Moments-Correlating DPA

Amir Moradi
Ruhr-Universität Bochum, Germany
Horst Görtz Institute for IT Security
amir.moradi@rub.de

François-Xavier Standaert
Université catholique de Louvain, Belgium
ICTEAM/ELEN/Crypto Group
fstandae@uclouvain.be

ABSTRACT

We generalize correlation-enhanced power analysis collision attacks into moments-correlating DPA. The resulting distinguisher is applicable to the profiled and non-profiled (collision) settings and is able to exploit information lying in any statistical moment. It also benefits from a simple rule-of-thumb to estimate its data complexity. Experimental results show that such a tool allows answering with confidence to some important questions regarding the design of side-channel countermeasures (e.g. what is the most informative statistical moment in the leakages of a threshold implementation). We further argue that moments-correlating DPA is a natural candidate for leakage detection tests, enjoying the simplicity of correlation power analysis and advanced features for the evaluation of higher-order attacks with an easy-to-compute confidence level.

1. INTRODUCTION

Context. Correlation-Enhanced Power Analysis Collision Attacks (CEPACA) have been introduced at CHES 2010 [23]. Such distinguishers bring an interesting alternative in the side-channel analysis toolbox for two main reasons. First (and as any collision attack - see [37] and the following works), they trade the usual requirement of having a (sufficiently accurate) leakage model for the assumption that some of the operations performed in the target implementation leak according to a similar model. As a result, they may work without a precise understanding of this target implementation, which is typically useful in a non-profiled attack setting. Second, they naturally extend to advanced contexts, where the side-channel information to extract lies in higher-order statistical moments [20]. As a result, CEPACA have been applied in different scenarios, e.g. timing attacks [25] or implementations protected with different countermeasures [22, 24].

Yet, such attacks also suffer from some limitations. First (and as any collision attack), the "similar leakage model" requirement may not always be respected in practice, e.g. in the context of hardware implementations where the target operations are implemented with different physical resources (possibly affected by variability [35]), or even in software implementations, due to pipelining effects [12]. Second, CEPACA essentially correlate statistical moments estimated for different S-boxes. As a result (and if the similar leakage model requirement is fulfilled), the value of the correlation coefficient estimated for the correct hypothesis gradually reaches '1' whenever enough samples are used in the attack. While this is not a problem if the goal of the distinguisher is only to perform a key recovery, it implies that the value of this correlation coefficient is not informative regarding the complexity of the attacks. In other words, it cannot be used as an evaluation metric, which is in contrast with the "standard" use of Pearson's correlation coefficient in [3]. The latter one can indeed be used as a worst-case metric under certain conditions discussed in [17] (namely first-order DPA with a perfect leakage model, essentially) and anyway benefits from a simple rule-of-thumb to estimate its (not worst-case) data complexity otherwise.

Our contribution. In this paper, we contribute to these two issues by proposing a neat generalization of CEPACA, next denoted as Moments-Correlating DPA (MC-DPA). For this purpose, our starting observation is that the correlation of statistical moments used in CEPACA has a natural counterpart in the context of profiled attacks, where we can correlate the moments corresponding to a single S-box estimated twice: first during profiling and then "on-the-fly" during the attack. This brings us to distinguish between Moments-Correlating Profiled DPA (MCP-DPA) which exploits this observation, and Moments-Correlating Collision DPA (MCC-DPA), which works as in previous works, by correlating the moments corresponding to two target operations estimated on-the-fly. Our second observation is that one can tweak CEPACA in order to preserve the metric aspect of Pearson's correlation coefficient as exploited in standard DPA. For this purpose, we simply have to replace the correlation of "moments with moments" by the correlation of "moments with samples". That is, considering the profiled estimation of the moments in MCP-DPA (resp. the on-the-fly estimation of these moments for one of the target operations in MCC-DPA) as a model, and then correlate this model with samples obtained on-the-fly in the attack phase of MCP-DPA (resp. with samples corresponding to the other operation in MCC-DPA), possibly squared, cubed, ... in case of higher-order analyzes.

Do we care? MC-DPA can be viewed as a variant of CEPACA, extended to the two main side-channel attack set-

Tis'16, October 24 2016, Vienna, Austria
© 2016 ACM. ISBN 978-1-4503-4575-0/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2996366.2996369

tings (namely profiled and non-profiled). Before entering the details of its instantiation, we want to shortly motivate why such a variant brings interesting insights for security evaluations, and solves open problems. We will take the application of CEPACA to the PRESENT threshold implementation of [31] to illustrate our claims. In [20], CEPACA is used to argue that this threshold implementation is "first-order secure" as expected by the proofs in [29]. But this argument still depends on the similar leakage model requirement. The application of MCP-DPA allows getting rid of this assumption. Hence, by comparing MCP-DPA with MCC-DPA, we can also quantify how much information is lost if the target operations of a collision attack do not leak according to similar models. Next, an important question regarding threshold implementations relates to the most informative statistical moment in their leakage distributions. Being glitch-free, there should be no information in the first-order moments. But it is unclear whether the best adversarial strategy is to focus on second- or third-order moments.[1] As will be shown in Section 3, answering this question with the information theoretic analysis proposed in [39] turns out to be uneasy. The metric feature of MC-DPA directly brings a simple answer, with the most informative moment leading to a higher correlation. Admittedly, an alternative solution to this problem can be found in the work of Bilgin et al. [1], where (standard DPA and collision-based) attacks of different orders were launched against a threshold implementation of AES. But this leads to the last interesting property of MC-DPA. Namely, security evaluations against side-channel attacks should ideally be based on the repetition of multiple experiments to gain statistical confidence. Yet, the estimation of (e.g.) a success rate in this context can become too expensive when the attacks' data complexity increases (e.g. neither [1] nor [26] computed such a success rate for their $> 1,000,000$-trace attacks). Thanks to the metric feature of MC-DPA, we can also use the rule-of-thumb proposed in [16, 38] to approximate the data complexity based on the squared inverse of Pearson's correlation estimated for a single attack (i.e. with much less sampling than by direct estimation).

Wrapping up. MC-DPA brings an interesting complement to the existing literature on side-channel distinguishers, by extending the applicability of CEPACA to profiled attacks and evaluation metrics. As a result, we obtain an easy to manipulate and interpret tool, that directly applies to higher orders and for which we can estimate the data complexity with limited sampling (without having to compute a success rate explicitly), i.e. some of the reasons that have made Correlation Power Analysis (CPA) so popular. In the following, we illustrate these useful qualities by confirming previous results on threshold implementations. Namely, we show that the claim of first-order security that was obtained using CEPACA in [20] extends to a profiled evaluation (with a limited loss of information, hence confirming that the similar leakage model requirement reasonably holds in this case). We also observe that attacks focusing on second-order moments are more efficient than attacks focusing on third-order ones for this implementation (as already found in [1]), and provide a more confident and quantitative analysis of this fact, thanks to the metric feature of MC-DPA. We then conclude by showing that MC-DPA is a promising candidate for efficient leakage detection tests [19], with easy-to-compute confidence level indicators.

Quite naturally, the following discussion also has strong connections with first- and higher-order CPA. In particular, MCP-DPA can be viewed as a profiled higher-order CPA (see, e.g. [33]). In this sense, our work should be viewed as a consolidating one, bridging the gap between the use of Pearson's correlation in various relevant scenarios (namely non-profiled with a-priori models as in [3], non-profiled collision-based as in [23], profiled and higher-order).

