# On the Easiness of Turning Higher-Order Leakages into First-Order

Thorben Moos and Amir Moradi

Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Bochum, Germany
{firstname.lastname}@rub.de

**Abstract.** Applying random and uniform masks to the processed intermediate values of cryptographic algorithms is arguably the most common countermeasure to thwart side-channel analysis attacks. So-called masking schemes exist in various shapes but are mostly used to prevent side-channel leakages up to a certain statistical order. Thus, to learn any information about the key-involving computations a side-channel adversary has to estimate the higher-order statistical moments of the leakage distributions. However, the complexity of this approach increases exponentially with the statistical order to be estimated and the precision of the estimation suffers from an enormous sensitivity to the noise level. In this work we present an alternative procedure to exploit higher-order leakages which captivates by its simplicity and effectiveness. Our approach, which focuses on (but is not limited to) univariate leakages of hardware masking schemes, is based on categorizing the power traces according to the distribution of leakage points. In particular, at each sample point an individual subset of traces is considered to mount ordinary first-order attacks. We present the theoretical concept of our approach based on simulation traces and examine its efficiency on noisy real-world measurements taken from a first-order secure threshold implementation of the block cipher PRESENT-80, implemented on a 150nm CMOS ASIC prototype chip. Our analyses verify that the proposed technique is indeed a worthy alternative to conventional higher-order attacks and suggest that it might be able to relax the sensitivity of higher-order evaluations to the noise level.

## 1 Introduction

It has become a general knowledge that implementations of cryptographic algorithms are in danger of being attacked by means of side-channel analysis (SCA) key-recovery attacks, if dedicated countermeasures have not (or incorrectly) been integrated. Amongst the known and common SCA countermeasures, *masking* is by far the most-widely studied scheme and has interested both academia and industry. Its underlying sound proofs and theoretical foundation should be named among the reasons for such a popularity. Except particular constructions (e.g., [7, 12]), the security of masking schemes is based on the uniformity of the masks. More precisely, in an $(s+1)$-sharing construction, which is called $s$-order masking, for a particular $x$ each $(x_1, \ldots, x_{s+1})$ with $x = \bigoplus_{i=1}^{s+1} x_i$ should occur equally

likely[1]. Otherwise, it can be pretended that the randomness source is biased, which potentially leads to exploitable leakage.

With respect to the adversary model, security of masking schemes is evaluated based on two different models: *i*) probing model [10], and *ii*) bounded moment model [2]. The former one is primarily used for security proofs and more conservative than the later one, which is usually applied in practical evaluations. Our focus is mainly on the *bounded moment* model, and we call a device *without first-order leakage* if the leakages associated to two different given sets of operands $x$ and $y$ (of the same operation[2]) are not distinguishable[3] from each other through average, i.e., first-order statistical moment. Similarly the leakages should **further** not be distinguishable through variance, i.e., second-order centered moment, for second-order security, and likewise for higher orders. Optionally, the described setting can be incorporated by a pre-processing step, which combines different leakage points. Compared to *univariate* settings, where the combination of leakage points is not required, in a *multi-variate* scenario two (or more) different leakage points are combined prior to evaluation/attack (see [14] for more details).

In short, in order to attack an $s$-order masked implementation, multi-variate $(s+1)$-order statistical moments should be observed if the operations are serially performed on the shares (i.e., a typical software implementation with sequential nature). On the other hand, in case of a hardware implementation usually univariate $(s+1)$-order statistical moments are observed due to the inherent parallel processing fashion. It is noteworthy that the complexity of higher-order evaluations increases exponentially with $s$. Further, estimation of higher-order statistical moments becomes extremely hard in practice when the leakages are sufficiently noisy [22].

Instead of a conventional higher-order attack, we present in this work a trick that converts higher-order leakages to the first order and exploits them for key recovery. The focus of our scheme is **univariate** higher-order leakages, i.e., mainly targeting masked hardware implementations. It is essentially based on the principle of pruning the traces according to the distribution of leakage points. Its detailed expression is given in Section 3. Indeed, a similar approach has initially been considered in [24], to exploit the leakage of a masked dual-rail logic style (MDPL) [20]. We review the relevant state of the art in Section 2. Compared to a classical higher-order attack (e.g., mean-free square as an optimal second-order univariate attack) our scheme can be more efficient in particular cases. More precisely, it can exploit the leakage and recover the key while the classical higher-order attacks fail. As a case study, given in Section 4, we present practical results based on an ASIC prototype chip of a provably first-order secure threshold implementation (TI) [17] of the block cipher PRESENT [4].

[1] In case of Boolean masking.
[2] For example, two different plaintexts of an AES encryption with a fixed key.
[3] $t$-test can be used to detect the distinguishability [25].

## 2 State of the Art

For the majority of masking schemes it is a mandatory requirement that the masks are drawn from a uniform distribution. If this distribution is not uniform, but rather stems from a biased randomness source, vital security claims are not met and exploitable first-order leakage can emerge. Thus, an adversary might be interested in compromising the security of masked implementations deliberately by forcing a bias into the masks that conceal key-dependent intermediate values. One way of achieving this goal is to attack the randomness source directly by means of fault attacks. Of course, the feasibility of this approach depends highly on the particular implementation that is investigated. Another, more generic strategy, which has mainly been applied to compromise software-based masking schemes on microcontrollers, is to categorize the traces that are recorded in a power analysis attack into groups that only contain a biased subset of all possible masks. Intuitively, such an attack can be performed on a software-based masking scheme by determining a point in the power traces where the mask value is processed and then discarding all traces with a measured power consumption above (or below) a certain threshold at that sample point. Assuming now that the investigated device leaks information about the processed intermediate values by means of the Hamming weight (HW) model (which is a reasonable assumption for microcontrollers, see [13]), one has selected a subset of traces with a probability different from $\frac{1}{2}$ for each mask bit to be 1 (or 0). This allows a better-than-random guess what the mask value would be, e.g. all-one (or all-zero), which enables successful first-order attacks on the reduced set of traces. Hence, without preprocessing the power values in the traces, but only by ignoring a subset of the acquired measurements, one has moved the higher-order leakages to a setting where they can be exploited in the first order. Technically, due to the prior selection of power traces, this is still a higher-order attack, but in fact does not require the estimation of higher-order statistical moments. This kind of attack, which we extend and generalize for a different setting in the following course of this work, is referred to as biased mask attack, e.g. in [13] and [26]. Regardless of the surprisingly simple attack procedure, biased mask attacks have not gained much popularity since multi-variate higher-order attacks, utilizing the higher-order statistical moments of the *full* set of traces, are considered more powerful in the general case. Indeed, the loss of information due to disregarding a subset of the measurements is undeniable. Additionally, some kind of initial profiling has to be performed to find a sample point in the power traces where the mask value is leaked.

