second-order case and extend the approach to higher orders. Recall that the correlation for the bivariate second-order attack is computed in [3] as

$$\rho = \frac{n\lambda_1 - \lambda_2 s_3}{\sqrt{n\lambda_3 - {\lambda_2}^2}\sqrt{ns_9 - {s_3}^2}}, \tag{37}$$

where n denotes the number of traces and $\lambda_{\{1,2,3\}}$ are derived from the sums $s_{\{1,\ldots,13\}}$.

For the univariate second-order correlation, some of these sums are equivalent. Therefore, in this special case it is possible to reduce the number of sums required to be computed. For that, we first denote the d-th order sums as

$$S_d^{(t)} = \sum_{i=1}^{n} t_i^d, \qquad S_d^{(l)} = \sum_{i=1}^{n} l_i^d, \qquad S_d^{(t,l)} = \sum_{i=1}^{n} t_i^d l \tag{38}$$

with $s_3 = S_1^{(l)}$ and $s_9 = S_2^{(l)}$. The remaining parameters are then derived as

$$\lambda_1 = S_2^{(t,l)} - 2\frac{S_1^{(t)} S_1^{(t,l)}}{n} + \frac{S_1^{(t)} S_1^{(t)} S_1^{(l)}}{n^2}, \quad \lambda_2 = S_2^{(t)} - \frac{S_1^{(t)} S_1^{(t)}}{n}, \tag{39}$$

$$\lambda_3 = S_4^{(t)} - 4\frac{S_1^{(t)} S_3^{(t)}}{n} + 6\frac{S_1^{(t)} S_1^{(t)} S_2^{(t)}}{n^2} - 3\frac{S_1^{(t)} S_1^{(t)} S_1^{(t)} S_1^{(t)}}{n^3}. \tag{40}$$

For the higher-order correlation the basic structure of Equation (37) stays the same, and only the formulas for $\lambda_{\{1,2,3\}}$ change. We provided all necessary formulas in the following subsections.

A.1 Third Order

$$\lambda_1 = S_3^{(t,l)} - 3\frac{S_1^{(t)} S_2^{(t,l)}}{n} + 3\frac{\left(S_1^{(t)}\right)^2 S_1^{(t,l)}}{n^2} - \frac{\left(S_1^{(t)}\right)^3 S_1^{(l)}}{n^3}, \tag{41}$$

$$\lambda_2 = S_3^{(t)} - 3\frac{S_1^{(t)} S_2^{(t)}}{n} + 2\frac{\left(S_1^{(t)}\right)^3}{n^2}, \tag{42}$$

$$\begin{aligned}\lambda_3 = &S_6^{(t)} - 6\frac{S_1^{(t)} S_5^{(t)}}{n} + 15\frac{\left(S_1^{(t)}\right)^2 S_4^{(t)}}{n^2} - 20\frac{\left(S_1^{(t)}\right)^3 S_3^{(t)}}{n^3} \\ &+ 15\frac{\left(S_1^{(t)}\right)^4 S_2^{(t)}}{n^4} - 5\frac{\left(S_1^{(t)}\right)^6}{n^5}\end{aligned} \tag{43}$$

A.2 Fourth Order

$$\lambda_1 = S_4^{(t,l)} - 4\frac{S_1^{(t)} S_3^{(t,l)}}{n} + 6\frac{\left(S_1^{(t)}\right)^2 S_2^{(t,l)}}{n^2} - 4\frac{\left(S_1^{(t)}\right)^3 S_1^{(t,l)}}{n^3} + \frac{\left(S_1^{(t)}\right)^4 S_1^{(l)}}{n^4}, \tag{44}$$

$$\lambda_2 = S_4^{(t)} - 4\frac{S_1^{(t)} S_3^{(t)}}{n} + 6\frac{\left(S_1^{(t)}\right)^2 S_2^{(t)}}{n^2} - 3\frac{\left(S_1^{(t)}\right)^4}{n^3}, \tag{45}$$

$$\begin{aligned}\lambda_3 = &S_8^{(t)} - 8\frac{S_1^{(t)} S_7^{(t)}}{n} + 28\frac{\left(S_1^{(t)}\right)^2 S_6^{(t)}}{n^2} - 56\frac{\left(S_1^{(t)}\right)^3 S_5^{(t)}}{n^3} \\ &+ 70\frac{\left(S_1^{(t)}\right)^4 S_4^{(t)}}{n^4} - 56\frac{\left(S_1^{(t)}\right)^5 S_3^{(t)}}{n^5} + 28\frac{\left(S_1^{(t)}\right)^6 S_2^{(t)}}{n^6} - 7\frac{\left(S_1^{(t)}\right)^8}{n^7}\end{aligned} \tag{46}$$

A.3 Fifth Order

$$\begin{aligned}\lambda_1 = &S_5^{(t,l)} - 5\frac{S_1^{(t)} S_4^{(t,l)}}{n} + 10\frac{\left(S_1^{(t)}\right)^2 S_3^{(t,l)}}{n^2} - 10\frac{\left(S_1^{(t)}\right)^3 S_2^{(t,l)}}{n^3} \\ &+ 5\frac{\left(S_1^{(t)}\right)^4 S_1^{(t,l)}}{n^4} - \frac{\left(S_1^{(t)}\right)^5 S_1^{(l)}}{n^5},\end{aligned} \tag{47}$$

$$\lambda_2 = S_5^{(t)} - 5\frac{S_1^{(t)} S_4^{(t)}}{n} + 10\frac{\left(S_1^{(t)}\right)^2 S_3^{(t)}}{n^2} - 10\frac{\left(S_1^{(t)}\right)^3 S_2^{(t)}}{n^3} + 4\frac{\left(S_1^{(t)}\right)^5}{n^4}, \tag{48}$$

$$\begin{aligned}\lambda_3 = &S_{10}^{(t)} - 10\frac{S_1^{(t)} S_9^{(t)}}{n} + 45\frac{\left(S_1^{(t)}\right)^2 S_8^{(t)}}{n^2} - 120\frac{\left(S_1^{(t)}\right)^3 S_7^{(t)}}{n^3} + 210\frac{\left(S_1^{(t)}\right)^4 S_6^{(t)}}{n^4} \\ &- 252\frac{\left(S_1^{(t)}\right)^5 S_5^{(t)}}{n^5} + 210\frac{\left(S_1^{(t)}\right)^6 S_4^{(t)}}{n^6} - 120\frac{\left(S_1^{(t)}\right)^7 S_3^{(t)}}{n^7} + 45\frac{\left(S_1^{(t)}\right)^8 S_2^{(t)}}{n^8} - 9\frac{\left(S_1^{(t)}\right)^{10}}{n^9}\end{aligned} \tag{49}$$