Statistical Tools Flavor Side-Channel Collision Attacks

Amir Moradi

Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany
moradi@crypto.rub.de

Abstract. By examining the similarity of side-channel leakages, collision attacks evade the indispensable hypothetical leakage models of multi-query based side-channel distinguishers like correlation power analysis and mutual information analysis attacks. Most of the side-channel collision attacks compare two selective observations, what makes them similar to simple power analysis attacks. A multi-query collision attack detecting several collisions at the same time by means of comparing the leakage averages was presented at CHES 2010. To be successful this attack requires the means of the side-channel leakages to be related to the processed intermediate values. It therefore fails in case the mean values and processed data are independent, even though the leakages and the processed values follow a clear relationship. The contribution of this article is to extend the scope of this attack by employing additional statistics to detect the colliding situations. Instead of restricting the analyses to evaluation of means, we propose to employ higher-order statistical moments and probability density functions as the figure of merit to detect collisions. Thus, our new techniques remove the shortcomings of the existing correlation collision attacks using first-order moments. In addition to the theoretical discussion of our approach, practical evidence of its suitability for side-channel evaluation is provided. We provide four case studies, including three FPGA-based masked hardware implementations and a software implementation using boolean masking on a microcontroller, to support our theoretical groundwork.

1 Introduction

Integration of embedded computers into our daily life, e.g., in automotive applications and smartcard applications for financial purposes, led to a widespread deployment of security-sensitive devices. On the downside also adversaries benefit from the resulting easy physical accessibility, as it provides control over the devices and thus simplifies analyses. Consequently, today most sensitive embedded systems need to be considered as operating in a hostile environment. For this reason physical attacks, most notably side-channel analyses, are considered major threats. For instance, power analysis and the closely related electro-magnetic (EM) analysis can easily overcome the security features of unprotected designs by monitoring the power consumption of the executing device. In order to distinguish the correct key hypothesis amongst the others *differential power analysis*

D. Pointcheval and T. Johansson (Eds.): EUROCRYPT 2012, LNCS 7237, pp. 428–445, 2012.

(DPA) [13] and its successor form, the *correlation power analysis* (CPA) [7], use statistical tools: the *difference of means* and the *Pearson correlation coefficient*, respectively. The distinguisher is applied to side-channel observations classified into subsets defined by a boolean partitioning function in the case of a DPA or by means of a hypothetical power model in case of a CPA. The later introduced *mutual information analysis* (MIA) [11] provides a generic distinguisher that lifts the need of sophisticated power models at the cost of an increased number of required side-channel observations. Generally speaking, MIA is able to recover secret information when the CPA fails due to the lack of a suitable power model. However, the efficiency of MIA also relies on the availability of a good hypothetical model that reflects the dependencies of the actual data-dependent leakage provided by the side-channel observations. The loss of efficiency becomes most visible and even critical when the underlying function of the target device is not a many-to-one mapping (see the detailed discussion provided in [26]).

In order to develop an attack method that does not require a device dependent model, a new type of side-channel attacks has been introduced: the side-channel based collision attacks [2, 3, 5, 24, 25]. These methods adopt collision attacks to side-channel analyses and allow efficiently extracting secrets from side-channel measurements using only a small number of observations, especially when the design architecture of the target implementation is known to the adversary (see e.g., [4] where collision attacks are combined with DPA). Collision attacks, however, get infeasible when facing very noisy observations or in presence of both, time-domain and data-domain randomizing countermeasures. Recent works propose a couple of techniques, e.g., in [3] to deal with false-positive collision detections and [10] which reports a successful attack on a mask-protected software implementation which exploits reused masks. Another recent attack method [16] named correlation collision attack exploits conditions that lead to a multitude of collisions whenever a key-dependent relation between the processed input values is fulfilled. More precisely, it compares the sets of leakages (averaged with respect to a fixed relevant input) of one e.g., S-box instance when it processes two distinct input sets, each of which associated with a different part of the secret key. The relation between the inputs of the two sets, that causes all averages to collide, exposes information on the secret key. During the last years the independence of side-channel collision attacks from hypothetical models and the effects of process variations which harden side-channel attacks in nanoscale devices [23] increasingly attracts the attention of the community to the new attack methods leading to new applications and variations as in [17] and [10].

A correlation collision attack [16] applies a statistical tool, i.e., the Pearson correlation coefficient, on the means of side-channel observations that were classified with respect to known input data. This method is successful when the means of the classified side-channel observations are different when they are estimated using a large (but feasible) amount of observations. If the mean values do not show the required dependency to processed data, the attack will fail, even in case there is a clear relation between the processed values and the observed side-channel leakages. To give an example, we refer to *threshold implementations* [19, 20], which claim that the averages of the side-channel leakages are independent of the processed values. In this case a MIA might still be able to exploit the leakage to recover a secret key [21]. Similarly a correlation collision attack is not able to recover the desired secret in this case (as also stated in [18]), as it relies on mean values. Indeed, this was one of the motivations for this work to apply other statistics in side-channel collision attacks in order to enable detection of colliding situations also in cases where the mean values do not provide any exploitable information.

In this article we discuss how to extend the scope of correlation collision attacks to exploit dependencies in different central moments from probability theory and statistics. Furthermore, we elaborate on preprocessing schemes which can be performed to improve correlation collision attacks. We show that in certain situations applying a preprocessing step prior to a correlation collision attack on mean values is equivalent to the same attack targeting a high-order statistical moment. In order to generalize the scheme we moreover propose to compare probability density functions (pdf) instead of any specific moments. Although accurately estimating the pdfs is an expensive task that requires a high number of observations, this generalized approach does not require any assumptions about the type and shape of the leakage distributions and may thus be worth the additional efforts.

In order to practically investigate our proposed schemes on different implementations we have considered both, an FPGA-based platform as well as a microcontroller. Three different masked hardware implementations were mapped to our target FPGA device. These include *i*) an AES encryption engine using the masked S-box presented in [8], *ii*) an implementation of PRESENT [6] using the threshold implementation countermeasure as presented in [22], and *iii*) a threshold implementation of the AES as reported in [18]. Since the masked values and the masks are processed simultaneously in all the aforementioned implementations, a univariate attack method is an applicable choice. We show how to use different statistical moments in a collision attack to recover the desired secret and we discuss their efficiencies. As a fourth case study a software implementation of first-order boolean masking on a microcontroller is analyzed. Here, since the masks and the masked values are processed sequentially, a multivariate attack needs to be applied. We use this case study to illustrate possible solutions including multivariate collision attacks and univariate ones which employ a combining function.

2 Preliminaries

In the following we introduce the notation used in this paper and explain the adopted side-channel model. Afterwards, Section 2.2 provides a short review of linear collision attacks followed by a formal specification of correlation collision attacks.

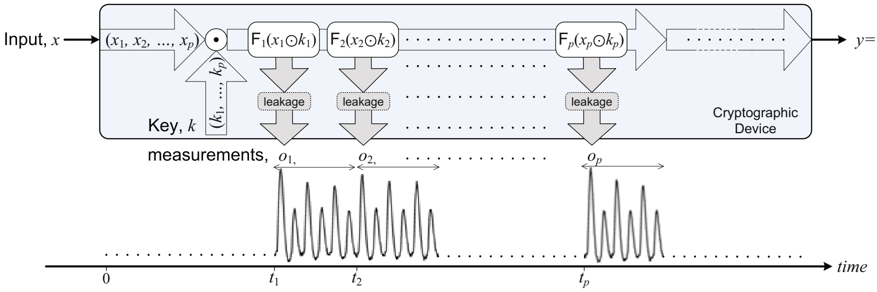


Fig. 1. Side-channel model

2.1 Notations and Side-Channel Model

We consider a cryptographic device that performs the cryptographic operation E on the given input x. E depends on the secret key k and outputs the value $y = \mathsf{E}_k(x)$ (see Fig. 1). The algorithmic computations depending on x and k cause internal state transitions (e.g., bit flips). The internal state transitions affect the side-channel observations o, which are noisy measurements of the leakages.