The described procedure can not be mapped directly to hardware implementations, because in parallel designs the mask is not processed discretely but usually together with the masked data and a number of further intermediate values at the same time. Consequently, only the cumulative leakage of mask and masked data can be observed in a univariate fashion and is not only buried in electronic noise, like for software implementations, but also in the switching noise originating from the remaining parts of the circuit (see [13]). On the one hand, due to the univariate nature of the leakages, the necessity for a profiling phase

is removed, but on the other hand the categorization of the traces based on the leakage of the mask value is much less precise. Nevertheless several attempts have been made to perform biased mask attacks on hardware implementations of gate- and algorithmic-level masking schemes. In [27], such an approach is considered for the first time. It is shown by toggle count simulations of a small test circuit (S-box + key XOR) that categorizing power traces with a simple threshold filter is sufficient to remove the one bit of entropy that is introduced by the use of the logic style Random Switching Logic (RSL). The affiliated work in [24] utilizes gate-level simulations of an AES chip design to show that routing imbalances in the DPA-resistant logic style MDPL [20] can be exploited to estimate the mask bit. Again, this can be used to remove the effect of the masking scheme by performing conventional first-order DPA attacks exclusively on the subset of traces that is obtained through a simple filtering operation. In [8] the authors extend their approach to an algorithmic-level hardware masking scheme for the first time. In accordance to the biased mask attacks on software-based implementations the authors are able to verify that a secure hardware masking scheme can equally be compromised by means of simple first-order distinguishers, when only a subset of the traces is considered. Unfortunately, the article fails to investigate how to select a suitable subset of traces that is most informative for an attack. Even more importantly it is not examined at all whether a first-order attack on their specific (or any other choice of) subset can outperform a univariate second-order attack using the mean-free square on the full set of traces. Finally, none of the listed works on hardware masking schemes verified the described attack procedures with practical measurements, taken from a physical hardware device. To the best of our knowledge, no subsequent work explores any of these data points either.

The last branch of research that can be considered related to our approach uses a subset of power traces to enhance the correlation in CPA [6] attacks in general, without concentrating on protected implementations or circumventing specific countermeasures in particular. These works, presented e.g., in [11] and [19], focus on selecting power traces with a high Signal-to-Noise ratio (SNR). They come to the conclusion that, considering the distribution of power values at the point of interest, especially those traces with a small probability density function value, have the highest SNR. In a simplified phrasing this means that concentrating on the power traces whose value at the point of interest is extraordinarily low or high (leftmost or rightmost slices of the leakage distribution) leads to the best correlation for the correct key candidate.

## 3 Underlying Approach

In this section we introduce and define our novel approach to exploit higher-order leakages. For the sake of simplicity, let us focus on a single sample point of side-channel leakages. The main idea is to observe the distribution of the **univariate** leakages, categorize them into e.g., two non-overlapping parts, and then perform the attack(s) on each part independently. This indeed is the same

concept which has been applied in [11] on *unprotected* implementations with the goal of improving the attacks with respect to the required number of traces (see Section 2). However, we employ more-or-less the same technique to exploit higher-order leakages. Let us express the underlying concept with simulation results. Suppose that the leakage of a device under test (DUT) can be represented by a noisy Hamming weight (HW) model as

$$l(x) = HW(x) + \mathcal{N}(\mu, \delta^2),$$

with mean $\mu = 0$ and standard deviation $\delta$. Further, suppose that the intermediate values of the DUT are masked following the concept of first-order Boolean masking. Hence, every value $x$ is represented by $(x_m, m)$ with $x_m = x \oplus m$ and $m$ being a random mask with uniform distribution. In a univariate setting, the leakage of the DUT associated to $x$ is represented by

$$l(x_m) + l(m) = HW(x_m) + HW(m) + \mathcal{N}(0, \delta^2).$$

If we simulate $1,000,000$ times the leakage for two different $x \in \{0,1\}^8$ values and a particular $\delta = 2$, two different distributions are observed, that are depicted in Figure 1(a). These two distributions are not distinguishable from each other through their means, i.e., a first-order distinguisher would not be able to differentiate them. Along the same lines, $t$ statistics of a Welch's $t$-test would give a low-confidence result as well, i.e., $t$ being smaller than 4.5.

However, if we consider only those leakages which are less than a threshold, see Figure 1(b), the leakages are distinguishable from each other through their means. For example, in this case the $t$ statistics yields the value 133, i.e., high confidence of a first-order distinguisher. The threshold in this example has been defined in such a way that 20% of the leakages are below the threshold and the remaining 80% above. As shown in Figure 1(c) to Figure 1(e), considering the upper 80%, lower 80% or upper 20% leakages would lead to distinguishability through means as well. However, in case of Figure 1(f) and Figure 1(g) when the middle part or the side parts of the distributions are considered, the mean does not reveal any distinguishability. This is indeed due to the symmetric form of the original distributions shown in Figure 1(a).

We should highlight that these observations are not limited to first-order masking. As an example, we repeated the same simulation under second-order Boolean masking with univariate leakage

$$l(x_m) + l(m_1) + l(m_2) = HW(x_m) + HW(m_1) + HW(m_2) + \mathcal{N}(0, \delta^2),$$

where $x_m = x \oplus m_1 \oplus m_2$ and the uniform distribution for $m_1$ and $m_2$. The distributions and the $t$ statistics as distinguishability measure after classifying the leakages based on a particular threshold are shown in Figure 2. Following the concept of second-order masking, the distributions are distinguishable only through their skewness (see Figure 2(a)). However, by categorizing them based on a 20% threshold (either above or below the threshold) the means reveal the difference between the distributions. Interestingly, the symmetric forms, i.e.,

