A Statistical Model for Higher Order DPA on Masked Devices

A. Adam Ding[1], Liwei Zhang[1], Yunsi Fei[2], and Pei Luo[2]

[1] Department of Mathematics, Northeastern University, Boston, MA 02115
[2] Department of Electrical and Computer Engineering
Northeastern University, Boston, MA 02115

Abstract. A popular effective countermeasure to protect block cipher implementations against differential power analysis (DPA) attacks is to mask the internal operations of the cryptographic algorithm with random numbers. While the masking technique resists against first-order (univariate) DPA attacks, higher-order (multivariate) attacks were able to break masked devices. In this paper, we formulate a statistical model for higher-order DPA attack. We derive an analytic success rate formula that distinctively shows the effects of algorithmic confusion property, signal-noise-ratio (SNR), and masking on leakage of masked devices. It further provides a formal proof for the centered product combination function being optimal for higher-order attacks in very noisy scenarios. We believe that the statistical model fully reveals how the higher-order attack works around masking, and would offer good insights for embedded system designers to implement masking techniques.

Keywords: Side-channel attack, differential power analysis, statistical model

1 Introduction

Differential Power Analysis (DPA) and its variants, Correlation Power Attack (CPA) [1], Mutual Information Attack (MIA) [2], and template attacks [3,4], have been invented to successfully attack cryptographic implementations in many embedded systems [5]. Often these attacks exploit the correlation between the observed measurements and one intermediate data, so-called univariate or first-order attacks. Masking was proposed as an effective countermeasure to protect block cipher systems against first-order attacks. In masking, a random mask M is generated for each execution of the cryptographic algorithm and applied to the internal operations. During the execution, any intermediate data Z is replaced by its masked counterpart $f(Z, M)$ with a carefully designed masking function f. Various masking methods for AES have been investigated [6,7,8,9]. The boolean (exclusive OR) masking $f(Z, M) = Z \oplus M$ is the most commonly used one and will be considered in this paper.

Theoretically, the leakage at any time point of the execution on a boolean-masked device is independent of the secret key, and therefore cannot leak the key. The boolean masking protects cryptosystems against all first-order attacks that use only leakage measurements at one time point (or at multiple time points all related to the same intermediate data). However, higher-order attacks using leakages at more than one time points corresponding to multiple intermediate data are able to reveal the secret key. Particularly, let us consider the second-order attack that uses leakages $L(t_0)$ and $L(t_1)$ at two time points t_0 and t_1 on the device protected by a single mask variable M. A second-order attack can break the protected system by selecting the key k_g that maximizes the correlation between the guessed intermediate data Z^g (before the masking) and a combination function of the two leakages $L(t_0)$ and $L(t_1)$. Two combination functions are studied most in previous literatures. The absolute difference combination function $|L(t_0) - L(t_1)|$ was first proposed by Messerges [10] and analyzed mathematically by Joye et al. [11]. The centered product combination function $[L(t_0) - E(L(t_0))] \times [L(t_1) - E(L(t_1))]$ was proposed by Chari et al. [12] and analyzed by Schramm and Paar [13]. Gierlichs et al. analyzed the higher-order MIA attack using the centered combination function [14]. Oswald et al. compared several combination functions with simulation studies [15]. Prouff et al. provided a mathematical analysis of the second-order attack [16]. They showed that the centered product combination function is the best among product combination functions for CPA, and it is better than the absolute difference combination function in noisy situations. This analysis, however, does not tell if there exists other kinds of combination functions better than the

product combination function. Standaert et al. applied the information theoretical framework to analyze second-order attacks [17]. They showed that when the noise increases, the information leakage of the centered product combination function gets close to the upper bound (the information leakage of the joint distribution), while for small noises, the information leakage of the absolute difference combination function gets close to the upper bound.

Recently, Prouff and Rivain [18] provided a formal security proof for a mask refresh scheme by a secure masking oracle as a leakage-resilient cryptographic primitive. Our work does not consider such sophisticated mask refreshing scheme, but attempts to bound the success rate of higher-order attack on standard and practical masking schemes. Previous side-channel modeling and analysis work [19] derives a simple success rate formula for first-order DPA attack on unmasked devices, which is explicitly dependent on the algorithmic confusion coefficients introduced in [20]. The formula shows the effect of both implementation-determined signal-to-noise-ratio (SNR) and algorithmic confusion properties. They demonstrated that the formula conforms with the empirical single-bit DPA attacks on DES and AES algorithms.