2. MOMENTS-CORRELATING DPA

Notations. We illustrate the attack with the key addition and S-box operations found in most block ciphers. For this purpose, let us denote a plaintext byte as x, a key byte as k, a key addition as $y = x \oplus k$, the execution of a b-bit S-box S as $z = \mathsf{S}(x \oplus k)$, and the leakage trace generated by this S-box computation as $z = \mathsf{S}(x \oplus k) \rightsquigarrow l_z$. We further use $\mathsf{E}(.)$ for the expectation operator. MC-DPA makes use of statistical moments that we specify as follows. Let X be a (univariate) random variable. The dth-order raw statistical moments are defined as $M_x^d = \mathsf{E}(X^d)$, with $\mu_x = \mathsf{E}(X)$ the mean. The dth-order central moments are defined as $CM_x^d = \mathsf{E}\left((X-\mu)^d\right)$, with $\sigma_x^2 = \mathsf{E}\left((X-\mu)^2\right)$ the variance. The dth-order standardized moments are defined as $SM_x^d = \mathsf{E}\left(\left(\frac{X-\mu}{\sigma}\right)^d\right)$, with $\gamma_x = \mathsf{E}\left(\left(\frac{X-\mu}{\sigma}\right)^3\right)$ the skewness and $\delta_x = \mathsf{E}\left(\left(\frac{X-\mu}{\sigma}\right)^4\right)$ the kurtosis. Eventually, we use $\rho(X,Y)$ for Pearson's correlation coefficient, add the hat operator for estimations.

2.1 Moments-Correlating Profiled DPA

Let $\mathbf{l}_{x,k}$ be an N-element vector of leakage traces corresponding to N intermediate values $z = \mathsf{S}(x \oplus k) \rightsquigarrow l_z$, e.g. $[l_0, l_{16}, l_{51}, \ldots]$, $\bar{\mu}_{x,k}$ be the N-element vector of the corresponding (estimated) mean values, e.g. $[\hat{\mu}_0, \hat{\mu}_{16}, \hat{\mu}_{51}, \ldots]$, $\bar{\sigma}^2_{x,k}$, $\bar{\gamma}_{x,k}$, $\bar{\delta}_{x,k}$ and $\mathbf{M}^d_{x,k}$, $\mathbf{CM}^d_{x,k}$, $\mathbf{SM}^d_{x,k}$ be similar vectors for the variance, skewness, kurtosis and dth-order (raw, central, standardized) moments. We denote the estimation of one of those vectors from an N_p-element vector of profiling leakage traces $\mathbf{l}^{\mathsf{p}}_{x,k}$ as (e.g. for the dth-order raw moments): $\hat{\mathbf{M}}^d_{x,k} \leftarrow \mathbf{l}^{\mathsf{p}}_{x,k}$. MCP-DPA will select the key candidate according to (again for the raw moments):

$$\tilde{k} = \underset{k^*}{\text{argmax}}\ \hat{\rho}(\hat{\mathbf{M}}^d_{x,k^*}, (\mathbf{l}^{\mathsf{t}}_{x,k})^d),$$

where $\hat{\mathbf{M}}^d_{x,k^*}$ is the dth-order (estimated) statistical moment vector permuted according to a key hypothesis k^*, and $\mathbf{l}^{\mathsf{t}}_{x,k}$ is an N_t-element vector of test traces. If a central or standardized moment is used, the second argument in the correlation coefficient will be replaced by $(\mathbf{l}^{\mathsf{t}}_{x,k} - \bar{\mu}_{x,k})^d$ and $(\frac{\mathbf{l}^{\mathsf{t}}_{x,k} - \bar{\mu}_{x,k}}{\bar{\sigma}_{x,k}})^d$, respectively (which is then similar to [33]).

2.2 Moments-Correlating Collision DPA

The previous attack requires a profiling step to estimate the 2^b statistical moments corresponding to the leakage of the target intermediate values $z = \mathsf{S}(x \oplus k)$. In a non-profiled scenario, an alternative is to target a pair of S-box

[1] Especially in the context of hardware implementations manipulating the three shares in parallel as we consider next, since finding the points-of-interest will be equally difficult (in terms of time complexity) for second- and third-order attacks in this case.

computations, e.g. $z_0 = \mathsf{S}(x_0 \oplus k_0) \rightsquigarrow l_{z_0}$ and $z_1 = \mathsf{S}(x_1 \oplus k_1) \rightsquigarrow l_{z_1}$, to estimate these 2^b moments for the first S-box "on-the-fly", i.e. $\hat{\mathbf{M}}^d_{x_0,k_0} \leftarrow \mathbf{l}^{\mathrm{t}}_{x_0,k_0}$, and to correlate the moment vector with the leakage samples corresponding to the second S-box permuted according to a value Δ added to the key, i.e. $\mathbf{l}^{\mathrm{t}}_{x_1,k_1\oplus\Delta}$. MCC-DPA will select the value of Δ according to (again for the raw moments):

$$\tilde{\Delta} = \underset{\Delta}{\mathrm{argmax}}\ \hat{\rho}(\hat{\mathbf{M}}^d_{x_0,k_0}, (\mathbf{l}^{\mathrm{t}}_{x_1,k_1\oplus\Delta})^d).$$

As mentioned in introduction, this attack can be viewed as a tweaked CEPACA, where the adversary would compute the correlation between two vectors of statistical moments, i.e. $\hat{\rho}(\hat{\mathbf{M}}^d_{x_0,k_0}, \hat{\mathbf{M}}^d_{x_1,k_1\oplus\Delta})$. The main advantage of this tweak is that while the value of the correlation coefficient in CEPACA gradually tends to '1' when the number of test traces increases (if the S-boxes leak according to the same leakage function), it is now dependent on the "informativeness" of the statistical moment exploited in the attack (essentially because we correlate moments with samples, rather than moments with moments). As a result (and compared to CEPACA), the MCC-DPA described in this subsection additionally provides a metric to quantify the number of measurements needed to perform a key recovery with a given success rate (directly derived from [16, 17, 38]):

$$N_{\mathrm{sr}} = c \cdot \frac{1}{(\hat{\rho}(\hat{\mathbf{M}}^d_{x_0,k_0}, (\mathbf{l}^{\mathrm{t}}_{x_1,k_1\oplus\Delta})^d)^2}, \qquad (1)$$

where c is a constant that depends on the number of hypotheses in the attack (i.e. 2^b) and the target success rate.[2] A similar formula can be used for MCP-DPA. By running such tools for different orders d, cryptographic designers directly get insights about the origin of the weaknesses in their implementations.

3. SIMULATED EXPERIMENTS

One of the goals of MC-DPA is to provide an easy-to-manipulate tool for the detection of the most informative statistical moments, e.g. in threshold implementations. In this section, we take advantage of a simulated case-study to analyze this problem in a well-controlled environment, and detail why it is challenging. For this purpose, we will consider the following three types of leakage samples:

$$\begin{aligned} l_z^{\mathrm{u1}} &= \mathsf{HW}(z) + N, \\ l_z^{\mathrm{m2}} &= \mathsf{HW}(z \oplus m) + \mathsf{HW}(m) + N, \\ l_z^{\mathrm{mf}} &= \mathsf{HW}(z \oplus m) + \mathsf{HW}(m) + f \times \mathsf{HW}(z) + N, \end{aligned}$$

where HW is the Hamming weight function and N is a Gaussian random noise with variance σ_n^2. l_z^{u1} typically corresponds to the (first-order) leakage of an unprotected implementation. l_z^{m2} typically corresponds to the (second-order) leakage of a masked implementation. l_z^{mf} typically correspond to the (first- and second-order) leakage of a masked implementation with a first-order flaw (e.g. due to glitches as in [18]). We additionally use a parameter f to capture the fact that this first-order flaw may have a smaller amplitude than the second-order signal. Note that we do not claim that this setting strictly corresponds to any physical implementation. We just use it to put forward intuitions regarding the most informative moments in side-channel attacks. In order to analyze these different leakage scenarios, we will perform the information theoretic analysis put forward in [39] and first applied to masked implementations in [40]. This implies computing the following mutual information metric:

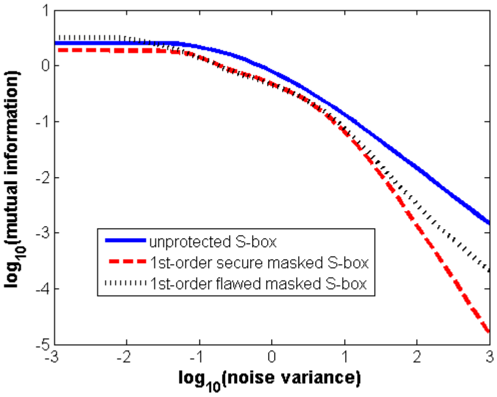


Figure 1: Information theoretic analysis of unprotected, masked and flawed masked implementation.

$$\begin{aligned} \mathrm{I}(K; L, X) \quad = \quad & \mathrm{H}[K] - \sum_{k\in\mathcal{K}} \Pr[k] \sum_{x\in\mathcal{X}} \cdot \Pr[x] \\ & \cdot \sum_{l\in\mathcal{L}} \Pr[l|k,x] \cdot \log_2 \Pr[k|x,l], \qquad (2) \end{aligned}$$

where L is the random variable corresponding to leakage samples l, that we replace by L_z^{u1}, L_z^{m2} or L_z^{mf} depending on whether we consider an unprotected, masked or flawed masked simulated implementation.[3] We ran this information theoretic analysis (for $b = 8$, $f = 0.2$) in function of the noise variance and report our results in Figure 1, from which we observe that:

1. All curves start by a plateau region, where the noise is small compared to the difference between the leakage values (here, Hamming weights).

2. As the noise increases, the slope of the curves reveals the security order of the implementations, e.g. -1 (resp. -2) for the unprotected (resp. masked) one.

3. For the masked S-box with first-order flaw, the information theoretic curve first follows the masked one, and then becomes parallel to the unprotected one for large noise levels. This indicates that second-order (resp. first-order) moments are more informative for low (resp. large) noise levels.

These experiments recall the fundamental masking equation "order of the statistical moment to estimate + measurement noise variance = security level", first hinted by Chari et al. [4]. While they indeed put forward that (depending on the noise level), one or another statistical moment may be

[2] This formula has been refined by Fei et al. [11] and Thillard et al. [41], at CHES 2012 and CHES 2013, respectively. We keep its older version for simplicity.

[3] In our simulated setting, we assume that the adversary's model exactly corresponds to the true leakage function. In the case of masked implementations, it implies summing over all the m's and computing $\Pr[k|x,l]$ as $\sum_m \Pr[l|x,l,m] \cdot \Pr[m]$. Hence this metric strictly corresponds to the (worst-case) mutual information (vs. the perceived information when this condition does not hold, as discussed in [35]).

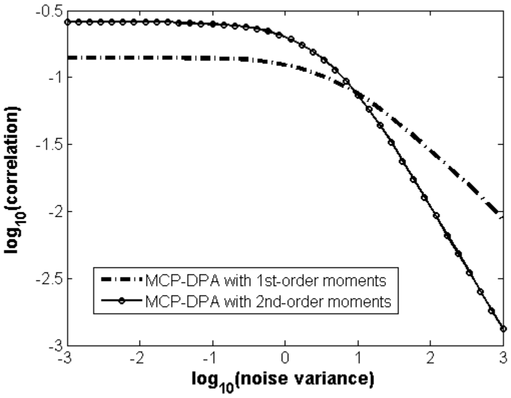


Figure 2: MCP-DPA metric for the flawed masked implementations.

more informative, such information theoretic curves are still limited in exhibiting exactly the noise threshold where the most informative moment changes. This is because an information theoretic analysis captures the worst-case adversary exploiting *all* the statistical moments jointly (i.e. the full distribution). As a result, the intuition regarding the orders is only revealed in the slopes of these curves. So while we can reasonably assume that the noise threshold we are looking for lies approximately where the dashed (red) and dotted (black) curves in the left part of Figure 1 separate (i.e. close to $\sigma_n^2 = 10^1$), a strict decision is hard to make here. Besides, and maybe more importantly, any analysis of an actual chip will be done for a single noise level (i.e. corresponds to a single point in the information theoretic curves), and therefore will not exhibit any slope. Interestingly, this is exactly where MC-DPA will come in handy. Indeed, by launching such attacks for different statistical moments, we can obtain intuition about their respective informativeness. For illustration, we launched such attacks against the flawed masked implementations, with first- and (central) second-order moments. The results in Figure 2 now clearly allow distinguishing for which noise level the first-order moments become more informative, and indeed confirm the previous intuitions (i.e. the threshold is close to $\sigma_n^2 = 10^1$). Furthermore, this information is obtained by comparing the correlation coefficient values for each noise level independently, so could be applied to an actual chip as well.

Summarizing, these simulated experiments show that MC-DPA provide an easy answer to the question of what is the most informative moment in a leaking implementation. To the best of our knowledge, it could not be obtained with previous distinguishers or metrics (the only known alternative would have been to compute success rates directly, i.e. a significantly more intensive task).

4. MEASURED EXPERIMENTS

For the practical experiments, we considered a threshold implementation of the PRESENT cipher [2]. Our design is the same as *Profile 2* in [31] which is based on a serialized architecture. Following the minimum settings of threshold implementations, all intermediate values of the cipher are represented by three Boolean shares, and we exploit the 2-stage masked S-box described in [29]. As described in Figure 3, the shared S-box input (y_1, y_2, y_3) – where $y = y_1 \oplus y_2 \oplus y_3$ – is first given to the G function, stored in the middle registers and then given to the F function which makes the shared S-box output as (z_1, z_2, z_3) – where $z_1 \oplus z_2 \oplus z_3 = z := \mathsf{S}(y)$. This process is repeated for the 16 S-boxes as represented on the figure, where the state is stored in shift registers which provide the S-box inputs nibble-by-nibble. We have also taken the same implementation platform as the one of [20] and [23], i.e. a Xilinx Virtex-II Pro FPGA embedded on SASEBO [28]. The leakage traces are collected by means of a LeCroy digital oscilloscope and a differential probe monitoring the voltage drop by a 1 Ω resistor placed at internal Vdd path of the target FPGA. The sampling rate was set to 1 GS/s and the target FPGA clock was driven at a frequency of 3 MHz.

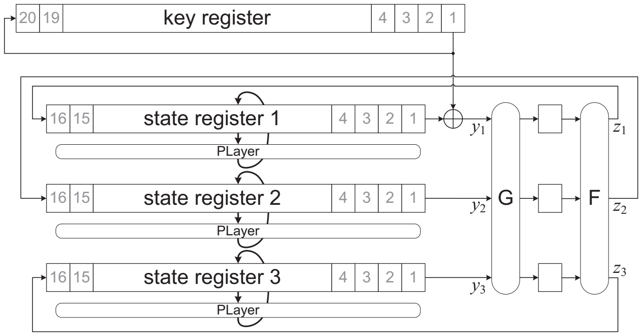


Figure 3: Architecture of the target design of PRESENT threshold implementation.