In Fig. 1 we suppose that the considered cryptographic operation is an iterated symmetric block cipher that starts with a *key whitening* step represented by the general conjunction $\odot$. To denote the small parts of input data and the key used in a *divide-and-conquer* key recovery scheme on the cryptographic operation E, we use the subscripts i, i.e., x_i and k_i, where $i \in \{1, \ldots, p\}$ and p is the number of different parts used. Furthermore, we introduce the functions F_i (usually nonlinear), that independently process the key-whitened inputs $x_i \odot k_i$. Although for simplicity we have supposed that each function F_i is performed at a different time ${t_i}^1$, sequential or parallel execution of the functions F_i depends on the actual implementations platform and architecture.

Performing q queries to the target device an adversary acquires the side-channel measurements $\boldsymbol{o}^1, \ldots, \boldsymbol{o}^q$ corresponding to the device's processing of the supplied inputs $x^1, \ldots, x^q$. The j-th measurement $\boldsymbol{o}^j$ consists of p parts $\boldsymbol{o}^j_1, \ldots, \boldsymbol{o}^j_p$ corresponding to the computations of the functions F_i at times t_i. Note that each side-channel measurement itself still consists of multiple samples. That is, the i-th part of the j-th measurement, i.e., the vector $\boldsymbol{o}^j_i$, denotes s subsequently measured samples $o^j_{i,1}, \ldots, o^j_{i,s}$.[2]

For example in a CPA attack, for a specific portion i the adversary determines $\boldsymbol{w}_\mathsf{i}$ as a vector of estimated internal state transitions w^j_i, $\forall\ 1 \leq j \leq q$ using the input portion x^j_i and a hypothesis for the key portion k_i. Then, he evaluates his guess by comparing the leakage modeled by $\widehat{L}(\boldsymbol{w}_\mathsf{i})$ to the actual measurements $o^j_{\mathsf{i},\mathsf{s}}$.

[1] Times are measured relative to the the start of each processing of E.

[2] Note that in each measurement j the measurement parts $\boldsymbol{o}^\mathsf{j}_{i=1,\ldots,p}$ may overlap in some sample points. This is helpful when the exact time instances t_i are uncertain but their distances, e.g., the number of clock cycles between the consecutive t_i, are known.

Hereby the leakages of the sample points $\mathsf{s} \in \{1, \ldots, s\}$ are considered independently. The most appealing advantage of the collision attacks, which are restated in the following, is to avoid requiring the hypothetical leakage model $\widehat{L}(\cdot)$.

2.2 Correlation Collision Attack

In the case that two functions $\mathsf{F}_{\mathsf{i}_1}$ and $\mathsf{F}_{\mathsf{i}_2}$ ($\mathsf{i}_1 \neq \mathsf{i}_2 \in \{1, \ldots, p\}$) are identical (see Fig. 1), a collision attack might be possible. Analyzing the measurements $\boldsymbol{o}_{i_1}$ and $\boldsymbol{o}_{\mathsf{i}_2}$ a collision attack aims at detecting situations where both functions process the same value. In this case injective functions $\mathsf{F}_{\mathsf{i}_1} = \mathsf{F}_{\mathsf{i}_2}$ (e.g., the AES S-box) allow concluding

$$\begin{aligned}
& & \mathsf{F}_{\mathsf{i}_1}(x_{\mathsf{i}_1} \odot k_{\mathsf{i}_1}) &= \mathsf{F}_{\mathsf{i}_2}(x_{\mathsf{i}_2} \odot k_{\mathsf{i}_2}) \\
&\Leftrightarrow & x_{\mathsf{i}_1} \odot k_{\mathsf{i}_1} &= x_{\mathsf{i}_2} \odot k_{\mathsf{i}_2} \\
&\Leftrightarrow & (x_{\mathsf{i}_1})^{-1} \odot x_{\mathsf{i}_1} \odot k_{\mathsf{i}_1} \odot (k_{\mathsf{i}_2})^{-1} &= (x_{\mathsf{i}_1})^{-1} \odot x_{\mathsf{i}_2} \odot k_{\mathsf{i}_2} \odot (k_{\mathsf{i}_2})^{-1} \\
&\Leftrightarrow & \Delta k_{\mathsf{i}_1,\mathsf{i}_2} = k_{\mathsf{i}_1} \odot (k_{\mathsf{i}_2})^{-1} &= (x_{\mathsf{i}_1})^{-1} \odot x_{\mathsf{i}_2},
\end{aligned}$$

where $(k_{\mathsf{i}_2})^{-1}$ and $(x_{\mathsf{i}_1})^{-1}$ are respectively a right inverse of k_{i_2} and a left inverse of x_{i_1}, i.e., $k_{\mathsf{i}_2} \odot (k_{\mathsf{i}_2})^{-1} = e_r$ and $(x_{\mathsf{i}_1})^{-1} \odot x_{\mathsf{i}_1} = e_l$, where e_r and e_l are respectively a right and a left identity element of operation $\odot$. Since x_{i_1} and x_{i_2} are supposed to be known to the adversary, $\Delta k_{\mathsf{i}_1,\mathsf{i}_2}$ gets revealed detecting such a collision. If additional instances of the function F_i are processed within the analyzed algorithm E, all available instances can be pairwise evaluated to reveal terms $\Delta k_{\cdot,\cdot}$ as described above. Depending on the target algorithm this allows an adversary to either determine all parts of the key or to significantly shrink the key space, what allows for feasible exhaustive key searches.

When the target device implements the AES, the functions F_i are AES S-boxes and the conjunction $\odot$ is the first call to the AddRoundKey operation (i.e., $x_i \oplus k_i$, $\oplus$ denoting bitwise XOR) prior to the first round of the encryption. Then, detecting a collision (called linear collision on AES [3]) $\Delta k_{\mathsf{i}_1,\mathsf{i}_2} = k_{\mathsf{i}_1} \oplus k_{\mathsf{i}_2} = x_{\mathsf{i}_1} \oplus x_{\mathsf{i}_2}$ is recovered. In this case, the adversary can recover a maximum of 15 linearly independent relations between the key portions allowing the key search space to be restricted to 2^8.

In the first generation of side-channel collision attacks, e.g., [2–5, 24, 25], the collision detection process is implemented by pairwise comparing measurement parts $(\boldsymbol{o}_{\mathsf{i}_1}^{\mathsf{j}_1}, \boldsymbol{o}_{\mathsf{i}_2}^{\mathsf{j}_2})$ where $\mathsf{j}_1, \mathsf{j}_2 \in \{1, \ldots, q\}$. Also, different methods were used to perform the comparison (e.g., the Euclidean distance in [25]). Although one needs to deal with false-positive comparison results, this attack is feasible when the target device and architecture sequentially processes the algorithm, e.g., a microcontroller. Also, the more clock cycles the observations $(\boldsymbol{o}_{\mathsf{i}_1}^{\mathsf{j}_1}, \boldsymbol{o}_{\mathsf{i}_2}^{\mathsf{j}_2})$ include, the more robust the detection gets, leading to a more feasible attack.