Our contributions: In this paper, we adopt the algorithmic confusion analysis and apply it to higher-order attacks on masked devices and derive an explicit success rate formula. The analytical formula allows us to decouple and quantify the effect of the algorithmic confusion properties, SNR, and masking on the effectiveness of power analysis attacks, which will be useful to system designers when designing, implementing, and evaluating side-channel attack resistant cryptosystems. We will formally prove in this paper, for the first time, that the centered product combination function (CPCF) attack is the best possible combination function attack in noisy situations.

The rest of the paper is structured as follows. Section 2 presents some preliminaries on which our statistical model for higher-order DPA is based. Section 3 derives an analytical model for second-order DPAs and also extends to general higher-order attacks. We then use numerical studies on both real measurement data and synthetic data in Section 4 to validate the derived model. More discussions and conclusions are given in Section 5.

2 Preliminaries

2.1 Success Rate of Maximum Likelihood (ML) Attacks

SCA on a cryptographic system utilizes the correlation between the noisy physical leakage observation L and a key-sensitive intermediate value $Z(X,k)$ to reveal the secret key k, where X denotes a known input plaintext (or ciphertext). We denote $p(L|k)$ as the conditional probability density function (pdf) for L given k is the true key. With n independent realizations of L, $l_1, ..., l_n$, the most powerful side-channel statistical test is the maximum likelihood (ML) test [21]:

$$\hat{k} = \underset{k_g \in S}{argmax} \frac{1}{n} \sum_{i=1}^{n} log[p(l_i|k_g)] \tag{1}$$

Here k_g denotes a guessed key and $S = \{k_1, ..., k_{N_k}\}$ denotes the set of N_k candidate keys. The secret key embedded in the system is denoted as k_c. We define:

$$\Delta(k_c, k_g) = \frac{1}{n} \sum_{i=1}^{n} [\log p(l_i|k_c) - \log p(l_i|k_g)] \tag{2}$$

as the difference between the two likelihoods for k_c and k_g. With $(N_k - 1)$ incorrect keys, we have a $(N_k - 1)$-dimensional vector, $\tilde{\Delta}$, with an entry $\Delta(k_c, k_g)$ for each k_g. The ML attack (1) succeeds when n is large enough to yield all the entries of $\tilde{\Delta}$ positive. We denote $\tilde{\Delta}_1$ as $\tilde{\Delta}$ with only one leakage observation l_1, and the mean and variance of $\tilde{\Delta}_1$ are a vector, $\boldsymbol{\mu}$, and a $(N_k - 1) \times (N_k - 1)$ matrix, $\boldsymbol{\Sigma}$, respectively. With n independent realizations of L, $l_1, ..., l_n$, according to the Central Limit Theorem [22], $\tilde{\Delta}$ converges

in law to the $(N_k - 1)$-dimensional Gaussian distribution, $N(\boldsymbol{\mu}, \boldsymbol{\Sigma}/n)$. The overall *s*uccess rate of the ML attack, defined as the probability that $\tilde{\Delta}$ is a non-negative vector given n, is therefore:

$$SR = \Phi_{N_k-1}(\sqrt{n}\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}) \tag{3}$$

where $\Phi_{N_k-1}(\boldsymbol{x})$ is a known function, the cumulative distribution function (cdf) of the $(N_k - 1)$-dimensional standard Gaussian distribution. Equation (3) holds generally for most SCA, while the mean vector and variance matrix would vary for different attacks. We found that the entries in the mean vector $\boldsymbol{\mu}$ are in fact the conditional entropies similar to those defined in the seminal work of mutual information analysis [21]. However, the success rate formula in (3) considers not only the effect of the mean vector $\boldsymbol{\mu}$, but also the variance matrix $\boldsymbol{\Sigma}$ on SCAs. The mean $\boldsymbol{\mu}$ reflects the overall system side-channel signal and $\boldsymbol{\Sigma}$ reflects the system noise. The term $\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}$ can be taken as the system signal-to-noise ratio (SNR).