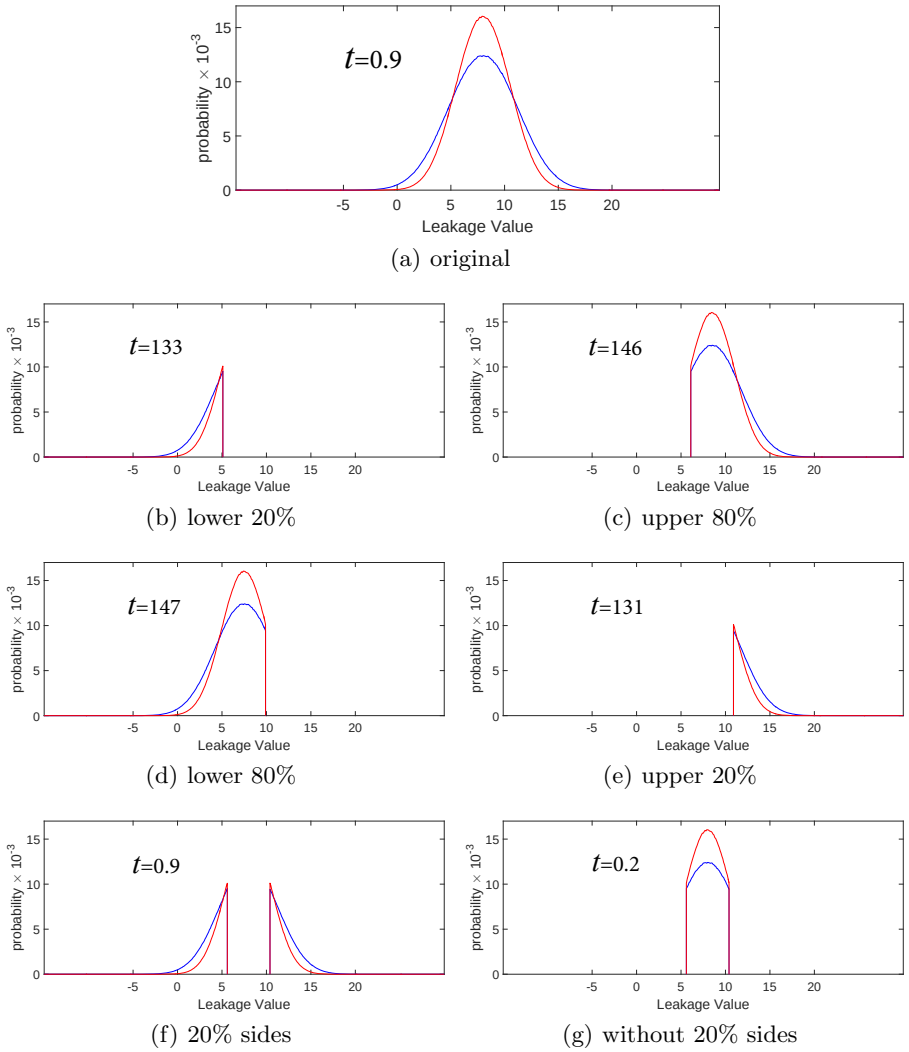
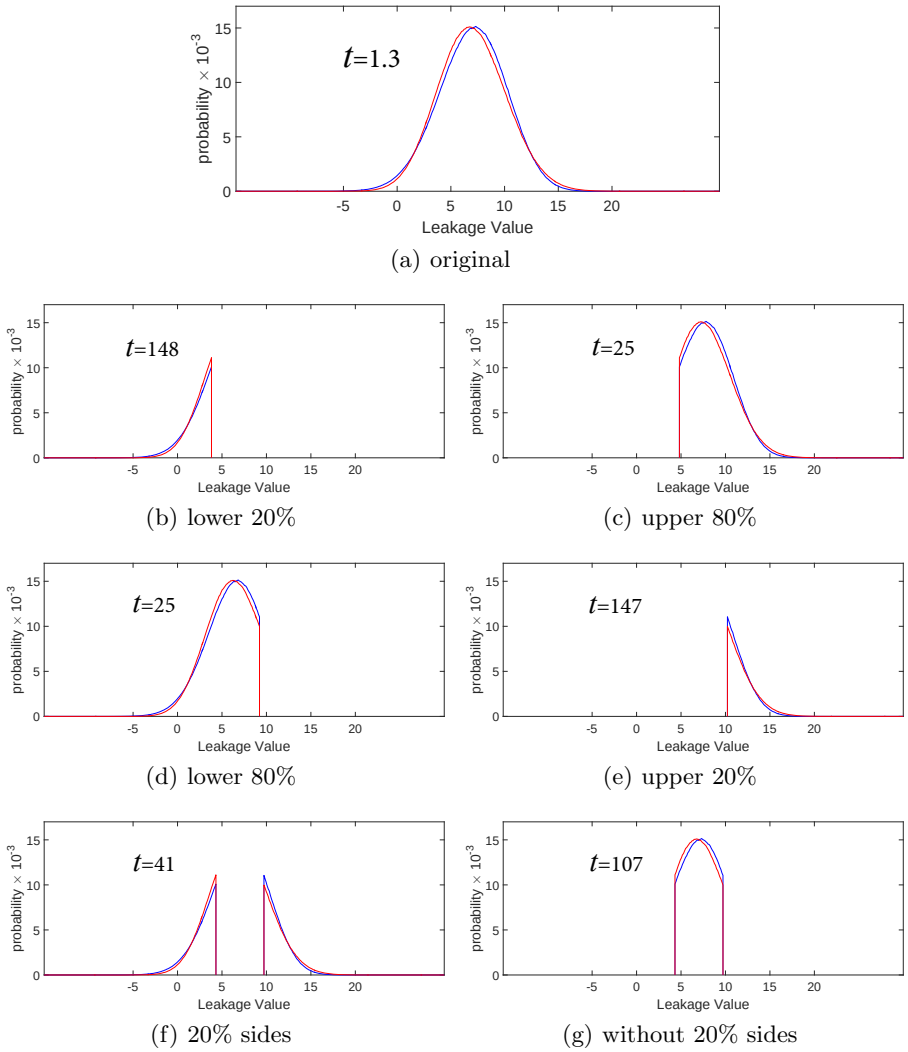



**Fig. 1.** Simulated leakage distributions of two different values represented by first-order masking, $t$ represents the statistics of the $t$-test.

middle part or the sides (Figure 2(f) and Figure 2(g)), also lead to high-evidence first-order distinguishability.