However, when attacking a hardware implementation which simultaneously performs multiple operations or when randomizing countermeasures or noise addition schemes are embedded into the target device, examining the similarity of a pair of measurement parts will probably fail to detect the collisions. Also, in these cases each measurement part $\boldsymbol{o}_i$ usually covers only a single clock cycle. The attack introduced in [16] (the so-called *correlation collision attack*) uses a different scheme to overcome such problems. As the instances of the functions $\mathsf{F}_{\mathsf{i}_1}$ and $\mathsf{F}_{\mathsf{i}_2}$ always collide whenever the condition $x_{\mathsf{i}_2} = x_{\mathsf{i}_1} \odot \Delta k_{\mathsf{i}_1,\mathsf{i}_2}$ holds, $\Delta k_{\mathsf{i}_1,\mathsf{i}_2}$ can be recovered by means of a hypothesis test. In order to do so, two sets of mean vectors, denoted by $\mathbb{M}_{\mathsf{i}_1}$ and $\mathbb{M}_{\mathsf{i}_2}$, are computed. Each set $\mathbb{M}_{\mathsf{i}}$ consists of 2^n mean vectors $\left\{\boldsymbol{m}_{\mathsf{i}}^0, \ldots, \boldsymbol{m}_{\mathsf{i}}^{2^n-1}\right\}$, where n is the bit-length of a plaintext (or key) portion. Each mean vector $\boldsymbol{m}_{\mathsf{i}}^{\mathrm{x}}$, $\mathrm{x} \in \mathbb{F}_{2^n}$ again consists of s mean samples $(m_{\mathsf{i},1}^{\mathrm{x}}, \ldots, m_{\mathsf{i},s}^{\mathrm{x}})$ which are defined as

$$m_{\mathsf{i},\mathsf{s}}^{\mathrm{x}} = \frac{1}{q_{\mathsf{i}}^{\mathrm{x}}} \sum_{j=1, x_{\mathsf{i}}^j = \mathrm{x}}^{q} o_{\mathsf{i},\mathsf{s}}^j, \qquad \mathsf{s} \in \{1, \ldots, s\}, \qquad \mathrm{x} \in \mathbb{F}_{2^n},$$

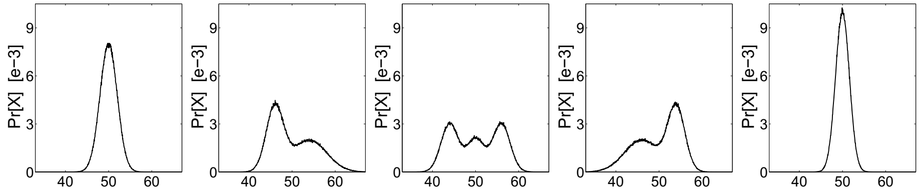
where $q_{\mathsf{i}}^{\mathrm{x}}$ denotes the cardinality of the set $\left\{j : 1 \leq j \leq q \,|\, x_{\mathsf{i}}^j = \mathrm{x}\right\}$. Now based on $\widehat{\Delta\mathsf{k}}$, i.e., a hypothesis for $\Delta k_{\mathsf{i}_1,\mathsf{i}_2}$, two vectors $\boldsymbol{m}'_{\mathsf{i}_1,\mathsf{s}}$ and ${}^{\widehat{\Delta\mathsf{k}}}\boldsymbol{m}'_{\mathsf{i}_2,\mathsf{s}}$ are extracted from the two sets $\mathbb{M}_{\mathsf{i}_1}$ and $\mathbb{M}_{\mathsf{i}_2}$ defined above:

$$\begin{aligned} \boldsymbol{m}'_{\mathsf{i}_1,\mathsf{s}} &= (\, m'^{0}_{\mathsf{i}_1,\mathsf{s}}, \ldots,\, m'^{2^n-1}_{\mathsf{i}_1,\mathsf{s}}), & m'^{\mathrm{x}}_{\mathsf{i}_1,\mathsf{s}} &= m^{\mathrm{x}}_{\mathsf{i}_1,\mathsf{s}}, & \mathrm{x} &\in \mathbb{F}_{2^n} \\ {}^{\widehat{\Delta\mathsf{k}}}\boldsymbol{m}'_{\mathsf{i}_2,\mathsf{s}} &= (\, {}^{\widehat{\Delta\mathsf{k}}}m'^{0}_{\mathsf{i}_2,\mathsf{s}}, \ldots,\, {}^{\widehat{\Delta\mathsf{k}}}m'^{2^n-1}_{\mathsf{i}_2,\mathsf{s}}), & {}^{\widehat{\Delta\mathsf{k}}}m'^{\mathrm{x}}_{\mathsf{i}_2,\mathsf{s}} &= m^{\mathrm{x}\odot\widehat{\Delta\mathsf{k}}}_{\mathsf{i}_2,\mathsf{s}}, & \mathrm{x} &\in \mathbb{F}_{2^n}. \end{aligned}$$

Now Pearson's correlation coefficient can be used to measure the similarity of the pair of vectors $\boldsymbol{m}'_{\mathsf{i}_1,\mathsf{s}}$ and ${}^{\widehat{\Delta\mathsf{k}}}\boldsymbol{m}'_{\mathsf{i}_2,\mathsf{s}}$. The most similar vectors at the analyzed sample point s indicate the most probable $\widehat{\Delta\mathsf{k}}$. This procedure – similar to most of the non-profiled attacks – is repeated for each sample point s independently. The time instances t_{i_1} and t_{i_2} (see Fig. 1) are initially not known to an adversary without detailed information on the implemented architecture of the target device, but [16] proposes a method to reveal this information. The suggested method is to analyze the variance of $\{m_{\mathsf{i},\mathsf{s}}^{\mathrm{x}} : \forall\ \mathrm{x} \in \mathbb{F}_{2^n}\}$: If the means of the measurements at sample point s depend on the inputs x_{i}, the variance at sample point s is significantly increased compared to other sample points.

3 Shortcomings and Our Solutions

Since only the mean values contribute to the comparison metric of the original correlation collision attack, it cannot detect collisions whenever the means of the leakages do not depend on processed data, even if the distributions of the leakages show a strong data dependence. As an example consider the distributions in Fig. 2. Since the shown distributions have the same mean, the attack will fail, although the distributions can clearly be distinguished by their shape.



(a) (b) (c) (d) (e)

Fig. 2. Examples of probability distributions with the same mean

3.1 Higher-Order Moments

While the mean of all the probability distributions shown in Fig. 2 is the same, their higher-order moments are different. For instance, Figures 2(b), (c), and (d) can be discriminated by their skewnesses. Also, Fig. 2(a) can be distinguished from Figures 2(b), (c), and (d) by the variance, and from Fig. 2(e) by the kurtosis. Therefore, in order to extend the scheme, one can exploit the differences in the higher statistical moments similarly to the analysis of the mean values performed before. In other words, extending the notations given in Section 2.2 we can calculate the sets of the d-th central moments ($d > 1$) ${}_d\mathbb{\mu}_{\mathsf{i}_1}$ and ${}_d\mathbb{\mu}_{\mathsf{i}_2}$ of the i_1-th and i_2-th measurement parts. As before, each set ${}_d\mathbb{\mu}_{\mathsf{i}}$ consists of 2^n vectors $\left\{{}_d\boldsymbol{\mu}^0_{\mathsf{i}}, \dots, {}_d\boldsymbol{\mu}^{2^n-1}_{\mathsf{i}}\right\}$, and each vector ${}_d\boldsymbol{\mu}^{\mathsf{x}}_{\mathsf{i}}$ includes s elements $\left({}_d\mu^{\mathsf{x}}_{\mathsf{i},1}, \dots, {}_d\mu^{\mathsf{x}}_{\mathsf{i},s}\right)$ which are the d-th central moment values for the different sample points. The d-th central moment for a sample point is calculated by

$${}_d\mu^{\mathsf{x}}_{\mathsf{i,s}} = \frac{1}{q^{\mathsf{x}}_{\mathsf{i}}} \sum_{j=1,\, x^j_{\mathsf{i}}=\mathsf{x}}^{q} \left(o^j_{\mathsf{i,s}} - m^{\mathsf{x}}_{\mathsf{i,s}}\right)^d, \qquad \mathsf{s} \in \{1, \dots, s\}, \qquad \mathsf{x} \in \mathbb{F}_{2^n}.$$