The higher-dimension Gaussian distribution $\Phi_{N_k-1}(\boldsymbol{x})$ in (3) is the asymptotic limit of ML-attack statistics coming from the Central Limit Theorem, and is independent of the actual noise distribution in the system leakage. Hence, formula (3) is general and does not require any assumption on the noise distribution. When assuming Gaussian power noise as in [3,23], the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can have analytic forms constituted by algorithmic properties as defined in [19,20] and side-channel SNR. In this paper, we also consider other noise distributions, like Laplace, in Section 3.2 and present the results in Section 4.3.

2.2 First-order Power Leakage Model on Unmasked Devices

For a cryptographic device, a commonly used linear power leakage model is:

$$L = c + \varepsilon V + \sigma r \tag{4}$$

with r as a standard Gaussian noise, $N(0, 1)$, and $V = V(X, k_c)$ is the select function on the intermediate data Z that depends on the known input X and the secret key k_c. At a leakage time point corresponding to Z's switching, L is a univariate random variable. Here c is a constant, representing the base level power consumption of the system, which is independent of both operations and data. The ε reflects the side-channel signal strength and σ is the standard deviation of power measurements, i.e., noise from both measurement and other parts of the device. The side channel signal-to-noise ratio (SNR) is defined as $\delta = \varepsilon/\sigma$. Under this model, the probability density function $p(L|k) = \phi(\frac{L-c-\varepsilon V(X,k)}{\sigma})$ with $\phi(\cdot)$ as the pdf of the standard Gaussian distribution. For a single-bit DPA, $V(X, k)$ is chosen as one bit of the non-linear SBox output $Z = SBox(X, k)$. The ML-attack with unknown parameters (c, ε, σ) is equivalent to the distance-of-means (DoM) attack that selects the key k_g to maximize the DoMs. For multi-bit CPA, often $V = H(Z)$ where $H(Z)$ is the Hamming weight (or distance) of the SBox output Z. The ML-attack with unknown parameters (c, ε, σ) is equivalent to choosing the key k_g that maximizes the Pearson's correlation between L and $V^g = H(Z^g) = H[SBox(X, k_g)]$. That is, the Hamming weight power model results in the Correlation Power Attack (CPA).

2.3 First-order DPA and CPA Models on Unmasked Devices with Confusion Coefficients

In general, the physical power leakage L is affected by both the implementation and algorithm. To measure the effect of the algorithm, Luo and Fei [20] introduced the notion of confusion coefficients for single-bit DPA to reveal the distance between keys in terms of side-channel leakage. Let $S = \{k_1, ..., k_{N_k}\}$ denote the set of N_k candidate keys. The *confusion coefficient* κ over any two keys (k_i, k_j) is defined as:

$$\kappa = \kappa(k_i, k_j) = \Pr\left[(V|k_i) \neq (V|k_j)\right] \tag{5}$$

Here V is a chosen bit of the SBox output $Z = SBox(X, k)$.

Fei et al. [19] further showed that the success rate of the DoM attack follows (3) and the mean vector $\boldsymbol{\mu}$ and variance matrix $\boldsymbol{\Sigma}$ can be explicitly expressed in confusion coefficients and the SNR $\delta = \varepsilon/\sigma$ as:

$$\boldsymbol{\mu} = \frac{1}{2}\delta^2\boldsymbol{\kappa};\ \ \boldsymbol{\Sigma} = \delta^2\boldsymbol{K} + \frac{1}{4}\delta^4(\boldsymbol{K} - \boldsymbol{\kappa}\boldsymbol{\kappa}^T). \tag{6}$$

Here $\boldsymbol{\kappa}$ is a (N_k-1)-dimensional *confusion vector* with elements $\kappa(k_c, k_{g_i})$, $i = 1, ..., N_k - 1$, defined in Equation (5), and $\boldsymbol{K}$ is a $(N_k-1)\times(N_k-1)$ *confusion matrix* that consists of three-way confusion coefficients:

$$\varkappa_{ij} = \kappa(k_c, k_{g_i}, k_{g_j}) = \Pr\left[V|k_{g_i} = V|k_{g_j}, V|k_{g_c} \neq V|k_c\right] = \tfrac{1}{2}[\kappa(k_c, k_{g_i}) + \kappa(k_c, k_{g_j}) - \kappa(k_{g_i}, k_{g_j})]. \quad (7)$$