4.1 PRNG off

As a preliminary, we analyzed a setting where the PRNG was switched off (i.e. with all the masks stuck to '0' during the measurements). We expect first-order leakages to be detectable by both MCP-DPA and MCC-DPA in this case. For this purpose, we collected 100,000 traces, one of which is shown in the top part of Figure 4, where we can observe that each trace covers 6 clock cycles linked to the full computation of 5 S-boxes on 5 key-whitened plaintext nibbles.

Starting with MCP-DPA, we targeted the 7th nibble corresponding to plaintext x_7 and key k_7, and used $N_p = 50,000$ traces for profiling the first-order moments $\hat{\mathbf{M}}^1_{x_7,k_7}$. Since the PRESENT S-box is bijective and the initial key whitening is linear, by permuting the vector $\hat{\mathbf{M}}^1_{x_7,k_7}$ we obtain the estimated moments $\hat{\mathbf{M}}^1_{x_7,k_7^*}$ for all other possible key nibbles k_7^*. Then, we used another $N_t = 50,000$ traces to compute their correlation with the samples $\mathbf{l}^{\mathrm{t}}_{x_7,k_7}$, and applied this process to each time sample independently. Since $b = 4$ in this case, we obtained 16 correlation curves depicted in the middle part of Figure 4, where the curve corresponding to the correct key hypothesis is plotted in black.

A similar treatment was applied to MCC-DPA, for which we targeted the 7th and 8th nibbles. In this case, we estimated the moments for the first nibble, used the samples of the second one to compute the correlation, and targeted the difference $\Delta_{7,8} = k_7 \oplus k_8$. The corresponding S-box computations are computed consecutively in our implementation, with an interval of one clock cycle, i.e. 333 sample points.

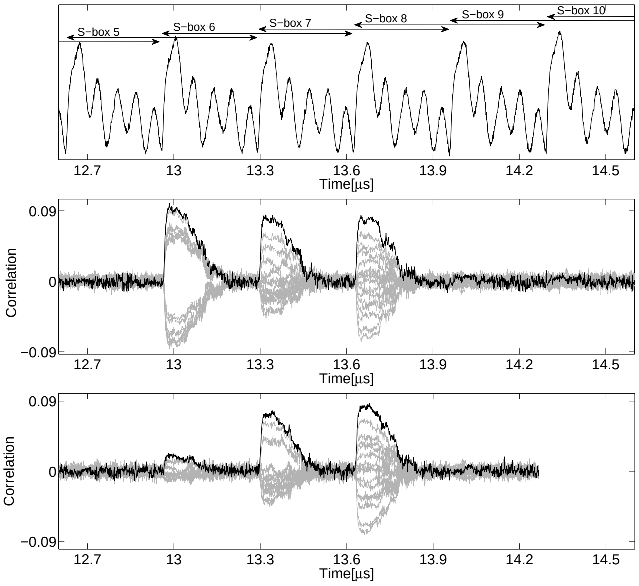


Figure 4: Experiments with PRNG off (N_p=N_t=50k traces). Top: leakage trace. Middle: MCP-DPA with first-order moments. Bottom: MCC-DPA with first-order moments.

Therefore, the leakage traces $\mathbf{l}^{\mathrm{t}}_{8,k_8}$ are shifted by 333 sample points to the left in order to be aligned with the estimated moments $\hat{\mathbf{M}}^1_{x_7,k_7}$. This time we used a single set of $50,000$ traces to estimate the moments $\hat{\mathbf{M}}^1_{x_7,k_7}$ and their correlation with $\mathbf{l}^{\mathrm{t}}_{x_8,k_8}$. By permuting $\mathbf{l}^{\mathrm{t}}_{x_8,k_8}$ according to $\Delta_{7,8}$ and estimating the correlation coefficient between $\hat{\mathbf{M}}^1_{x_7,k_7}$ and $\mathbf{l}^{\mathrm{t}}_{x_8,k_8\oplus\Delta_{7,8}}$ for each sample point independently, we obtained the results shown in bottom part of Figure 4.

These results lead to the following observations. First, both MCP-DPA and MCC-DPA are successful in recovering k_7 and $\Delta_{7,8}$, respectively. It indicates that (as expected) the threshold implementation has well identified first-order leakages when masks are stuck to '0'. Next, the leakages are spread over three clock cycles for MCP-DPA, which can be explained by the architecture of Figure 3. Namely, when the plaintext nibble x_7 appears at the last stage of the state shift register, it is XORed with the corresponding key nibble and given to the G function. This happens at the second clock cycle of the trace shown in the top of Figure 4. At the next clock cycle, the middle register stores $\mathsf{G}(x_7 \oplus k_7)$ which then appears at the input of the F function. Eventually, $\mathsf{F}(\mathsf{G}(x_7 \oplus k_7))$ is saved in the first stage of the state shift register during the third clock cycle, at the same time the F function's input changes to $\mathsf{G}(x_8 \oplus k_8)$. Since the leakages of sequential circuits usually depend on the transition between two consecutive states, observing some key dependencies in three states in a row is reasonable. Interestingly, we see that the situation slightly differs in the case of MCC-DPA. But this can also be explained by the target design. Namely, although there is a single S-box implemented, the key addition corresponding to plaintext nibbles x_7 and x_8 are not the same. Moreover, the controlling signals at the corresponding two clock cycles are not exactly the same either. The combination of these effects justifies the difference between the MCP-DPA and MCC-DPA curves.

One last question remains to investigate. That is, MCP-DPA theoretically allows to select different values for N_p and N_t. It leads to the problem of choosing these values adequately. We answered this question by running MCP-DPA and MCC-DPA in function of N_t and N_p according to two strategies.[4] In the *balanced* strategy, we always use $N_p = N_t$. In the unbalanced strategy, we set N_p to some arbitrary values. As illustrated in Appendix, Figures 6 and 7, the balanced strategy is usually sufficient to obtain well estimated moments (which was expected since the estimation of these moments is essentially what will be done in the online phase as well, by correlating them with samples raised to some power d). Besides, the figures confirm the rule-of-thumb for estimating the attacks data complexity (since the data complexities of MCP-DPA and MCC-DPA are similar, as the value of their correlation coefficient in Figure 4).

4.2 PRNG on

Quite naturally, the most interesting setting for threshold implementations is when the PRNG outputs uniformly random shares. This time we collected $20,000,000$ traces for our experiments to examine the efficiency of our proposed attacks. Similar to the case of PRNG off, we used the first half of the traces for moment estimation and the next $10,000,000$ traces for correlation estimation in MCP-DPA, and a single set of $10,000,000$ traces to perform both tasks in MCC-DPA. We started by running a first-order MCP-DPA for which the result is shown in the top part of Figure 5. As expected – theoretically proven by [29] and confirmed by [20] – the attack is not successful supporting the effectiveness of the threshold implementation scheme to prevent first-order leakages.

We then performed MCP-DPA and MCC-DPA with second-order central moments and third-order standardized moments. As illustrated on the figure, we observe that the investigated leakages contain information in both moments. Yet, a number of interesting additional observations can be made. First, and compared to the previous section, we see that MCP-DPA is slightly more efficient than MCC-DPA in exploiting them. While a precise reasoning about this fact seems uneasy, we conjecture that it relates to the similar leakage model requirement that becomes more sensitive as the order of the statistical moment exploited increases. Next, we can confirm that second-order moments are more informative, as previously reported in [1]. Compared to this reference, we gain a more quantitative statement about the respective informativeness of these moments, since the corresponding correlation coefficients obtained (e.g. for MCP-DPA) are respectively worth $4 \cdot 10^{-3}$ and $1.5 \cdot 10^{-3}$. This corresponds to a ratio between the data complexity of the corresponding attacks of approximatively $(\frac{4}{1.5})^2 \approx 7.11$. This result is also well in line with our simulated analyzes, since lower-order moments should always become more informative as the noise increases – and FPGAs are noisy platforms in general. Eventually, we see that only a single clock cycle leads to significant information when the PRNG is running, which means that the leakage related to the G function dominates in this case. We confirmed these observations by run-

[4] Of course, the experiment is artificial for MCC-DPA, since the same number of N_t traces is used for moment estimation and correlation estimation in this case. We just ran experiments with N_p traces for the moment estimation for completeness.

ning the attacks according to the balanced and unbalanced strategies in Appendix, Figures 8 to 11.