Note that ${}_2\mathbb{\mu}_{\mathsf{i}}$ indicates the variances, and for $d > 2$ it is recommended to use the standardized central moments defined as $\frac{{}_d\mu^{\mathsf{x}}_{\mathsf{i,s}}}{\left(\sqrt{{}_2\mu^{\mathsf{x}}_{\mathsf{i,s}}}\right)^d}$. The remaining task is to create vectors of the sets defined again in order to compare them. Using the same rules as before, a hypothesis $\widehat{\Delta\mathsf{k}}$ is used to construct the two vectors

$$\begin{aligned}
{}_d\boldsymbol{\mu}'_{\mathsf{i}_1,\mathsf{s}} &= \left({}_d\mu'^{0}_{\mathsf{i}_1,\mathsf{s}}, \dots, {}_d\mu'^{2^n-1}_{\mathsf{i}_1,\mathsf{s}}\right), & {}_d\mu'^{\mathsf{x}}_{\mathsf{i}_1,\mathsf{s}} &= {}_d\mu^{\mathsf{x}}_{\mathsf{i}_1,\mathsf{s}}, & \mathsf{x} &\in \mathbb{F}_{2^n} \\
{}^{\widehat{\Delta\mathsf{k}}}_{d}\boldsymbol{\mu}'_{\mathsf{i}_2,\mathsf{s}} &= \left({}^{\widehat{\Delta\mathsf{k}}}_{d}\mu'^{0}_{\mathsf{i}_2,\mathsf{s}}, \dots, {}^{\widehat{\Delta\mathsf{k}}}_{d}\mu'^{2^n-1}_{\mathsf{i}_2,\mathsf{s}}\right), & {}^{\widehat{\Delta\mathsf{k}}}_{d}\mu'^{\mathsf{x}}_{\mathsf{i}_2,\mathsf{s}} &= {}_d\mu^{\mathsf{x}\odot\widehat{\Delta\mathsf{k}}}_{\mathsf{i}_2,\mathsf{s}}, & \mathsf{x} &\in \mathbb{F}_{2^n}.
\end{aligned}$$

Using the same comparison technique as in the original correlation collision attack, one can compare the aforementioned vectors using the Pearson correlation coefficient at each sample point and for each $\widehat{\Delta\mathsf{k}}$ independently.

In fact, the use of high-order central statistical moments is equivalent to perform a preprocessing step on the side-channel observations before running the original correlation collision attack. For instance, for $d = 2$ the use of ${}_2\mathbb{\mu}_i$ (variance) is identical to squaring the mean-free traces and then computing the mean sets ($\mathbb{\mu}_i$). $d = 3$ and $d = 4$ (skewness and kurtosis if standardized) are the same as cubing and getting the fourth power of the mean-free traces. As shown later in Section 4 the use of high-order moments leads to efficient attack methods to analyze masked implementations that process the masks and the masked data simultaneously. We should highlight that the higher the moment, the harder it is to estimate. That is, a large number of observations q are required to obtain a reasonably precise estimation. Thus, the use of higher-oder moments $(d > 4)$ is very limited in practice. Nevertheless, there might be architectures where the attacks can still benefit from going to these higher-order moments.

3.2 Collision Detection Using Probability Density Functions

In order to generalize the scheme we also evaluated collision detection by comparing pdfs instead of focusing on a particular moment. To do so, we define $\mathfrak{P}_{\mathsf{i}}$ as a family of 2^n sets $\left\{ \mathbb{P}_{\mathsf{i}}^{0}, \ldots, \mathbb{P}_{\mathsf{i}}^{2^n-1} \right\}$. Each set $\mathbb{P}_{\mathsf{i}}^{\mathrm{x}}$ consists of s probability density functions $\left\{ f_{\mathsf{i},1}^{\mathrm{x}}(O), \ldots, f_{\mathsf{i},s}^{\mathrm{x}}(O) \right\}$ defined as follows.

$$f_{\mathsf{i,s}}^{\mathrm{x}}(O = o) = \Pr\left[H(O_{\mathsf{i,s}}) = o | X_{\mathsf{i}} = \mathrm{x}\right], \qquad \mathsf{s} \in \{1, \ldots, s\}, \qquad \mathrm{x} \in \mathbb{F}_{2^n}$$

Here we introduced the random variables $O_{\mathsf{i,s}}$ and X_{i} describing the distribution of the observed values $o_{\mathsf{i,s}}$ and the input portions x_{i} respectively. Furthermore, we introduced a new random variable O, which is used to estimate the pdf of $O_{\mathsf{i,s}}$. We denote the sample space of O as $\mathcal{O}$ and samples as o. We further introduced a function $H(O_{\mathsf{i,s}})$ (e.g., bins of a histogram), which maps samples of $O_{\mathsf{i,s}}$ to elements of $\mathcal{O}$, i.e., the sample space $\mathcal{O}$ used to estimate the pdf may differ from the sample space of the observed values.

We continue as before and extract the sets $\mathbb{P}'_{\mathsf{i}_1,\mathsf{s}}$ and ${}^{\Delta\widehat{\mathsf{k}}}\mathbb{P}'_{\mathsf{i}_2,\mathsf{s}}$ from the families $\mathfrak{P}_{\mathsf{i}_1}$ and $\mathfrak{P}_{\mathsf{i}_2}$, each of which includes 2^n pdfs

$$\mathbb{P}_{\mathsf{i}_1,\mathsf{s}} = \left\{ f'^{0}_{\mathsf{i}_1,\mathsf{s}}(O), \ldots, f'^{2^n-1}_{\mathsf{i}_1,\mathsf{s}}(O) \right\}, \qquad f'^{\mathrm{x}}_{\mathsf{i}_1,\mathsf{s}}(O) = f^{\mathrm{x}}_{\mathsf{i}_1,\mathsf{s}}(O), \qquad \mathrm{x} \in \mathbb{F}_{2^n}$$
$${}^{\Delta\widehat{\mathsf{k}}}\mathbb{P}'_{\mathsf{i}_2,\mathsf{s}} = \left\{ {}^{\Delta\widehat{\mathsf{k}}}f'^{0}_{\mathsf{i}_2,\mathsf{s}}(O), \ldots, {}^{\Delta\widehat{\mathsf{k}}}f'^{2^n-1}_{\mathsf{i}_2,\mathsf{s}}(O) \right\}, \quad {}^{\Delta\widehat{\mathsf{k}}}f'^{\mathrm{x}}_{\mathsf{i}_2,\mathsf{s}}(O) = f^{\mathrm{x} \odot \Delta\widehat{\mathsf{k}}}_{\mathsf{i}_2,\mathsf{s}}(O), \quad \mathrm{x} \in \mathbb{F}_{2^n}.$$

In contrast to the central moments discussed before, we now need to compare vectors of distributions instead of scalar vectors in order to find a similarity metric that allows distinguishing collisions. Fortunately, comparing pdfs is a well-studied task used in many different research fields, e.g., pattern recognition. The well-known methods include the *Squared Euclidean*, *Kullback-Leibler*, *Jeffreys*, *f-divergence*, and several others (for a comprehensive list see [9]). In the following we summarize the Kullback-Leibler (KL) divergence [14], which is the basis of several other schemes and including the metric we used in our experiments.

Kullback-Leibler Divergence is a non-negative measure of the difference between two probability distributions $p(O)$ and $q(O)$. For the discrete case it is defined as

$$\mathrm{D}_{\mathrm{KL}}(p(O)||q(O)) = \sum_{o \in \mathcal{O}} p(o) \log \frac{p(o)}{q(o)}.$$

In fact, KL divergence is not a true distance metric as it is not symmetric, i.e., $\mathrm{D_{KL}}(p(O)||q(O)) \neq \mathrm{D_{KL}}(q(O)||p(O))$. Therefore, other schemes have been introduced to develop a symmetric metric with similar properties. For instance,

$$\mathrm{D_J}(p(O)||q(O)) = \mathrm{D_{KL}}(p(O)||q(O)) + \mathrm{D_{KL}}(q(O)||p(O)) = \sum_{o \in \mathcal{O}} (p(o) - q(o)) \log \frac{p(o)}{q(o)},$$

the symmetric form of the KL divergence is constructed using the addition method. This metric is also known as the *Jeffreys* divergence [12] and is used to perform our experiments. While we use a discrete sample space $\mathcal{O}$ for the remainder of this paper, there is an extension of our approach to continuous distributions, which replaces the discrete KL divergence with its continuous equivalent.