When evaluating the effectiveness of this approach it is important to know for which threshold value the attack performs best. To identify the optimal threshold, we conducted another simulation based on first-order masking. We have randomly selected a vector of $n$ elements as $X : (x^1, \ldots, x^n)$, where $x^i \in \{0,1\}^8$. Then, by two separate uniformly-distributed $n-$element mask vectors $M_1$ and $M_2$ we formed $X_{M_1} = (x^1_{m_1}, \ldots, x^n_{m_1})$, where $x^i_{m_1} = x^i \oplus m^i_1$ (resp. for




(a) original

(b) lower 20%

(c) upper 80%

(d) lower 80%

(e) upper 20%

(f) 20% sides

(g) without 20% sides

**Fig. 2.** Simulated leakage distributions of two different values represented by second-order masking, $t$ represents the statistics of the $t$-test.

$X_{M_2}$). Following the univariate noisy Hamming weight leakage model, we formed two leakage vectors $L_1 : (l_1^1, \ldots, l_1^n)$ and $L_2 : (l_2^1, \ldots, l_2^n)$ in such a way that for example

$$l_1^i = HW(x_{m_1}^i) + HW(m_1^i) + \mathcal{N}(0, \delta^2).$$

Following the concept of Moments-Correlating DPA (MC-DPA) [15], we first formed a model $\dot{L}_1 : (\dot{l}_1^1, \ldots, \dot{l}_1^n)$ as

$$\dot{l}_1^i = \mu\left(\{\forall l_1^j | x^j = x^i\}\right),$$

and finally estimated the correlation $\rho(\dot{L}_1, L_2)$ as the first-order correlation. For the second-order correlation, we first formed a model $\ddot{L}_1 : (\ddot{l}_1^1, \ldots, \ddot{l}_1^n)$ as

$$\ddot{l}_1^i = \delta^2 \left( \{ \forall l_1^j | x^j = x^i \} \right),$$

and respectively made $L_2'$ as mean-free square of $L_2$ as

$$l_2'^i = \left( l_2^i - \mu \left( \{ \forall l_2^j | x^j = x^i \} \right) \right)^2.$$

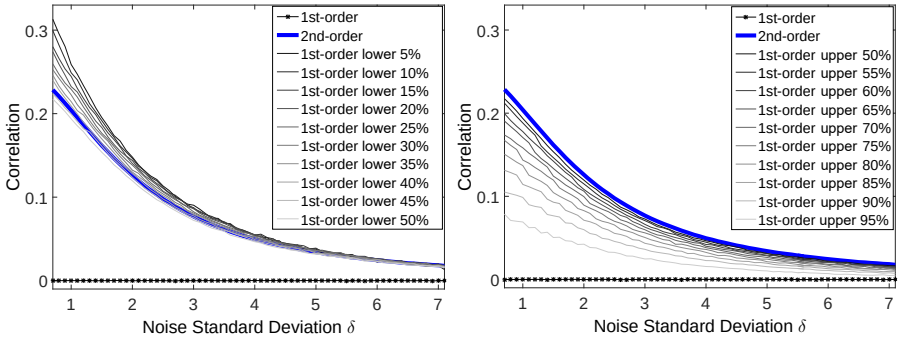
Hence, correlation $\rho(\ddot{L}_1, L_2')$ can be estimated as the second-order correlation. On the other hand, we selected a part of $L_1$ and $L_2$ based on a threshold and following the above procedure estimated the first-order correlation. We conducted this simulation for $n = 1,000,000$ and several values for noise standard deviation $\delta$. For each setting, we examined different thresholds to split the leakages. More precisely, from lower 5% up to lower 50% and from upper 50% to upper 95%, each with steps of 5%. The results are shown in Figure 3(a).

As shown by the graphics, none of the cases, where over 50% of the leakages are considered, can compete with the optimal second-order distinguisher. In contrast, when less than 50% of the leakages are considered, the underlying approach outperforms the second-order one. Further, by increasing the noise level they all become similar and close to the second-order distinguisher. It is noteworthy that due to the symmetry of the distributions in case of this simulation (i.e., first-order masking) the results of the other cases, i.e., upper $< 50\%$ and lower $> 50\%$, are not shown.

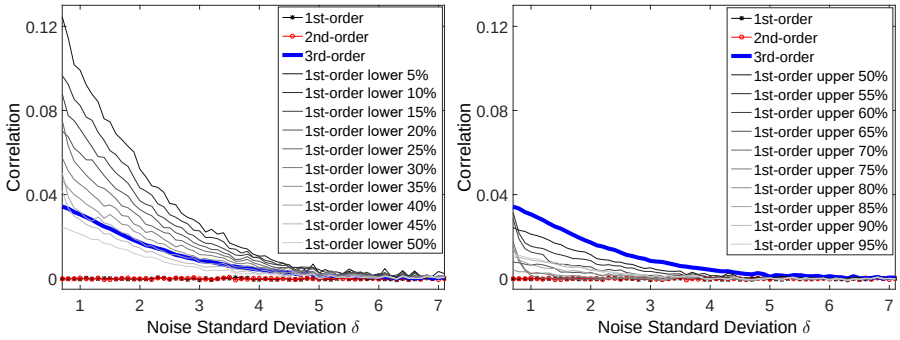
This simulation has been repeated following the above-explained univariate leakage of second-order Boolean masking. Figure 3(b) shows the corresponding results. As expected, the first- and second-order distinguishers would not reveal any dependency. Interestingly, the underlying approach extremely outperforms the optimal third-order distinguisher, and even by increasing the noise standard deviation it still performs better.

We should note that any other distinguisher, where instead of any particular statistical moment the distribution of the leakages are considered, would also differentiate the univariate higher-order leakages. But, these distinguishers (e.g., MIA [9]) would need to predict the probability distributions, e.g., by histogram where the number of bins and the size of each bin play an important role for the efficiency of the distinguisher, alternatively by Kernel where the important issues include the type of the Kernel function and the associated parameters. The diversity of their results based on the selected parameters can make such distinguishers more complicated or less efficient compared to higher-order attacks. However, in the approach presented here we just consider the distribution obtained based on pure histogram. More precisely, the histogram made by the nature of the SCA measurements (i.e., 256 bins as the result of the 8-bit ADC[4] of the acquisition equipment digital oscilloscope) would suffice to find the threshold for a given percentage, e.g., lower 20%.

[4] Analog to Digital Converter.




(a) first-order Boolean masking




(b) second-order Boolean masking

**Fig. 3.** Correlation (based on MC-DPA), simulated univariate (a) second-order and (b) third-order leakages, comparison between different distinguishers for different threshold values over noise standard deviation.