5. DISCUSSION: RECYCLING CEPACA TO GAIN CONFIDENCE

The previous sections put forward that MC-DPA is a natural extension of CEPACA, which can deal with profiled and non-profiled (collision-based) attack scenarios, and preserves the "metric" feature of CPA. By exploiting these potentialities, we could confirm previous results on threshold implementations and make our analyzes more precise. In this section, we want to conclude by arguing why MC-DPA consequently makes an interesting candidate for leakage detection tests, such as the T-test and MI-tests discussed in [19].

In this context, the usual tradeoff is between the efficiency and the genericity of the tests. For example, T-tests are primarily designed for the detection of univariate leakages in unprotected devices, and are extremely efficient in such cases. By contrast, MI-tests are able to capture more general dependencies, but are usually more data consuming. In this respect, we first observe that a moment-based approach as we suggest brings a possible compromise between these two solutions. Admittedly, T-tests could also be applied to squared, cubed, . . . traces – which makes them viable options to capture the leakage of protected implementations as well. Yet, MC-DPA brings two additional advantages, as we now detail. First, it naturally applies to the detection of multi-class leakages (while T-tests primarily focus on the two-class cases). Second, a fundamental question whenever performing a leakage detection is to determine whether the conclusions were obtained with sufficient confidence. A standard approach for answering this question is to perform cross-validation, which was recently suggested as an important part of leakage certification procedures [7]. Yet, this comes at the cost of additional data and time requirements. Interestingly, we show next that CEPACA can be recycled in order to efficiently gain some easy-to-compute confidence level. For this purpose, we just observe that while the value of the correlation coefficient produced by such attacks should gradually tend to '1' as the number of samples used in the attack/evaluation increases, the fact that this correlation is close to '1' indeed indicates that the estimations are confident. As a result, we can simply complement the previous MCP-DPA and MCC-DPA by computing the Moments against Moments Profiled Correlation (MMPC) and Moments against Moments Collision Correlation (MMCC), defined as follows:

$$\mathrm{MMPC} = \hat{\rho}(\hat{\mathbf{M}}^d_{x,k^*}, \hat{\mathbf{M}}^d_{x,k}),\ \mathrm{MMCC} = \hat{\rho}(\hat{\mathbf{M}}^d_{x_0,k_0}, \hat{\mathbf{M}}^d_{x_1,k_1\oplus\Delta}).$$

In the case of a collision-based attack, a low MMCC value could still indicate that the similar leakage model requirement is not fulfilled. But when applied to MCP-DPA, a large value of the MMPC criteria ensures that the estimates used in the evaluation are good. For illustration and in order to confirm our experiments, we computed this criteria for our MCP-DPA evaluations of Figure 5 and observed that both for the second- and third-order moments, it was larger than 90% (hence confirming that our evaluations were confident). In general, and since this criteria shares the same meaning for any order d, it could be set as a goal to reach if comparisons between different evaluations have to be performed.

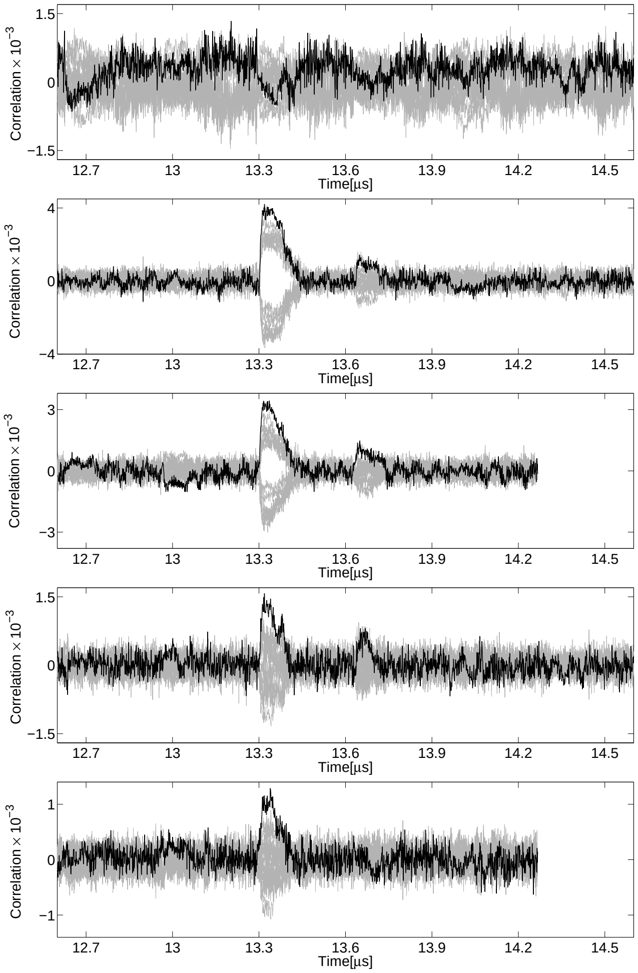


Figure 5: Experiments with PRNG on (N_p=N_t=10M traces). Top: MCP-DPA with first-order moments. Middle Up: MCP-DPA with 2nd-order central moments. Middle: MCC-DPA with 2nd-order central moments. Middle Down: MCP-DPA with 3rd-order standardized moments. Bottom: MCC-DPA with 3rd-order standardized moments.

Putting things together, we see that MC-DPA combines several advantages for side-channel leakage-detection and evaluation. Namely, it applies to any order d and can be implemented very efficiently, enjoying an intuitive rule-of-thumb to estimate the attacks data complexity with limited sampling. Furthermore, it directly comes with a way to quantify the confidence in the analyzes performed. Summarizing, it combines the advantages that have made CPA one of the most popular side-channel distinguishers and extends their applicability to new settings (namely profiled attacks, non-profiled attacks and leakage detection). Admittedly, its profiled version remains suboptimal compared to template attacks [5], but we believe it brings an interesting complement to such standard tools, either to be launched as a preliminary experiment, or in order to answer questions that template attacks cannot (e.g. the "most informative moment" question that we investigated for our threshold implementation in Section 4). Besides, MCP-DPA could be as efficient as template attacks in certain conditions (e.g. information lying in a single statistical moment). Note finally that the experiments in this work considered univariate side-channel attacks as a meaningful case-study. But the proposed tools could naturally be extended to the multivariate setting as well, by using mixed statistical moments [14].

6. FOLLOW UP WORKS

This work on MC-DPA has been online on ePrint for a couple of years already. To conclude this paper, we therefore list its applications that confirm its relevance. First from a methodological point of view, MC-DPA has been a building block to improve recent works on leakage detection/assessment [10, 9, 36] and for leakage certification [8] (to appear at CHES 2016). It is also a useful tool to discuss independence issues in masking proofs [6]. Second, from a practical point-of-view, it has been used in various concrete security evaluations of threshold implementations, but also other hardware countermeasures [21, 13, 27, 32].