Practical Considerations: In this section we want to highlight a few aspects to help practitioners to adopt our approach:

- Methods like this, that rely on estimating pdfs (e.g., MIA) allow for a variety of estimation methods to be used, such as histograms or parameter estimation. In Section 4 we show results derived from histograms.
- As the Jeffrys divergence measures a distance, the smallest value indicates the most similar distributions.
- Any scheme similar to the Jeffreys divergence compares only two pdfs, while our method requires to compare two sets of pdfs. To compensate this, we employ the metric of a weighted mean of the Jeffreys divergence values:

$$\mathrm{D}^{\widehat{\Delta \mathsf{k}}}_{\mathsf{i}_1,\mathsf{i}_2,\mathsf{s}} = \sum_{\mathsf{x}=0}^{2^n-1} \left(\mathrm{D_J}\left(f'^{\mathsf{x}}_{\mathsf{i}_1,\mathsf{s}}(O) \,||\, {}^{\widehat{\Delta \mathsf{k}}}f'^{\mathsf{x}}_{\mathsf{i}_2,\mathsf{s}}(O) \right) \cdot \Pr\left[X_{\mathsf{i}_1} = \mathsf{x} \,\middle|\, X_{\mathsf{i}_2} = X_{\mathsf{i}_1} \odot \widehat{\Delta \mathsf{k}} \right] \right).$$

While we introduced our approaches for univariate moments and distributions, an extension to multivariate analyses is straightforward. We provide an example of a multivariate analysis in Section 4.4, where we demonstrate an attack on an *all-or-nothing* secret sharing scheme.

4 Practical Experiments

We used two different platforms to perform our practical analyses: the Xilinx Virtex-II Pro FPGA embedded in a SASEBO [1] board and a multi-purpose smartcard based on an Atmel ATMega163 microcontroller. Four implementations, all employing different masking schemes, were used to evaluate our new approach. Three of these implementations ran on the hardware platform (FPGA), the remaining one was a software solution executed on the smartcard. A LeCroy WP715Zi 1.5GHz oscilloscope equipped with a differential probe was used to collect power consumption traces in the VDD path of both platforms. In the following, we first present our results analyzing the hardware implementations. The case study of the protected software implementation is detailed in Section 4.4, which provides a glance at multivariate collision attacks.

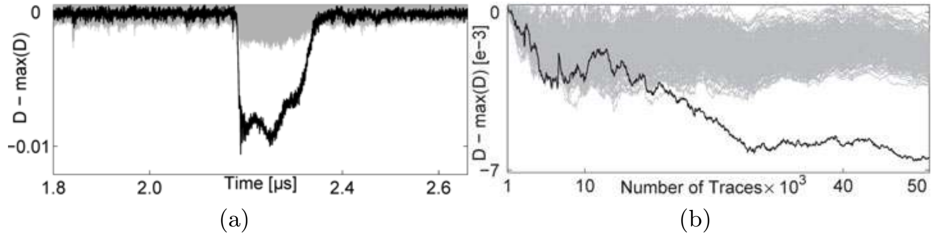


Fig. 3. Result of the collision attack using pdfs on the masked AES implementation based on [8] (a) using 200 000 traces and (b) at point 2.19μs over the number of traces

4.1 Canright-Batina's Masked AES S-Box

In [16] a serialized masked AES encryption is analyzed, where a single masked S-box instance using the design from [8] is used to subsequently process all SubBytes transformations. The interested reader can find an abstract schematic of this architecture in Fig. 7 in the Appendix (architecture is detailed in [16]). The existence of first-order leakage of masked S-boxes implemented in hardware is well-known to the side-channel community [15]. Therefore, a correlation collision attack employing first-order moments (means) can exploit this first-order leakage caused by glitches using around 20 000 measurements. At this, all random masks followed a uniform distribution, and no masks were reused (see [16]).

Since the first-order moments have already shown a dependency on processed data, an analysis of the higher-order moments is not required to perform an attack. Nevertheless, in order to evaluate the feasibility and efficiency of our attack, we implemented the most general form of the attack, the one using pdfs to detect the collisions, on a set of 200 000 measurements. Using histograms with 8 bins we estimated the families of pdfs $\mathfrak{P}_{\mathsf{i}_1}$ and $\mathfrak{P}_{\mathsf{i}_2}$ for two processed portions (here bytes) i_1 and i_2. The result of computing $\mathrm{D}^{\widehat{\Delta\mathsf{k}}}_{\mathsf{i}_1,\mathsf{i}_2,\mathsf{s}}\ \forall\ \widehat{\Delta\mathsf{k}} \in \{0,\ldots,255\}$ for each sample point s, is shown in Fig. 3(a).[3] In addition to the increased complexity of the computations, we find that the attack using the pdfs also requires a slightly higher amount of measurements (cf. Fig. 3(b)). One reason, amongst others, of this is the low accuracy of the pdf estimation by means of histograms.

4.2 Threshold Implementation of PRESENT

Threshold implementations were proposed in [19] and later extended in [20] and [21] to overcome the first-order leakage caused by glitches when masks and masked data are processed by combinational hardware circuits. This scheme is a countermeasure at the algorithm level, and a couple of implementations of the PRESENT cipher based on that have been presented in [22]. We selected *profile 2* of [22], where only the data state is shared using 3 shares and only one instance

[3] For reasons of visualization we actually show the difference of the Jeffreys divergence to the largest observed value, i.e., $\mathrm{D}^{\widehat{\Delta\mathsf{k}}}_{\mathsf{i}_1,\mathsf{i}_2,\mathsf{s}} - \max(\mathrm{D}^{\Delta K}_{\mathsf{i}_1,\mathsf{i}_2,\mathsf{s}})$.

of the shared S-box is used by the design. Fig. 8 in the Appendix sketches the architecture and shows exemplary measurements. So far only CPA attacks using the straight forward power models, i.e., HW and HD, have been presented [22]. Our analysis provides the first collision attack on this architecture.

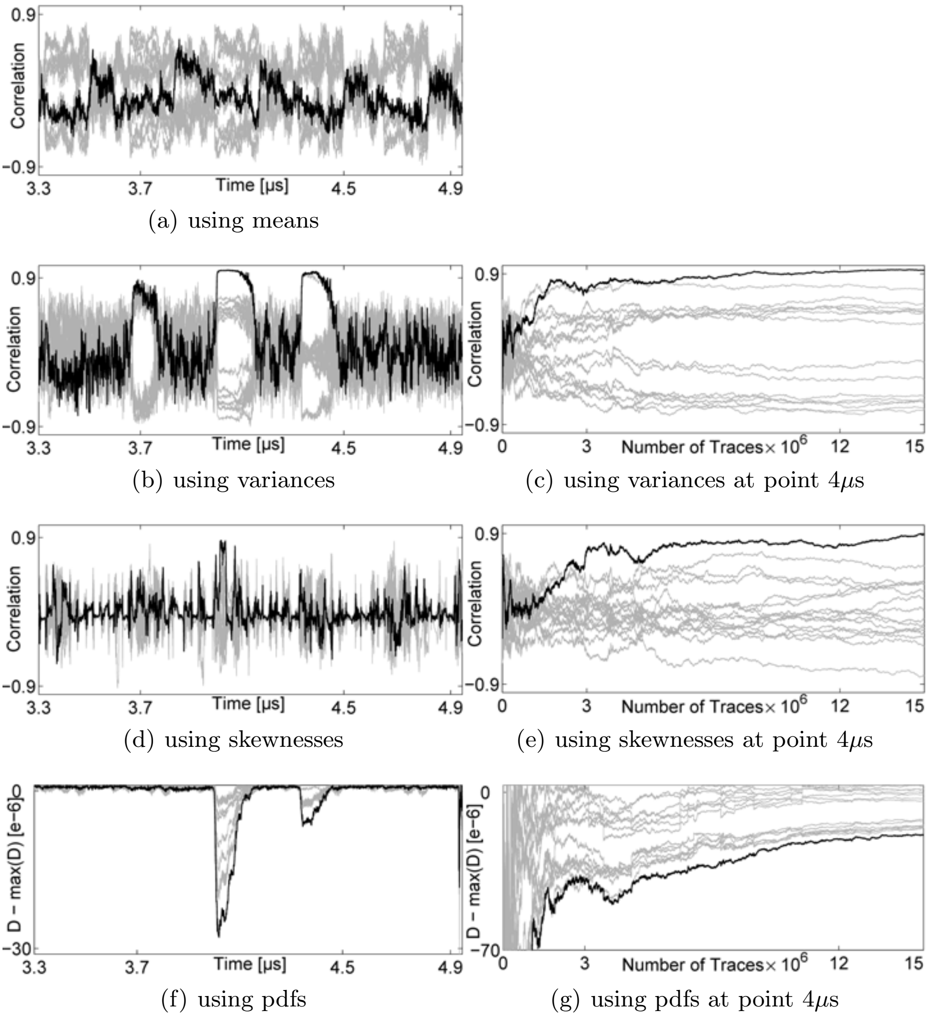
We collected 100 million traces of this implementation using uniformly distributed plaintexts and masks. Two plaintext/key portions (here nibbles), that are consecutively processed, are selected. In addition to the general approach using pdfs, the collision attacks using the first three moments (mean, variance, and skewness) have been performed. The corresponding results are shown in Fig. 4. According to Fig. 4(a) the first-order moments do not show any dependency to the processed values, what confirms the claim given in [21]. However, higher-order moments (see Fig. 4(b) and Fig. 4(d)) are strongly dependent on the unmasked values. As expected, also the attack using pdfs (Fig. 4(f)) allows recovering the secret. Since all attacks need roughly the same number of measurements to succeed, i.e., around 5 million (see Fig. 4(c), Fig. 4(e)), and Fig. 4(g)), analyzing statistical moments is to be preferred over the slower pdf approach. Note that using e.g., second-order moments is equivalent to having a preprocessing step squaring the mean-free traces. Successful attacks using high-order moments thus do not contradict the statement given in [21] that threshold implementations prevent first-order leakage.