## 4 Practical Results

Now that we have presented the theoretical concept of our approach, it is time to evaluate the soundness of the technique based on real-world measurements taken from the physical implementation of a hardware masking scheme. After a description of the target device and the measurement setup we analyze the side-channel leakage of the test chip by means of conventional higher-order attacks, which are based on the estimation of higher-order statistical moments. As a second step we present the results of our novel approach for different threshold values. At the end, both types of attacks are compared in terms of the required number of measurements for a successful key recovery and the convenience of the procedure from an attacker's point of view.

**Target.** The target platform for our practical evaluations is a 150 nm CMOS ASIC prototype chip. A layered view of the fabricated chip can be seen in Figure 4. The prototype contains 6 different cores and was specifically developed to

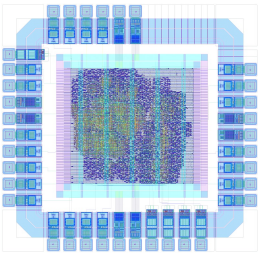


**Fig. 4.** ASIC prototype with 6 cores in 150 nm CMOS.

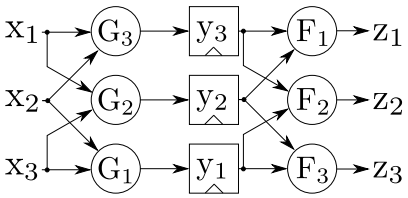



**Fig. 5.** Threshold implementation of the 4-bit PRESENT S-box with 3 shares.

evaluate the side-channel resistance of state-of-the-art block ciphers and DPA countermeasures in practice. The core of the ASIC that is targeted in the following experiments realizes the block cipher PRESENT-80 under 3-share first-order threshold implementation concept. PRESENT-80 is an ultra-lightweight block cipher (ISO/IEC 29192-2:2012 lightweight cryptography standard) that features a block size of 64 bit as well as a key length of 80 bit and consists of 31 computation rounds [4], whereas threshold implementations have been introduced as an efficient hardware masking scheme in [18].

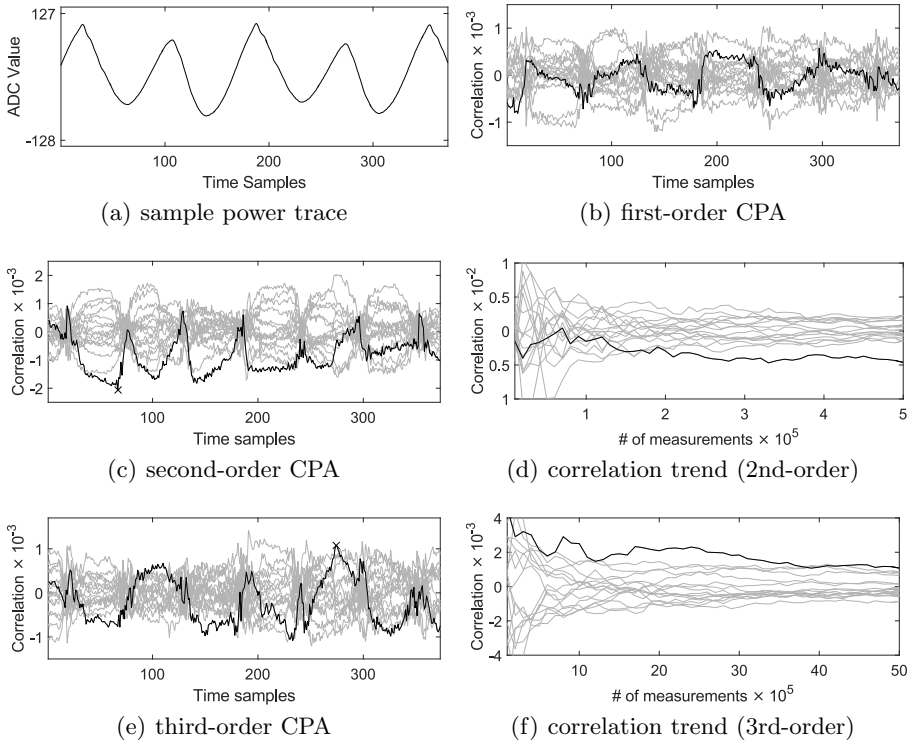
Concerning hardware implementations of masking schemes, it has historically been a challenging task to ensure that glitches in the combinatorial parts of the circuit do not recombine the shares and thus lead to exploitable leakage. Threshold implementations prevent this issue by adding the so-called non-completeness property to the masked computations [2]. Non-completeness means here that each fully combinatorial circuit must be independent of at least one of the shares. This is achieved by splitting the non-linear parts of a circuit into several shared functions that do not operate on all shares at once, but rather perform only one part of the overall computation that refers to its respective inputs. Accordingly, glitches can never recombine all shares at once, meaning that an adversary is not able to learn any information about the secret from the side-channel leakage of only one of these circuits. Indeed, multiple leakages of multiple combinatorial (sub-) circuits need to be combined to perform a successful (higher-order) attack. Following this concept, which is based on Boolean secret sharing and multi-party computation, the threshold implementation technique can be used to implement non-linear functions of symmetric block ciphers in such a way that provable security against first-order power analysis attacks can be guaranteed, even in the presence of glitches. Higher-order threshold implementations can furthermore be used to conceal the leakages at higher-order statistical moments [3]. A second property that has to be fulfilled when sharing a non-linear function is the uniformity of the outputs. For each unshared input to the non-linear function, each shared output should occur equally likely. In this way the output of the shared functions is still uniformly distributed and a remasking is not required. More precisely during the full execution of a block cipher that is implemented in this masking scheme no fresh masks needs to be fed. The plaintext is split up into

the required number of shares at the beginning of the algorithm (see [18]), which implies the generation of two or more plaintext-sized masks, and all further computations are performed on those shares. Compared to conventional masking, the drawback of this method is a higher number of required shares. In particular at least three shares (two masks) are required to realize each non-linear part of a circuit[5]. Additionally the number of shares increases with the degree of the function that needs to be implemented [18]. Hence, larger S-boxes, e.g. 8-bit, are difficult to implement efficiently in this scheme [5]. Nevertheless, for ciphers with small S-boxes, e.g., PRESENT-80, threshold implementation has become the de facto standard for hardware masking [2].