Acknowledgements. F.-X. Standaert is a research associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in parts by the European Commission through the ERC project 280141 (CRASH) and by the ARC project NANOSEC.

7. REFERENCES

[1] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. A More Efficient AES Threshold Implementation. *IACR Cryptology ePrint Archive*, 2013:697, 2013.

[2] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[3] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[4] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.

[5] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.

[6] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 401–429. Springer, 2015.

[7] François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to certify the leakage of a chip? In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 459–476. Springer, 2014.

[8] François Durvaux and François-Xavier Standaert. Towards easy leakage certification. *IACR Cryptology ePrint Archive*, 2015:537, 2015.

[9] François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 240–262. Springer, 2016.

[10] François Durvaux, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Jean-Baptiste Mairy, and Yves Deville. Efficient selection of time samples for higher-order DPA with projection pursuits. In Stefan Mangard and Axel Y. Poschmann, editors, *Constructive Side-Channel Analysis and Secure Design - 6th International Workshop, COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers*, volume 9064 of *Lecture Notes in Computer Science*, pages 34–50. Springer, 2015.

[11] Yunsi Fei, Qiasi Luo, and A. Adam Ding. A Statistical Model for DPA with Novel Algorithmic Confusion Analysis. In Prouff and Schaumont [34], pages 233–250.

[12] Benoît Gérard and François-Xavier Standaert. Unified and Optimized Linear Collision Attacks and Their Application in a Non-profiled Setting. In Prouff and Schaumont [34], pages 175–192.

[13] Andreas Gornik, Amir Moradi, Jürgen Oehm, and Christof Paar. A hardware-based countermeasure to reduce side-channel leakage: Design, implementation, and evaluation. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 34(8):1308–1319, 2015.

[14] Vincent Grosso, François-Xavier Standaert, and Emmanuel Prouff. Low entropy masking schemes, revisited. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, volume 8419 of *Lecture Notes in Computer Science*, pages 33–43. Springer, 2013.

[15] Tim Güneysu and Helena Handschuh, editors. *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*. Springer, 2015.

[16] Stefan Mangard. Hardware Countermeasures against DPA ? A Statistical Analysis of Their Effectiveness. In Tatsuaki Okamoto, editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.

[17] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.

[18] Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.

[19] Luke Mather, Elisabeth Oswald, Joe Bandenburg, and Marcin Wójcik. Does My Device Leak Information? An a priori Statistical Power Analysis of Leakage Detection Tests. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (1)*, volume 8269 of *Lecture Notes in Computer Science*, pages 486–505. Springer, 2013.

[20] Amir Moradi. Statistical Tools Flavor Side-Channel Collision Attacks. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 428–445. Springer, 2012.

[21] Amir Moradi and Vincent Immler. Early propagation and imbalanced routing, how to diminish in fpgas. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 598–615. Springer, 2014.

[22] Amir Moradi and Oliver Mischke. How Far Should Theory Be from Practice? - Evaluation of a Countermeasure. In Prouff and Schaumont [34], pages 92–106.

[23] Amir Moradi, Oliver Mischke, and Thomas Eisenbarth. Correlation-Enhanced Power Analysis Collision Attack. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 125–139. Springer, 2010.

[24] Amir Moradi, Oliver Mischke, and Christof Paar. Practical evaluation of DPA countermeasures on reconfigurable hardware. In *HOST*, pages 154–160. IEEE Computer Society, 2011.

[25] Amir Moradi, Oliver Mischke, and Christof Paar. One Attack to Rule Them All: Collision Timing Attack versus 42 AES ASIC Cores. *IEEE Trans. Computers*, 62(9):1786–1798, 2013.

[26] Amir Moradi, Axel Poschmann, San Ling, Christof Paar, and Huaxiong Wang. Pushing the Limits: A Very Compact and a Threshold Implementation of AES. In Paterson [30], pages 69–88.

[27] Amir Moradi and Alexander Wild. Assessment of hiding the higher-order leakages in hardware - what are the achievements versus overheads? In Güneysu and Handschuh [15], pages 453–474.

[28] Morita Tech. Side-channel Attack Standard Evaluation Board (SASEBO). http://www.morita-tech.co.jp/SAKURA/en/index.html.

[29] 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.

[30] Kenneth G. Paterson, editor. *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*. Springer, 2011.

[31] Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-Channel Resistant Crypto for Less than 2,300 GE. *J. Cryptology*, 24(2):322–345, 2011.

[32] Santos Merino Del Pozo and François-Xavier Standaert. Blind source separation from single measurements using singular spectrum analysis. In Güneysu and Handschuh [15], pages 42–59.

[33] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.

[34] Emmanuel Prouff and Patrick Schaumont, editors. *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*. Springer, 2012.

[35] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In Paterson [30], pages 109–128.

[36] Tobias Schneider and Amir Moradi. Leakage assessment methodology - extended version. *J. Cryptographic Engineering*, 6(2):85–99, 2016.

[37] Kai Schramm, Thomas J. Wollinger, and Christof Paar. A New Class of Collision Attacks and Its Application to DES. In Thomas Johansson, editor, *FSE*, volume 2887 of *Lecture Notes in Computer Science*, pages 206–222. Springer, 2003.

[38] F-X Standaert, Eric Peeters, Gaël Rouvroy, and J-J Quisquater. An overview of power analysis attacks against field programmable gate arrays. *Proceedings of the IEEE*, 94(2):383–394, 2006.

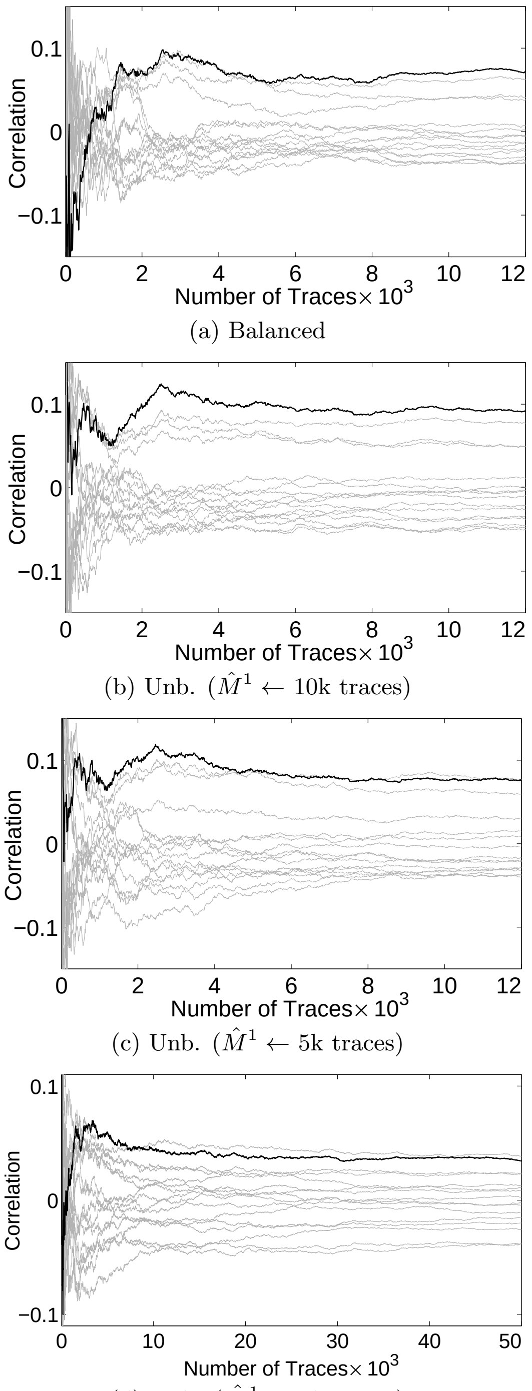
[39] François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

[40] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The World Is Not Enough: Another Look on Second-Order DPA. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.