The confusion analysis is extended to CPA in [24]. For first-order CPA that exploits leakage by multiple bits of an SBOX output, V in (4) is the Hamming weight (or distance) of the SBox output $Z = SBox(X, k)$. The ML-attack's success rate also follows (3) but with:

$$\boldsymbol{\mu} = \frac{1}{2}\delta^2\boldsymbol{\kappa};\ \ \boldsymbol{\Sigma} = \delta^2\boldsymbol{K} + \frac{1}{4}\delta^4(\boldsymbol{K}^* - \boldsymbol{\kappa}\boldsymbol{\kappa}^T). \quad (8)$$

where the definition of *confusion vector* $\boldsymbol{\kappa}$ is the same as before. However, its element, confusion coefficient, is more general:

$$\kappa(k_c, k_{g_i}) = E[(V|k_c - V|k_{g_i})^2] \quad (9)$$

Here $\kappa(k_c, k_{g_i})$ is no longer $Pr(V|k_c \neq V|k_{g_i})$, because $V = H[SBox(X, k)]$ takes values among $\{0, 1, 2, \cdots, b\}$ for a b-bit SBox output. In the variance matrix, there are two $(N_k-1)\times(N_k-1)$ *confusion matrices*, $\boldsymbol{K}$ and $\boldsymbol{K}^*$, with elements:

$$\varkappa_{ij} = \kappa(k_c, k_{g_i}, k_{g_j}) = E[(V|k_c - V|k_{g_i})(V|k_c - V|k_{g_j})] = \frac{1}{2}[\kappa(k_c, k_{g_i}) + \kappa(k_c, k_{g_j}) - \kappa(k_{g_i}, k_{g_j})], \quad (10)$$

$$\varkappa^*_{ij} = \kappa^*(k_c, k_{g_i}, k_{g_j}) = E[(V|k_c - \frac{b}{2})^2(V|k_c - V|k_{g_i})(V|k_c - V|k_{g_j})]. \quad (11)$$

When $b = 1$, these two matrices are the same, i.e., the first-order $\boldsymbol{K}$ with elements in (7) for single-bit DPA.

When δ is small, i.e., noisy situations, the higher-order δ^4 term can be ignored and the variance in (8) can be simplified to $\boldsymbol{\Sigma} = \delta^2\boldsymbol{K}$. Then the success rate becomes a simplified version as in [25]:

$$SR = \Phi_{N_k-1}(\frac{\sqrt{n}\delta}{2}\boldsymbol{K}^{-1/2}\boldsymbol{\kappa}). \quad (12)$$

3 Statistical Model for Higher-order DPA on Masked Devices

In this section, we first present the second-order power leakage model for masked devices. We then derive an approximation of the ML-test statistic under noisy situations to find the correspondingly equivalent optimal second-order DPA. Under the Hamming Weight leakage model, this turns out to be the CPCF attack. Finally, we derive the success rate formula for the optimal second-order DPA with explicit constituent terms of algorithmic properties and SNR. In the end, these derivations are generalized to higher J-th order masking models with J random masks.

3.1 Second-order Power Leakage Model on Masked Devices

We consider the boolean masking scheme where a secret intermediate data Z is masked by one random mask M. The mask M takes value uniformly in the set $\mathcal{M}$. Therefore, the masked variable $Z \oplus M$ follows a uniform distribution on $\mathcal{M}$, independent of $Z = Z(X, k_c)$, according to the property of the exclusive OR operation. Hence, the leakage at any selected time point only leaks the random $Z \oplus M$ and no longer leaks any key information, and therefore the first-order DPA will fail.

However, often the power consumption at another time point can leak the mask M, and can be combined with the leakage on the masked intermediate variable $Z \oplus M$ to break masked devices. We assume that t_0 and t_1 are the peak leakage time points for $V_0 = V_0(Z \oplus M)$ and $V_1 = V_1(M)$ respectively. Note here V_0 is key-sensitive and V_1 is key-independent. We denote $V^g_{M,0} = V_0(Z^g \oplus M)$ with $Z^g = Z(X, k_g)$ under key guess k_g. The ML-attack on the masked device is still of the same form as in (1)

with the log-likelihood $\frac{1}{n}\sum_{i=1}^{n}\log p(\boldsymbol{l}_i|k_g)$, taking a two-dimensional vector leakage input $\boldsymbol{l}_i = (l_{i,0}, l_{i,1})$, rather than a scalar one as in univariate (first-order) ML attack. Assuming the leakages at the two time points are independent of each other, the log-likelihood becomes

$$\frac{1}{n}\sum_{i=1}^{n}\log[p(\boldsymbol{l}_i|k_g)] = \frac{1}{n}\sum_{i=1}^{n}\log[\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} p_0(l_{i,0}|k_g, m)p_1(l_{i,1}|m)]. \tag{13}$$