The realization of the PRESENT-80 block cipher as a threshold implementation was introduced in [21]. The authors proposed several implementation profiles with different levels of security. Our targeted ASIC core implements profile 2, which refers to a nibble-serial implementation of the block cipher with a shared data path (with 3 shares) but an unshared key schedule. Hence, one instance of the shared S-box is implemented and the 4-bit nibbles of the cipher state are processed in a pipelined manner. A schematic view of the shared S-box, based on a decomposition to quadratic functions $F$ and $G$ with $S(x) = F(G(x))$, can be seen in Figure 5. Due to the register stage between the $G$- and the $F$ functions one full cipher-round takes 18 clock cycles[6]. It is noteworthy that although first-order threshold implementation corresponds to Boolean masking with 3 shares it provides only first-order security due to its underlying quadratic functions (i.e., $G$ and $F$ in Figure 5). In other words, this implementation is supposed to exhibit second- and third-order leakages.

**Measurement Setup.** We performed our measurements on a Side-channel Attack Standard Evaluation Board (SASEBO-R) [1] that was specifically developed to evaluate the side-channel resistance of cryptographic hardware. For this purpose it provides a socket for an ASIC prototype, which is connected by a 16-bit bidirectional data bus as well as a 16-bit address signal to a Xilinx Virtex-II Pro control FPGA, clocked by a 24-MHz oscillator. For the side-channel measurements a Teledyne LeCroy HRO 66zi oscilloscope was used. We collected 5 million measurements for random plaintexts and a fixed key by measuring the voltage drop over a 1 $\Omega$ resistor in the Vdd path, while the ASIC was operated at a frequency of 3 MHz and a supply voltage of 1.8 V. Each of the power traces contains 100,000 sample points recorded at a sampling rate of 500 MS/s with a resolution of 8 bits. Due to a very low amplitude of the signal two $\times 10$ AC amplifiers in series have been employed, resulting in a $\times 100$ gain. Figure 6(a) depicts a sample trace over the two clock cycles that we are referring to in the following course of this analysis. The two random and uniform 64-bit masks that are needed for the initial sharing of the plaintext are generated and delivered by a PRNG (AES-128 in counter mode) on the control FPGA of the SASEBO-R, which in turn is seeded by the PC via UART.

[5] Lower number of shares can be achieved at the price of additional fresh masks [23].
[6] The permutation layer in one separate clock cycle.




(a) sample power trace (b) first-order CPA

(c) second-order CPA (d) correlation trend (2nd-order)

(e) third-order CPA (f) correlation trend (3rd-order)

**Fig. 6.** Sample power trace and conventional first-, second- and third-order CPA with 5 million measurements using the HW of the $G$-box output.

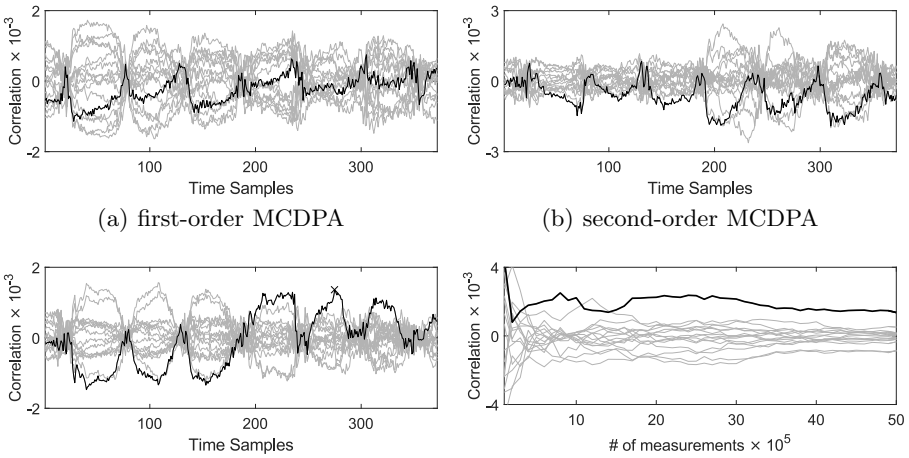
**Results of Conventional Attacks.** To evaluate the effectiveness of the presented approach on noisy real-world measurements it is necessary to assess the vulnerability of the underlying hardware masking scheme by means of conventional DPA attacks in a first step. To this end, we performed first-, second- and third-order Correlation Power Analysis (CPA) attacks [6] using the Hamming weight (HW) of the S-box output (which is the same as the output of the $F$ function in Figure 5). This did not lead to a successful recovery of any key nibble. Hence, we performed the same attack using the HW of the output of the $G$ function (i.e., the value of the intermediate register) and obtained the results which are depicted in Figure 6. All results are plotted over the two clock cycles that leak the targeted intermediate value. This is on the one hand the clock cycle in which the $G$-boxes are evaluated in parallel and on the other hand the succeeding clock cycle where the outputs of the $F$-boxes are computed based on the $G$-box outputs. As expected the first-order attack is not successful. The second-order CPA, on the other hand, reveals the correct key nibble, but only by a slight margin. The third-order attack does not succeed since the correct key candidate does not lead to the overall highest correlation during the targeted two

clock cycles. In particular several ghost peaks with a higher correlation can be identified. For both, the second- and the third-order CPA, we have plotted the evolution of the correlation for the most leaking time sample (marked by a cross in Figure 6(c) and Figure 6(e)). In this way we obtain a quantitative measure to express how many traces are required to reveal the higher-order leakages. For the second-order attack at least 200,000 traces are required, whereas for the third-order attack even with the entire 5,000,000 measurements the correct candidate might not be detectable. We observed the same results targeting several other key nibbles. Indeed, it can be concluded that our measurements are sufficiently noisy to serve as a suitable data source for our further analysis.

The efficiency of CPA attacks relies on the linear dependency between the hypothetical power model (here HW of the $G$-box output) and the actual leakage of the device. Alternatively, Moments-Correlating DPA (MCDPA) [15] can relax such a necessity at the price of (usually) requiring more traces compared to a corresponding CPA with a suitable power model. To examine whether a collision setting can improve the number of required measurements here, which would indicate an imperfect choice of the leakage model in the CPA evaluations, we performed an MCDPA on the same traces. Hereby, the leakage of one S-box is used to build a model which is then used in an attack on another S-box, leading to a recovery of the linear difference between the corresponding key nibbles. In our case the same hardware instance of the S-box is used for both steps, which ensures a similar leakage model. Figure 7 shows the results indicating that only the third-order MCDPA is able to reveal the correct key difference with 5 million measurements [7]. And even this is only true when exclusively the second leaking clock cycle is considered. Otherwise, there are again ghost peaks with a higher correlation. Nevertheless, 1.5 million measurements are required to exploit the third-order leakage. This result enhances our confidence that the Hamming weight of the output of the $G$-box is a suitable leakage model for our target.