4.3 Threshold Implementation of AES

The same countermeasure, i.e., threshold implementation, has been applied to AES in [18]. Although this design does not fulfill all the requirements of a threshold implementation, re-masked registers were employed to provide the missing property of *uniformity* (see [21] for the requirements and their meaning). It has been shown that the final design of [18], which applies several internal PRNGs to provide the required fresh masks, is resistant to correlation collision attacks based on means, even when as much as 400 million measurements are used. However, the authors reported that a MIA attack can exploit the leakage using 80 million measurements. Therefore this is a suitable target to evaluate our new methods using higher-order moments and/or pdfs. Similar to the design targeted in Section 4.2 again only one instance of the shared S-box is used in the analyzed architecture. Moreover, the S-box design is based on a four-stage pipeline, thus leakage may appear in several clock cycles. Again, a schematic illustrating the architecture can be found in the Appendix (Fig. 9).

We have collected 100 million measurements of this design implemented on our FPGA platform. We have selected two portions (here bytes) whose corresponding key-whitened plaintext bytes are processed consecutively. Collision attacks using the pdfs and the second- and third-order moments targeting the linear difference between the two selected key bytes have been performed.[4] The results, which are shown in Fig. 5, reveal a dependency on chosen processed data in the second-order moment, but not in the third-order moment. This might be due to the

[4] We ignored the first-order moment due to the results reported by in [18].



(a) using means

(b) using variances

(c) using variances at point 4μs

(d) using skewnesses

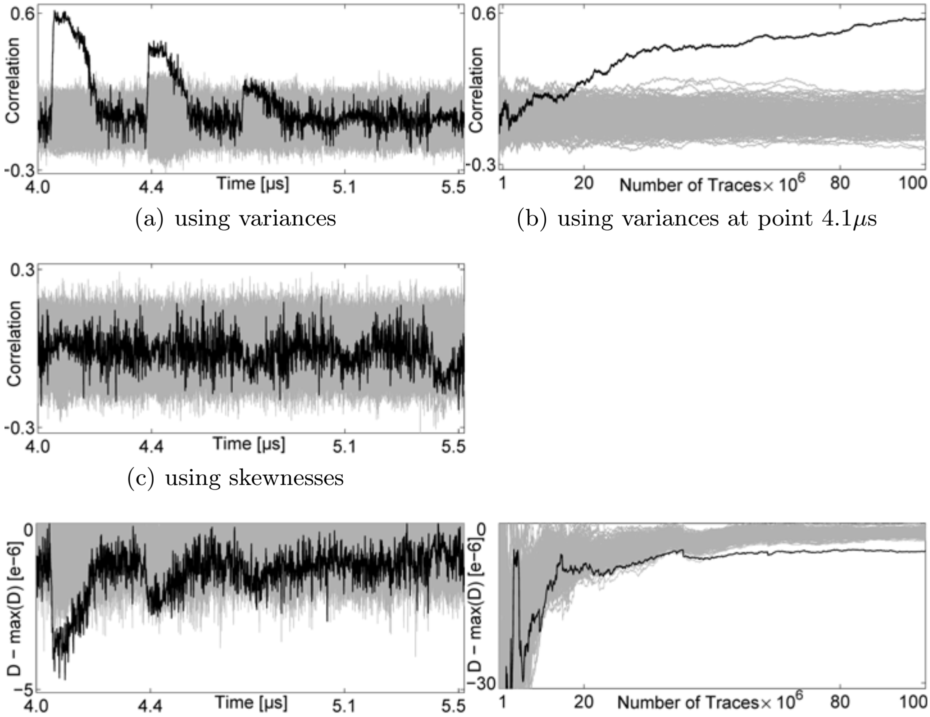
(e) using skewnesses at point 4μs

(f) using pdfs

(g) using pdfs at point 4μs

Fig. 4. Result of the collision attacks on a threshold implementation of PRESENT (left) using 100 million traces, (right) over the number of measurements

re-masked registers not present in the design investigated in Section 4.2. The number of required measurements is also interesting. Compared to that shown in [18] our attack needs around 20 million using variances and 50 million using pdfs (see Fig. 5(b) and Fig. 5(e)). This provides another example that employing statistical moments instead of pdfs is not only faster but also is more efficient with respect to the number of required measurements.



(a) using variances

(b) using variances at point 4.1μs

(c) using skewnesses

(d) using pdfs

(e) using pdfs at point 4.1μs

Fig. 5. Results the collision attacks on a threshold implementation of AES (left) using 100 million traces and (right) over the number of traces

4.4 Boolean Masking in Software

The last case study is a software implementation of the AES based on boolean masking. Two random mask bytes (input mask and output mask) are considered for each plaintext byte (in sum 256 mask bits) at the start of each encryption run. After masking the plaintext bytes using the input masks, the AddRoundkey operation is performed. Afterwards, for each state byte a masked S-box table is constructed in memory, which satisfies the state byte's input and output masks. See Fig. 10 in the appendix for a schematic of the design.

Since every intermediate result is masked by a random value, no univariate attacks can recover a secret. In order to perform a bivariate collision attack using pdfs, we (at the moment) suppose that the two interesting sample points $(\mathsf{s}_1, \mathsf{s}_2)$ in the measurement parts, that denote the time of processing the masked value and the corresponding mask, are known. Then, a set $\mathbb{P}_\mathsf{i}$ consists of joint probability density functions $f^{\mathsf{x}}_{\mathsf{i},\mathsf{s}_1,\mathsf{s}_2}(O_1, O_2)$. The attack then works analogue to the univariate one, except for the comparison step. Here Jeffreys divergence is extended to measure the distance between two joint pdfs as

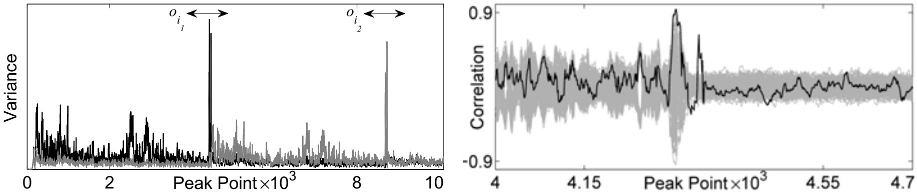
$$\mathrm{D_J}(p(O_1,O_2)||q(O_1,O_2)) = \sum_{o_1\in\mathcal{O}_1}\sum_{o_2\in\mathcal{O}_2}(p(o_1,o_2)-q(o_1,o_2))\log\frac{p(o_1,o_2)}{q(o_1,o_2)}.$$

To use the joint statistical moments, the analysis employs the $(d_1>0, d_2>0)$

$$_{d_1,d_2}\mu^{\mathsf{x}}_{\mathsf{i},\mathsf{s}_1,\mathsf{s}_2} = \frac{1}{q^{\mathsf{x}}_{\mathsf{i}}}\sum_{j=1,x^j_{\mathsf{i}}=\mathsf{x}}^{q}\left(o^j_{\mathsf{i},\mathsf{s}_1}-m^{\mathsf{x}}_{\mathsf{i},\mathsf{s}_1}\right)^{d_1}\left(o^j_{\mathsf{i},\mathsf{s}_2}-m^{\mathsf{x}}_{\mathsf{i},\mathsf{s}_2}\right)^{d_2}.$$

In fact, the attack analyzing $_{1,1}\mu^{\mathsf{x}}_{\mathsf{i},\mathsf{s}_1,\mathsf{s}_2}$ is equivalent to combining the corresponding sample points by means of a "multiplication" prior to the averaging step in a univariate collision attack. The dependencies on higher-moments are familiar from traditional higher-order attacks, which exploit them when applying combining functions.