The above log-likelihood expression involves an iteration of m over all possible mask values: $\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}}$. We use the notation E_m to denote such expectation over M. Hence we rewrite the ML-test statistic as:

$$T^g_{ML} = \frac{1}{n}\sum_{i=1}^{n}\log p(\boldsymbol{l}_i|k_g) = \frac{1}{n}\sum_{i=1}^{n}\log\{E_m[p_0(l_{i,0}|k_g, m)p_1(l_{i,1}|m)]\}. \tag{14}$$

The linear operation E_m above prevents separating the factors inside the log into sums. This results in a mixture distribution density function, and is computationally intensive.

Under the commonly used leakage power model, the power consumptions at the two time points in a masked device are:

$$L_j = L(t_j) = c_j + \varepsilon_j V_j + \sigma_j r_j, \qquad j = 0, 1. \tag{15}$$

where the noises r_0 and r_1 are independent standard Gaussian noise, $N(0, 1)$. For n executions of the cryptographic algorithm, each with a distinct input x_i and a random mask m_i, $i = 1, ..., n$, we denote the n realizations of (Z, V_0, V_1, r_0, r_1, L_0, L_1) as (Z_i, $V_{i,0}$, $V_{i,1}$, $r_{i,0}$, $r_{i,1}$, $l_{i,0}$, $l_{i,1}$). Then under model (15), the ML-test statistic from Equation (14) results from the mixture distribution:

$$T^g_{ML} = \frac{1}{n}\sum_{i=1}^{n}\log\{E_m[\phi(r^g_{m,i,0})\phi(r_{m,i,1})]\} \tag{16}$$

where $\phi(x) = e^{-x^2/2}/\sqrt{2\pi}$ is the pdf function of standard Gaussian distribution. $r^g_{m,i,0} = \frac{l_{i,0}-c_0-\varepsilon_0 V^g_{m,i,0}}{\sigma_0} = r_{i,0} + \delta_0(V_{i,0} - V^g_{m,i,0})$ is for time point t_0, where $V_{i,0} = V_0(Z^c_i \oplus m_i)$ is the correct select function at the point with the specific m_i, and $V^g_{m,i,0} = V_0(Z^g_i \oplus m)$ is the guessed one under k_g given a random m. $r_{m,i,1} = \frac{l_{i,1}-c_1-\varepsilon_1 V_{m,1}}{\sigma_1} = r_{i,1} + \delta_1(V_{i,1} - V_{m,1})$ is for time point t_1 (key-independent), where $V_{i,1}$ is the correct select function at the point with m_i and $V_{m,1} = V_1(m)$ is the select function given a random m. $\delta_j = \varepsilon_j/\sigma_j$ denotes SNR for $j = 0, 1$.

The ML-attack select the key k_g that maximizes the statistic T^g_{ML} in (16). In contrast, the centered product combination function (CPCF) attack select the key k_g that maximizes the statistic

$$\widetilde{T}^g = \frac{1}{n}\sum_{i=1}^{n}\widetilde{C}_i f(Z^g_i), \tag{17}$$

where $f(Z^g_i) = E_m(V^g_{m,i,0}V_{m,1})$, $\widetilde{C}_i = \tilde{l}_{i,0}\tilde{l}_{i,1}$ with centered leakage measurements at the two time points as

$$\tilde{l}_{i,j} = [l_{i,j} - E(L_{i,j})]/\sigma_j = r_{i,j} + \delta_j[V_{i,j} - E(V_{i,j})], \qquad \text{for} \qquad j = 0, 1. \tag{18}$$

Here $E(\cdot)$ denotes the unconditional expectation over all three sources of random variation in the leakage model (15): (a) the random mask M, (b) the random input X, and (c) the random noise vector $\boldsymbol{r} = (r_0, r_1)$. This is different from the $E_m(\cdot)$ operation defined earlier, which is in fact a conditional expectation integrating out the random variation from the first source (a) only.