[41] Adrian Thillard, Emmanuel Prouff, and Thomas Roche. Success through Confidence: Evaluating the Effectiveness of a Side-Channel Attack. In Guido Bertoni and Jean-Sébastien Coron, editors, *CHES*, volume 8086 of *Lecture Notes in Computer Science*, pages 21–36. Springer, 2013.

APPENDIX

A. ADDITIONAL FIGURES



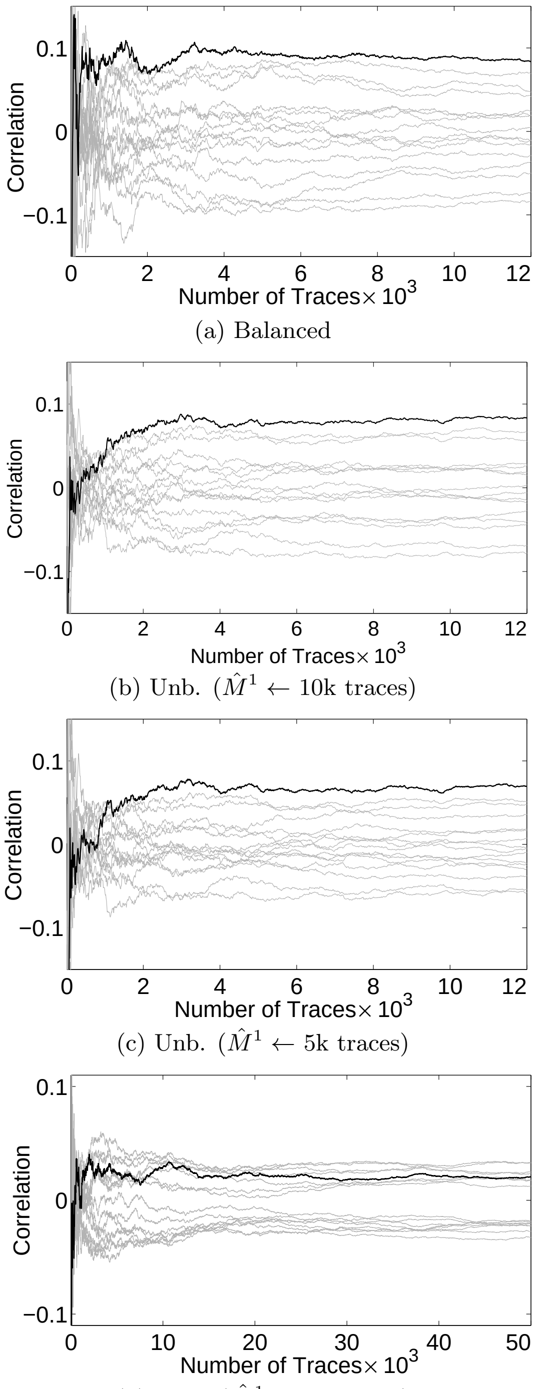
(a) Balanced

(b) Unb. ($\hat{M}^1 \leftarrow$ 10k traces)

(c) Unb. ($\hat{M}^1 \leftarrow$ 5k traces)

(d) Unb. ($\hat{M}^1 \leftarrow$ 1k traces)

Figure 6: PRNG off: MCP-DPA with first-order moments.



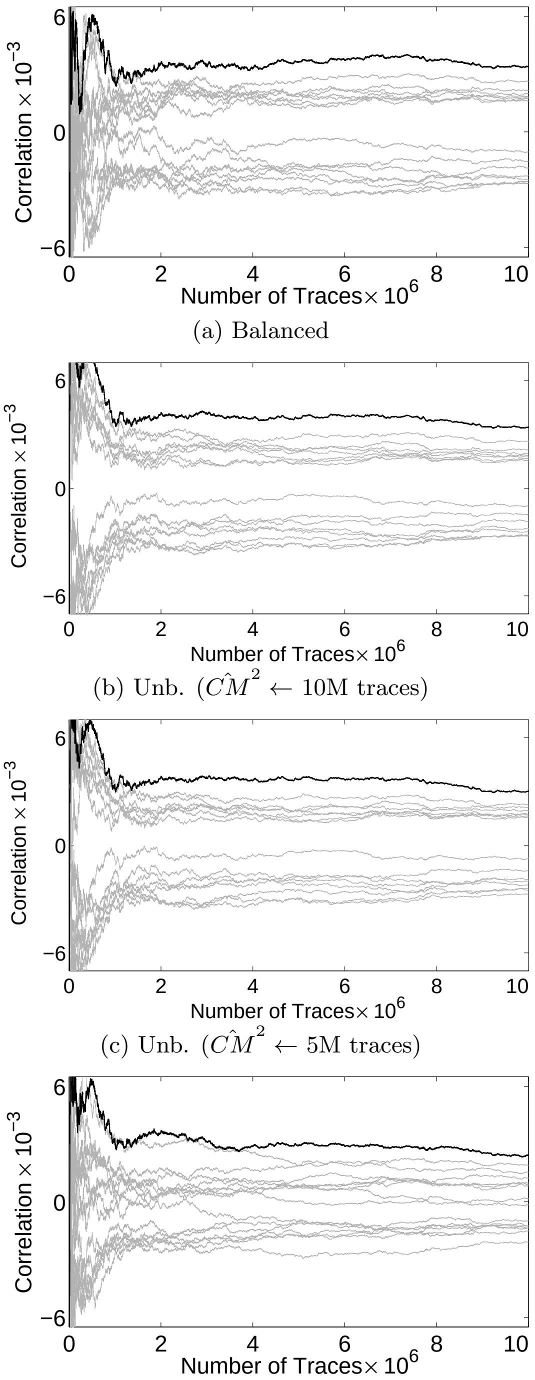
(a) Balanced

(b) Unb. ($\hat{M}^1 \leftarrow$ 10k traces)

(c) Unb. ($\hat{M}^1 \leftarrow$ 5k traces)

(d) Unb. ($\hat{M}^1 \leftarrow$ 1k traces)

Figure 7: PRNG off: MCC-DPA with first-order moments.