**Results of Our Novel Approach.** Hereafter, we concentrate on applying our novel approach (expressed in Section 3) on the same traces. In this regard we first obtained a histogram for each sample point using all 5,000,000 traces. The histograms – as given before – have been made by 256 bins, i.e., the full range of signed 8-bit integers -128 to 127 which reflect the sampled power consumption values unaltered (direct result of the oscilloscope ADC). Therefore, for each given $x\%$ threshold we obtain a threshold trace. This trace contains a threshold value for each sample point individually in such a way that $x\%$ of the traces have a value smaller than the threshold at that sample point and $(100 - x)\%$ have a higher value. As the next step, we conducted the attacks on a subset of traces either as "lower $x\%$" or "upper $(100 - x)\%$". It should be noted that such a separation of traces as well as the attack is performed on each sample point separately. In other words, for each sample point it is individually decided which traces to be considered in the attack.

[7] Only positive correlation values indicate a collision in an MCDPA attack.



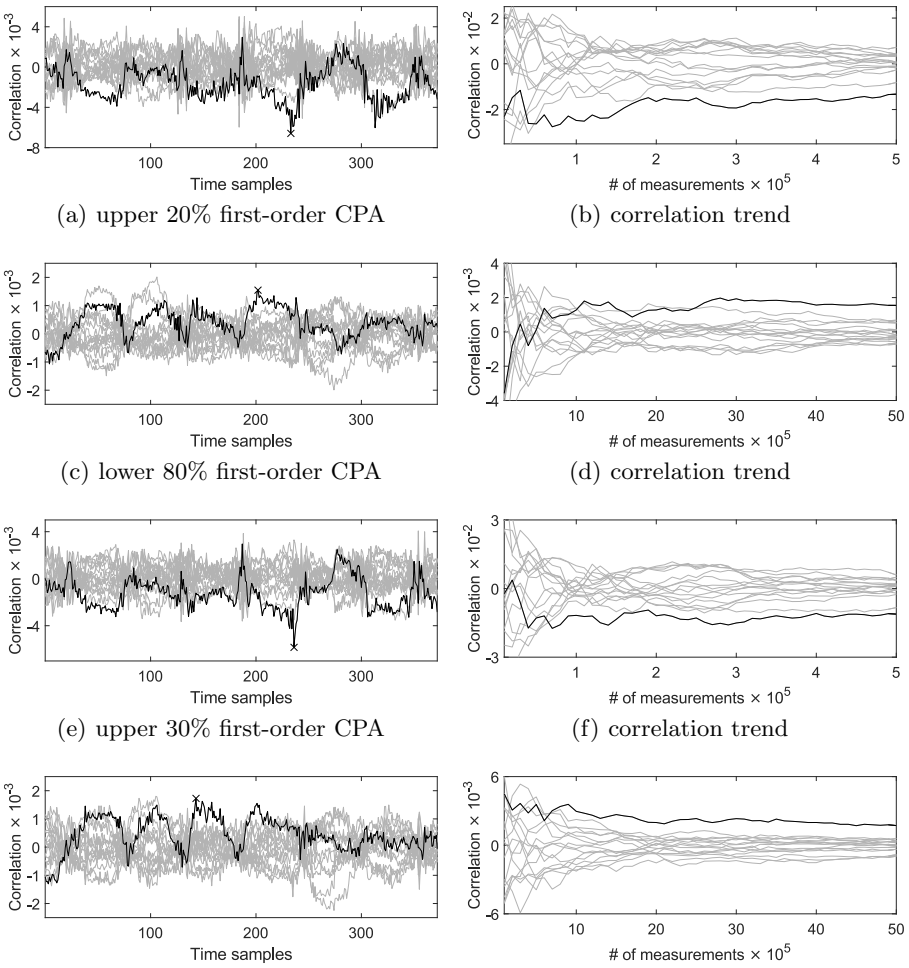

(a) first-order MCDPA (b) second-order MCDPA

(c) third-order MCDPA (d) correlation trend (3rd-order)

**Fig. 7.** Conventional first-, second- and third-order MCDPA with 5 million measurements.

We have examined the threshold values between 5% and 95% with intervals of 5%. In Figure 8 we represent the result of the attacks (CPA with HW of the $G$-box output) for the most successful settings, i.e., 20% and 30% thresholds. Interestingly it can be noted that attacks on subsets with a power consumption below the threshold, i.e., lower 80% and lower 70%, lead to a positive correlation for the correct key candidate, and vise versa for the corresponding upper 20% and upper 30%. This is in fact due to the different biases that are introduced into the three shares by selecting measurements with a power consumption either above or below a certain threshold.

**Comparison.** When comparing our approach to the corresponding conventional second-order CPA, the value of the highest correlation for the correct key candidate is not very meaningful. Due to the fact that a much smaller number of measurements contributes to the results of our approach the correlation values are usually significantly higher compared to the conventional attacks. Hence we have to rely on the required number of measurements as well as a visual inspection of the results as the only available metrics for a comparison. Regarding the required number of measurements we can refer to Figure 8(b) and Figure 8(f) that only 50,000 and respectively 70,000 measurements are required to reveal the leakage with our approach. It should be noted that these numbers as well as Figure 8(b) and Figure 8(f) reflect the number of traces used to both, find the threshold and perform the attack on. In other words, when it is shown that 50,000 traces are required for a "upper 20%" attack, all 50,000 traces are used to find the threshold. Amongst them, around 50,000×20%=10,000 traces are used in the attack. Hence, compared to the conventional second-order attack, the at-




(a) upper 20% first-order CPA (b) correlation trend

(c) lower 80% first-order CPA (d) correlation trend

(e) upper 30% first-order CPA (f) correlation trend

(g) lower 70% first-order CPA (h) correlation trend

**Fig. 8.** First-order CPA on different slices of the 5 million measurements using the Hamming weight of the $G$-box output.

tack with "upper 20%" required 4 times less traces altogether and, due to the fact that only a subset is considered, includes 20 times less traces in the actual CPA computations. In accordance to the simulation results (in Section 3) we can see that the attacks on subsets of traces, that include more than 50% of the measurements, are not able to outperform the conventional attack. More precisely, the "lower 80%" and "lower 70%" attacks (Figure 8(d) and Figure 8(h)) need respectively around 2,500,000 and 700,000 traces while the conventional second-order attack requires 200,000 traces.