In section 3.2, we prove the equivalence between the ML-attack (16) and the CPCF attack (17) in noisy situations summarized in the following Theorem 1. Then under Hamming Weight model, the prediction function $f(Z^g_i)$ can be simplified. We then derive an explicit success rate formula for the equivalent attack in terms of the confusion coefficients in section 3.3.

Theorem 1 *Under the second-order leakage model (15), as the noises increase, $\delta \to 0$, the ML-attack is asymptotically equivalent to the CPCF attack.*

The main idea of the proof is to check the Taylor expansion of the T^g_{ML} in (16) under noisy situations ($\delta \to 0$). Many leading terms in the expansion are in fact key-independent constants. The first leading key-dependent term is proportional to $\delta_0\delta_1\widetilde{T}^g$ in (17).

The previous work [26] also analyzes second-order ML attack, and approximates the Gaussian mixture density (16) by a bivariate Gaussian distribution using the techniques in [27]. They show that CPCF attack maximizes the likelihood for the best Gaussian approximation. However, there is no measure of the information lost in using such a Gaussian approximation. We prove formally that the CPCF attack approximation of ML-attack becomes exact in noisy situations ($\delta \to 0$), and would indeed provide the same security bound asymptotically.

3.2 Approximate ML-attack Statistic Under Noisy Situations

While the ML-attack is the strongest statistical attack, the E_m operation in calculating (16) is time-consuming with complexity $O(|\mathcal{M}|)$. Particularly, the complexity increases exponentially with the order J of masking, as $O(|\mathcal{M}|^J)$. Hence, the exact ML-attack is computationally prohibitive in higher-order masking, say, $J = 8$. In practice, adversaries can use attacks based on some combination functions to avoid the E_m operation. Since the E_m of powers of $r^g_{m,i,0}$ and $r_{m,i,1}$ can be known with explicit forms, we wish to approximate the ML-attack statistic and therefore find practical but yet asymptotic equivalent attack to the ML-test. This is achieved by taking a Taylor expansion of (16).

We aim to extract the key-dependent components from the ML test statistic. We observe that, when the SNRs at both time points approach zero, by model (15) T^g_{ML} becomes key-independent:

$$T_0 = \frac{1}{n}\sum_{i=1}^{n} \log[\frac{1}{2\pi}e^{-\frac{r^2_{i,0}+r^2_{i,1}}{2}}]$$

with the noises $r_{i,j} = (l_{i,j} - c_j)/\sigma_j$ for $j = 0, 1$. Removing this constant from (16), we get the rest key-sensitive part of the test statistic:

$$T^g_{ML} - T_0 = \frac{1}{n}\sum_{i=1}^{n} \log(S^g_i) = \frac{1}{n}\sum_{i=1}^{n} \log\{E_m[e^{R^g_{m,i}}]\} \tag{19}$$

where $S^g_i = E_m[e^{R^g_{m,i}}]$, $R^g_{m,i} = -\frac{1}{2}(A^g_{m,i} + A_{m,i,1}) = O(\delta)$ and $\delta = max(\delta_0, \delta_1)$ with

$$\begin{array}{l} A^g_{m,i} \;= (r^g_{m,i,0})^2 - r^2_{i,0} = 2\delta_0(V_{i,0} - V^g_{m,i,0})r_{i,0} + \delta^2_0(V_{i,0} - V^g_{m,i,0})^2 = O(\delta_0); \\ A_{m,i,1} = r^2_{m,i,1} - r^2_{i,1} = 2\delta_1(V_{i,1} - V_{m,1})r_{i,1} + \delta^2_1(V_{i,1} - V_{m,1})^2 = O(\delta_1). \end{array} \tag{20}$$

When $\delta \to 0$, we have the Taylor expansion $S^g_i = E_m[e^{R^g_{m,i}}] = 1 + E_m(R^g_{m,i}) + O(\delta^2)$. However, this leading term $E_m(R^g_{m,i}) = E_m[-\frac{1}{2}(A^g_{m,i} + A_{m,i,1})]$ does not contribute to the key selection since it is a key-independent constant. This comes from a simple but very useful fact summarized as:

Lemma 1 *For any statistic S^g of the leakage measurements at a single time point, $E_m(S^g)$ is independent of key k_g.*

The above Lemma is due to the fact that, as m iterates over the range $\mathcal{M}$, $Z^g \oplus m$ also iterates over $\mathcal{M}$. So the sum over the range $\mathcal{M}$ would be independent of the actual value of Z^g. Hence, after the $E_m(\cdot)$ operation, any statistic of $Z^g \oplus m$ becomes independent of Z^g (hence independent of key k_g).