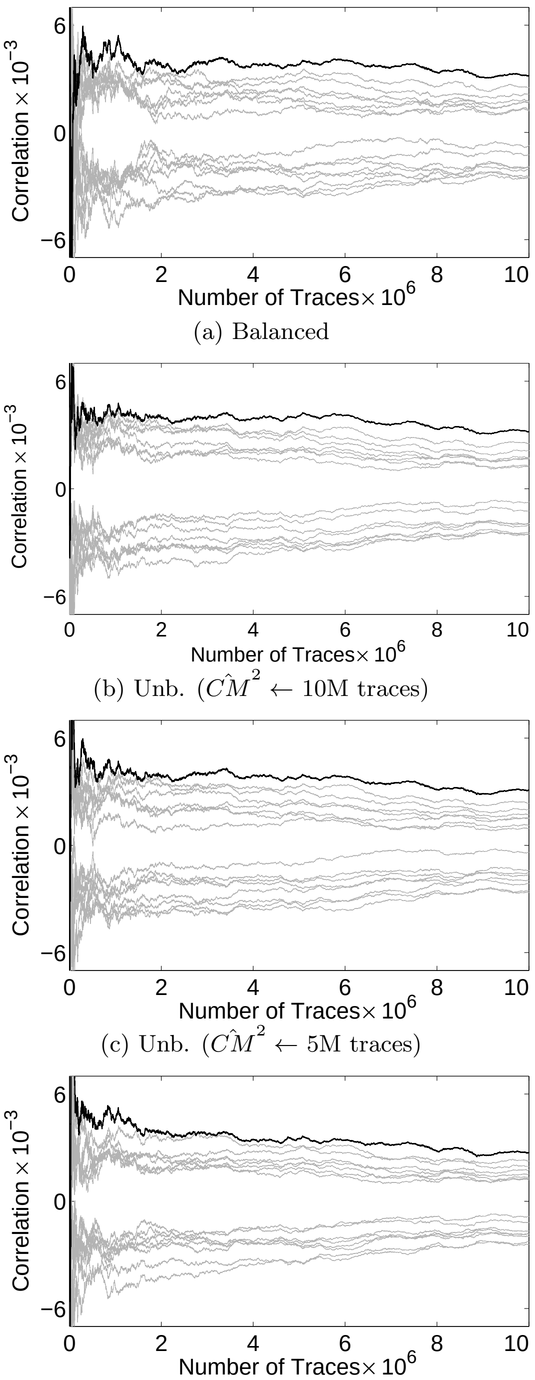
(a) Balanced

(b) Unb. ($\hat{CM}^2 \leftarrow$ 10M traces)

(c) Unb. ($\hat{CM}^2 \leftarrow$ 5M traces)

(d) Unb. ($\hat{CM}^2 \leftarrow$ 1M traces)

Figure 8: PRNG on: MCP-DPA with second-order central moments.



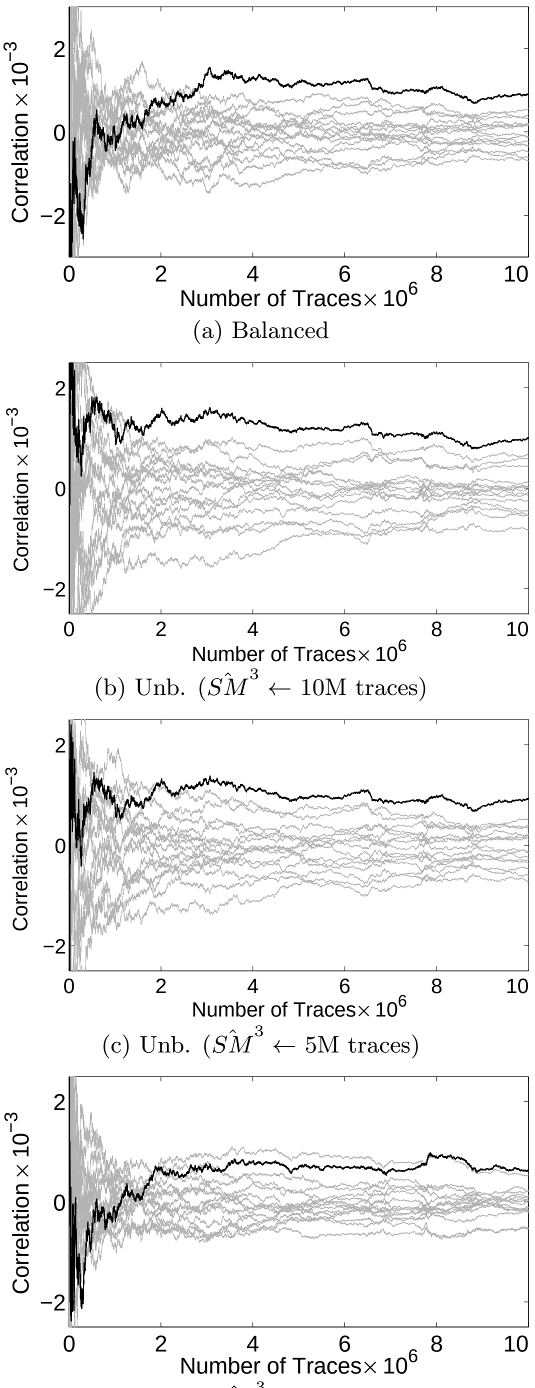
(a) Balanced

(b) Unb. ($\hat{CM}^2 \leftarrow$ 10M traces)

(c) Unb. ($\hat{CM}^2 \leftarrow$ 5M traces)

(d) Unb. ($\hat{CM}^2 \leftarrow$ 1M traces)

Figure 9: PRNG on: MCC-DPA with second-order central moments.



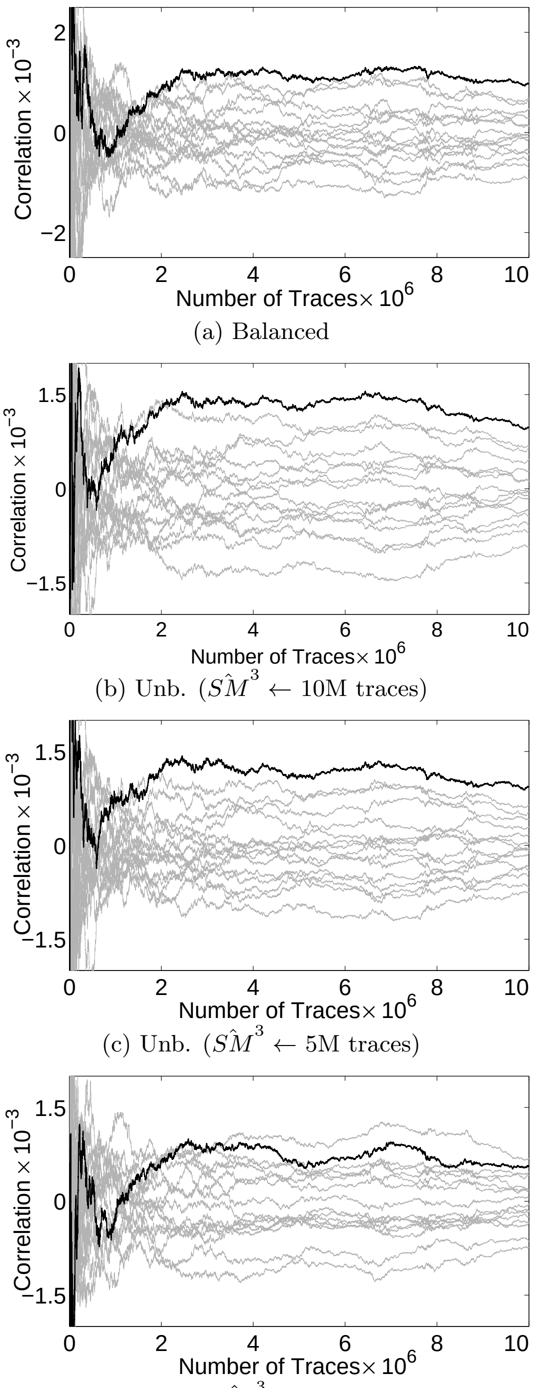
(a) Balanced

(b) Unb. ($S\hat{M}^3 \leftarrow$ 10M traces)

(c) Unb. ($S\hat{M}^3 \leftarrow$ 5M traces)

(d) Unb. ($S\hat{M}^3 \leftarrow$ 1M traces)

Figure 10: PRNG on: MCP-DPA with third-order normalized moments.



(a) Balanced

(b) Unb. ($S\hat{M}^3 \leftarrow$ 10M traces)

(c) Unb. ($S\hat{M}^3 \leftarrow$ 5M traces)

(d) Unb. ($S\hat{M}^3 \leftarrow$ 1M traces)

Figure 11: PRNG on: MCC-DPA with third-order normalized moments.