Since finding the interesting sample points $(\mathsf{s}_1,\mathsf{s}_2)$ in multivariate attacks is always a challenging task, we tried to make use of the moments to mitigate this problem. We collected 250 000 traces from our target implementation using uniformly selected plaintext and mask bytes. Since the construction of the masked S-box tables is time consuming, the measured traces are much longer than the ones of the previously shown case studies. Each trace covers 10 000 clock cycles and was compressed to a vector of 10 000 peaks corresponding to the peaks of the clock cycles. Since the masked value and the mask are processed with a time distance of – most likely – a small number of clock cycles, we defined a window of around 30 clock cycles to sum up adjacent peaks (sliding average). First, we assumed that each measurement part $\boldsymbol{o}^j_i$ covers all summed peak points. Computing the second-order central moments $_2\mu_{\mathsf{i}}$ for two portions i_1 and i_2 and getting the variance of each set at each summed peak point separately led to the two variance curves shown in Fig. 6(a). The graphics clearly exposes the (time) distance between the same process performed on each portion. With this knowledge the measurement parts can be accordingly selected and thus it allows executing a collision attack. The result from a collision attack on second-order moments depicted in Fig. 6(b) confirms our theoretical reasoning and provides evidence of the strength of the attack.



(a) two variances of the 2^nd^-order moments

(b) attack results using 2^nd^-order moments

Fig. 6. Result of the attacks on a software implementation of the AES (boolean masking) after two preprocessing steps: 1) peak extraction, 2) sum over a 30 peak point window

5 Conclusions

The attack presented in this work is fundamentally similar to the correlation collision attack presented in [16]. We extended the scheme to employ higher-order moments and introduced a general form of the attack, which makes use of the distribution of side-channel leakages. As supported by the experimental results, the presented methods allow improving univariate collision attacks. We showed that by slightly increasing the computation complexity (e.g., variance vs. mean) the collision attacks can defeat the security provided by one of the most prominent proposed masking schemes for hardware, i.e., threshold implementations. Additionally, we discussed the possible options to perform multivariate collision attacks using either high-order moments or joint probability distributions. We concluded our case studies analyzing a masked software implementation, and presented a scheme to localize the interesting points for a collision attack employing high-order moments.

The majority of the – usually unprotected – devices have a straightforward and known leakage behavior. Thus, in most cases traditional approaches, e.g., CPA using HW model, can be applied. However, in case that masking countermeasures are applied and the leakage points must be combined the leakage model may not be appropriately guessed and the issue addressed in [26] may become critical. In summary, the collision attacks are an essential tool for security evaluations in situations where the leakage model of the target device is not known and cannot be obtained by profiling.

Acknowledgment. The author would like to thank the anonymous reviewers of CHES 2011 for their helpful comments, Kerstin Lemke-Rust for fruitful discussions, Akashi Satoh and RCIS of Japan for the prompt and kind help in obtaining SASEBOs, and especially Markus Kasper for his great help improving the quality of the paper.

References

1. Side-channel Attack Standard Evaluation Board (SASEBO). Further information are available via, `http://www.rcis.aist.go.jp/special/SASEBO/index-en.html`
2. Bogdanov, A.: Improved Side-Channel Collision Attacks on AES. In: Adams, C., Miri, A., Wiener, M. (eds.) SAC 2007. LNCS, vol. 4876, pp. 84–95. Springer, Heidelberg (2007)
3. Bogdanov, A.: Multiple-Differential Side-Channel Collision Attacks on AES. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 30–44. Springer, Heidelberg (2008)
4. Bogdanov, A., Kizhvatov, I.: Beyond the Limits of DPA: Combined Side-Channel Collision Attacks. IEEE Transactions on Computers (2011), (to appear), A draft version at, `http://eprint.iacr.org/2010/590`
5. Bogdanov, A., Kizhvatov, I., Pyshkin, A.: Algebraic Methods in Side-Channel Collision Attacks and Practical Collision Detection. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 251–265. Springer, Heidelberg (2008)

6. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
7. Brier, E., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)
8. Canright, D., Batina, L.: A Very Compact "Perfectly Masked" S-Box for AES. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 446–459. Springer, Heidelberg (2008); the corrected version at `http://eprint.iacr.org/2009/011`
9. Cha, S.-H.: Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. Journal of Mathematical Models and Methods in Applied Sciences 1, 300–307 (2007)
10. Clavier, C., Feix, B., Gagnerot, G., Roussellet, M., Verneuil, V.: Improved Collision-Correlation Power Analysis on First Order Protected AES. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 49–62. Springer, Heidelberg (2011)
11. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual Information Analysis. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 426–442. Springer, Heidelberg (2008)
12. Jeffreys, H.: An Invariant Form for the Prior Probability in Estimation Problems. Royal Society of London Proceedings Series A 186, 453–461 (1946)
13. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
14. Kullback, S., Leibler, R.A.: On Information and Sufficiency. The Annals of Mathematical Statistics 22(1), 79–86 (1951)
15. Mangard, S., Pramstaller, N., Oswald, E.: Successfully Attacking Masked AES Hardware Implementations. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 157–171. Springer, Heidelberg (2005)
16. Moradi, A., Mischke, O., Eisenbarth, T.: Correlation-Enhanced Power Analysis Collision Attack. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 125–139. Springer, Heidelberg (2010), The extended version at `http://eprint.iacr.org/2010/297`
17. Moradi, A., Mischke, O., Paar, C., Li, Y., Ohta, K., Sakiyama, K.: On the Power of Fault Sensitivity Analysis and Collision Side-Channel Attacks in a Combined Setting. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 292–311. Springer, Heidelberg (2011)
18. Moradi, A., Poschmann, A., Ling, S., Paar, C., Wang, H.: Pushing the Limits: A Very Compact and a Threshold Implementation of AES. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 69–88. Springer, Heidelberg (2011)
19. Nikova, S., Rechberger, C., Rijmen, V.: Threshold Implementations Against Side-Channel Attacks and Glitches. In: Ning, P., Qing, S., Li, N. (eds.) ICICS 2006. LNCS, vol. 4307, pp. 529–545. Springer, Heidelberg (2006)
20. Nikova, S., Rijmen, V., Schläffer, M.: Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 218–234. Springer, Heidelberg (2009)
21. Nikova, S., Rijmen, V., Schläffer, M.: Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. J. Cryptology 24, 292–321 (2011)
22. Poschmann, A., Moradi, A., Khoo, K., Lim, C.-W., Wang, H., Ling, S.: Side-Channel Resistant Crypto for less than 2,300 GE. J. Cryptology 24, 322–345 (2011)

23. Renauld, M., Standaert, F.-X., Veyrat-Charvillon, N., Kamel, D., Flandre, D.: A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 109–128. Springer, Heidelberg (2011)
24. Schramm, K., Leander, G., Felke, P., Paar, C.: A Collision-Attack on AES. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 163–175. Springer, Heidelberg (2004)
25. Schramm, K., Wollinger, T., Paar, C.: A New Class of Collision Attacks and Its Application to DES. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 206–222. Springer, Heidelberg (2003)
26. Veyrat-Charvillon, N., Standaert, F.-X.: Generic Side-Channel Distinguishers: Improvements and Limitations. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 354–372. Springer, Heidelberg (2011)