All of the presented attacks have been repeated for other key nibbles and therefore on other parts of the power traces as well. These experiments revealed

that concentrating on the "upper 30%" part (for each sample point individually) was indeed most commonly the best choice, although the particular threshold values vary slightly between different key nibbles. Another tendency that could be observed is that the subsets which have been selected from above a threshold were generally significantly more informative than the subsets below a threshold (independent of being each others counterpart). However, for all targeted key nibbles our approach was able to outperform the conventional second-order attack in terms of the required number of measurements for at least one choice of subset.

## 5 Conclusions

In this work we have presented and examined an alternative approach to analyze the higher-order leakages of masked hardware implementations. The proposed technique is able to turn higher-order leakages with a simple selection procedure into a setting where they can be exploited by a first-order distinguisher. This does not only remove the necessity to estimate higher-order statistical moments when attacking masking schemes, which becomes exponentially more complex with an increasing statistical order, but it may also be able to relax the sensitivity of higher-order attacks to the noise level. We have presented the theoretical foundation of our approach by means of simulations and carried out several experiments on noisy real-world measurements to back up our claims. Our analyses lead to the conclusion that our approach indeed represents an alternative to conventional higher-order attacks, and even more importantly is able to outperform them in specific settings. In our setup for example a standard first-order CPA on the subset of traces, that contains only the 20% highest power consumption values (individuality at each sample point), is able to exploit the leakage with 4 times less traces than the conventional second-order CPA attack (i.e., by mean-free square). Hence, a significant improvement could be achieved by simply ignoring a specific part of the traces (at each sample point).

It has been given in literature that masking and hiding countermeasures should be combined to achieve a high level of security. In works like [16] hardware masking is implemented by power-equalization schemes to practically complicate higher-order attacks. As a future work, we will investigate the feasibility of the approach introduced here on such implementations. Another interesting approach to explore is whether it is worthwhile to combine the result of the attacks after splitting the traces. More precisely, we have shown the result of the attacks for "upper 20%" and "lower 80%". The question is whether combining these results would lead to a more effective attack.

**Acknowledgements**

The authors would like to acknowledge Axel Poschmann for the hardware designs and Stefan Heyse for his help on taping out the prototype chip. This work is partly supported by the German Research Foundation (DFG) through the project "NaSCA: Nano-Scale Side-Channel Analysis".

## References


1. Side-channel Attack Standard Evaluation Board SASEBO-R Specification – Version 1.0. `http://www.risec.aist.go.jp/project/sasebo/download/SASEBO-R_Spec_Ver1.0_English.pdf`. Research Center for Information Security, National Institute of Advanced Industrial Science and Technology, Japan.
2. Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel Implementations of Masking Schemes and the Bounded Moment Leakage Model. In *EUROCRYPT 2017*, Lecture Notes in Computer Science. Springer, 2017. to appear.
3. Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Higher-Order Threshold Implementations. In *ASIACRYPT 2014*, volume 8874 of *Lecture Notes in Computer Science*, pages 326–343. Springer, 2014.
4. 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 *CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
5. Erik Boss, Vincent Grosso, Tim Güneysu, Gregor Leander, Amir Moradi, and Tobias Schneider. Strong 8-bit Sboxes with Efficient Masking in Hardware. In *CHES 2016*, volume 9813 of *Lecture Notes in Computer Science*, pages 171–193. Springer, 2016.
6. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In *CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.
7. Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi. Leakage Squeezing of Order Two. In *INDOCRYPT 2012*, volume 7668 of *Lecture Notes in Computer Science*, pages 120–139. Springer, 2012.
8. Zhimin Chen and Patrick Schaumont. Slicing Up a Perfect Hardware Masking Scheme. In *HOST 2008*, pages 21–25. IEEE, 2008.
9. Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual Information Analysis. In *CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.
10. Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In *CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.
11. Yongdae Kim, Takeshi Sugawara, Naofumi Homma, Takafumi Aoki, and Akashi Satoh. Biasing power traces to improve correlation in power analysis attacks. In *COSADE 2010*, pages 77–80, 2010.
12. Houssem Maghrebi, Sylvain Guilley, and Jean-Luc Danger. Leakage Squeezing Countermeasure against High-Order Attacks. In *WISTP 2011*, volume 6633 of *Lecture Notes in Computer Science*, pages 208–223. Springer, 2011.
13. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2007.
14. Amir Moradi and Oliver Mischke. On the Simplicity of Converting Leakages from Multivariate to Univariate - (Case Study of a Glitch-Resistant Masking Scheme). In *CHES 2013*, volume 8086 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2013.
15. Amir Moradi and François-Xavier Standaert. Moments-Correlating DPA. In *Workshop on Theory of Implementation Security*, TIS '16, pages 5–15. ACM, 2016.

16. Amir Moradi and Alexander Wild. Assessment of Hiding the Higher-Order Leakages in Hardware. In *CHES 2015*, volume 9293 of *Lecture Notes in Computer Science*, pages 453–474. Springer, 2015.
17. 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.
18. 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.
19. Changhai Ou, Zhu Wang, Degang Sun, Xinping Zhou, Juan Ai, and Na Pang. Enhanced Correlation Power Analysis by Biasing Power Traces. In *ISC 2016*, volume 9866 of *Lecture Notes in Computer Science*, pages 59–72. Springer, 2016.
20. Thomas Popp and Stefan Mangard. Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints. In *CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*, pages 172–186. Springer, 2005.
21. 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.
22. Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.
23. Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating Masking Schemes. In *CRYPTO 2015*, volume 9215 of *Lecture Notes in Computer Science*, pages 764–783. Springer, 2015.
24. Patrick Schaumont and Kris Tiri. Masking and Dual-Rail Logic Don't Add Up. In *CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 95–106. Springer, 2007.
25. Tobias Schneider and Amir Moradi. Leakage Assessment Methodology - A Clear Roadmap for Side-Channel Evaluations. In *CHES 2015*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.
26. Stefan Tillich, Christoph Herbst, and Stefan Mangard. Protecting AES Software Implementations on 32-Bit Processors Against Power Analysis. In *ACNS 2007*, volume 4521 of *Lecture Notes in Computer Science*, pages 141–157. Springer, 2007.
27. Kris Tiri and Patrick Schaumont. Changing the Odds Against Masked Logic. In *SAC 2006*, volume 4356 of *Lecture Notes in Computer Science*, pages 134–146. Springer, 2006.