Appendix

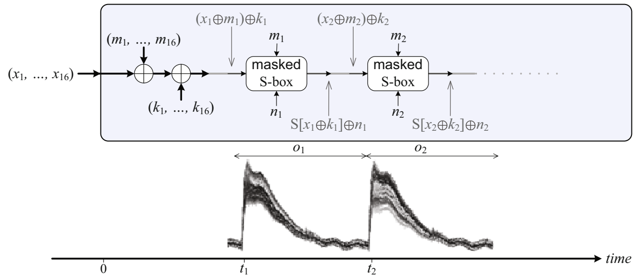
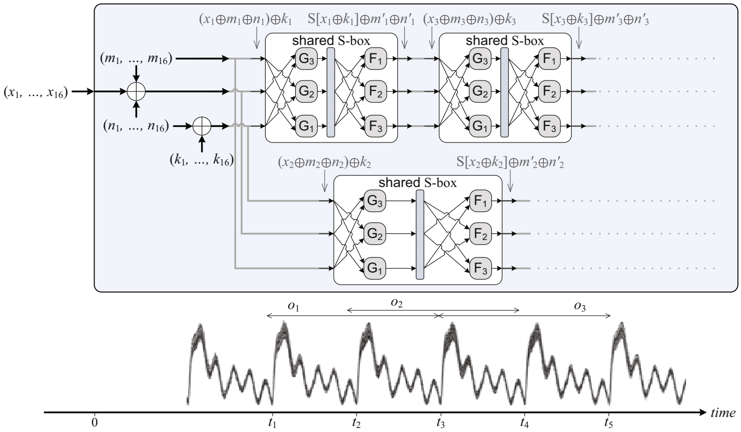


Fig. 7. Schematic of the first case study (a masked AES encryption module using [8])



$(x_1 \oplus m_1 \oplus n_1) \oplus k_1$
$S[x_1 \oplus k_1] \oplus m'_1 \oplus n'_1$
$(x_3 \oplus m_3 \oplus n_3) \oplus k_3$
$S[x_3 \oplus k_3] \oplus m'_3 \oplus n'_3$
shared S-box
shared S-box
$(m_1, ..., m_{16})$
$(x_1, ..., x_{16})$
$(n_1, ..., n_{16})$
$(k_1, ..., k_{16})$
G_3
G_2
G_1
F_1
F_2
F_3
$(x_2 \oplus m_2 \oplus n_2) \oplus k_2$
$S[x_2 \oplus k_2] \oplus m'_2 \oplus n'_2$
shared S-box
o_1
o_2
o_3
0
t_1
t_2
t_3
t_4
t_5
time

Fig. 8. Schematic of the second case study (a threshold implementation of PRESENT taken from [22])

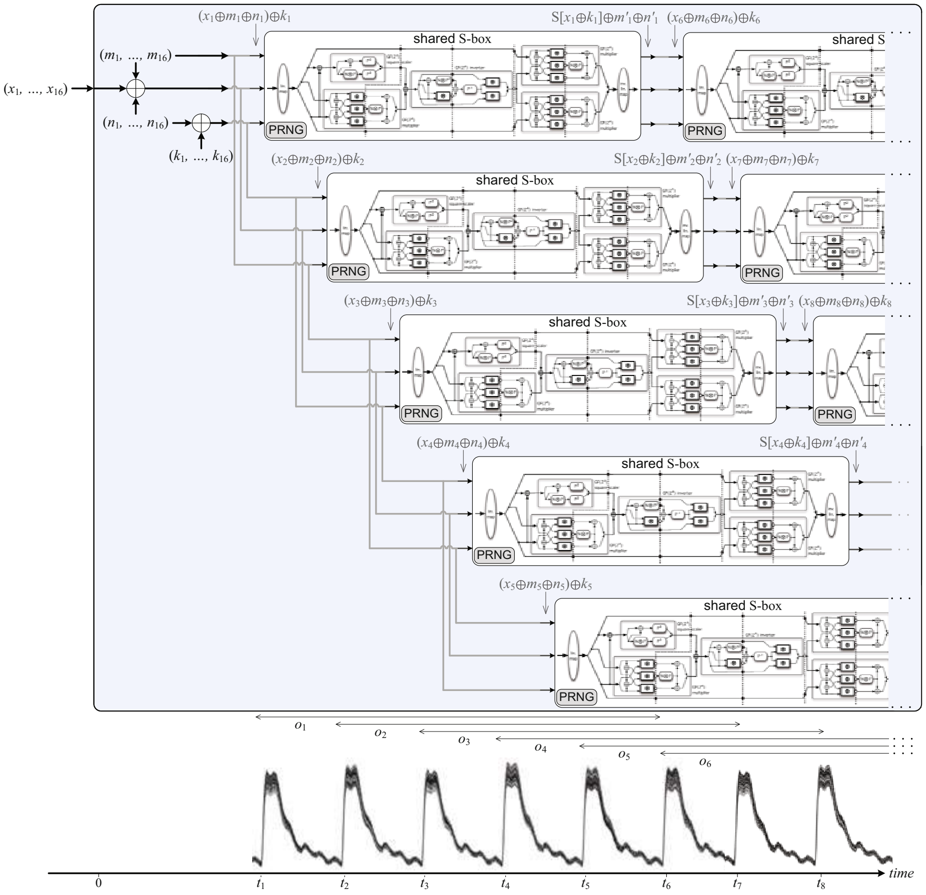

$(x_1, ..., x_{16})$
$(m_1, ..., m_{16})$
$(n_1, ..., n_{16})$
$(k_1, ..., k_{16})$
$(x_1 \oplus m_1 \oplus n_1) \oplus k_1$
$S[x_1 \oplus k_1] \oplus m'_1 \oplus n'_1$
$(x_6 \oplus m_6 \oplus n_6) \oplus k_6$
$(x_2 \oplus m_2 \oplus n_2) \oplus k_2$
$S[x_2 \oplus k_2] \oplus m'_2 \oplus n'_2$
$(x_7 \oplus m_7 \oplus n_7) \oplus k_7$
$(x_3 \oplus m_3 \oplus n_3) \oplus k_3$
$S[x_3 \oplus k_3] \oplus m'_3 \oplus n'_3$
$(x_8 \oplus m_8 \oplus n_8) \oplus k_8$
$(x_4 \oplus m_4 \oplus n_4) \oplus k_4$
$S[x_4 \oplus k_4] \oplus m'_4 \oplus n'_4$
$(x_5 \oplus m_5 \oplus n_5) \oplus k_5$
shared S-box
PRNG
o_1
o_2
o_3
o_4
o_5
o_6
0
t_1
t_2
t_3
t_4
t_5
t_6
t_7
t_8
time

Fig. 9. Schematic of the third case study (a threshold implementation of AES taken from [18])

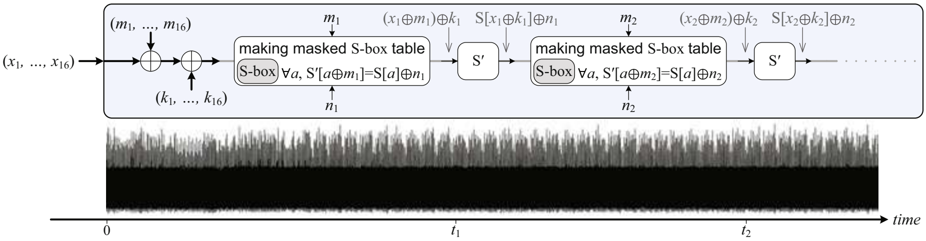

$(x_1, ..., x_{16})$
$(m_1, ..., m_{16})$
$(k_1, ..., k_{16})$
m_1
n_1
making masked S-box table
S-box
$\forall a, S'[a \oplus m_1]=S[a] \oplus n_1$
$(x_1 \oplus m_1) \oplus k_1$
$S[x_1 \oplus k_1] \oplus n_1$
S′
m_2
n_2
$\forall a, S'[a \oplus m_2]=S[a] \oplus n_2$
$(x_2 \oplus m_2) \oplus k_2$
$S[x_2 \oplus k_2] \oplus n_2$
0
t_1
t_2
time

Fig. 10. Schematic of the forth case study (a (boolean) masked software implementation of AES)