Lemma 1 implies that $E_m(R^g_{m,i})$ is key-independent which is the sum of two statistics on two different time points. We need to take in the next higher-order term in the Taylor expansion to find the leading

key-sensitive term in the ML-attack statistic, $S_i^g = 1 + E_m[R_{m,i}^g] + \frac{1}{2}E_m[(R_{m,i}^g)^2] + O(\delta^3)$. The key-sensitive part in $E_m[(R_{m,i}^g)^2]$ is $(-1/2)^2 E_m[2A_{m,i}^g A_{m,i,1}]$ after applying Lemma 1 again. Combining this with $log(S_i^g) = (S_i^g - 1) - (S_i^g - 1)^2 + O(\delta^3)$, we have

$$log(S_i^g) = A_i + \frac{1}{4}E_m[A_{m,i}^g A_{m,i,1}] + O(\delta^3), \tag{21}$$

with a key-independent constant A_i. From (21), we get:

$$T_{ML}^g = A^* + T^g + O(\delta^3), \text{ with } T^g = \frac{1}{4n}\sum_{i=1}^{n} E_m[A_{m,i}^g A_{m,i,1}], \tag{22}$$

where A^* is a constant, $A_{m,i}^g$ and $A_{m,i,1}$ are defined in (20). That is, *the ML-attack asymptotically (when $\delta \to 0$) is equivalent to selecting the key k_g that maximizes the test statistic T_g in (22).*

Remark 1: The error in the Taylor expansion of $e^{R_{m,i}^g}$ in (19) by $1 + R_{m,i}^g + (R_{m,i}^g)^2/2$ is bounded by $\frac{max(1,e^{R_{m,i}^g})}{6}|R_{m,i}^g|^3$. From the definition of $R_{m,i}^g$ in (20), considering that V is bounded and $R_{m,i}^g$ is linear in r_i, $E_m[\frac{max(1,e^{R_{m,i}^g})}{6}|R_{m,i}^g|^3]$ has finite moments for each i. By law of large numbers [28] there is a constant Q such that, with probability one for large n,

$$\frac{1}{n}\sum_{i=1}^{n} E_m[\frac{max(1,e^{R_{m,i}^g})}{6}|R_{m,i}^g|^3] \le Q\delta^3$$

uniformly for small enough δ. Hence the above approximation (22) holds uniformly at the rate of $O(\delta^3)$ for small δ with probability one for large n.

Now we try to simplify the expression of test statistic T^g. We can rewrite (20) as

$$\begin{array}{rl} A_{m,i}^g & = A_{i,0} - 2\delta_0\tilde{l}_{i,0}V_{m,i,0}^g + \delta_0^2 V_{m,i,0}^g[V_{m,i,0}^g - 2E(V_{i,0})]; \\ A_{m,i,1} & = A_{i,1} - 2\delta_1\tilde{l}_{i,1}V_{m,1} + \delta_1^2 V_{m,1}[V_{m,1} - 2E(V_{i,1})], \end{array} \tag{23}$$

where $A_{i,0}$ and $A_{i,1}$ are constants independent of guessed key k_g and the random masks m. Using (23), we find the leading key-sensitive term in $E_m[A_{m,i}^g A_{m,i,1}]$ relates to the CPCF attack statistics (17),

$$E_m[(-2\delta_0\tilde{l}_{i,0}V_{m,i,0}^g)(-2\delta_j\tilde{l}_{i,j}V_{m,j})] = 4\delta_0\delta_1\widetilde{C}_i f(Z_i^g) + O(\delta^3) = 4n\widetilde{T}^g + O(\delta^3).$$

Plug this into (22), and the approximate ML-test statistic T^g becomes $T^g = B^* + \delta_0\delta_1\widetilde{T}^g + O(\delta^3)$, with B^* as another key-independent constant. Hence we establish the equivalence to (17) and Theorem 1.

Given specific $V_0(\cdot)$ and $V_1(\cdot)$ functions, $f(Z_i^g) = E_m(V_{m,i,0}^g V_{m,1})$ is a deterministic function and E_m operation can be skipped over using algebraic properties. We further simplify the test statistic $\widetilde{T}^g$ in (17), eliminating the iteration over $\mathcal{M}$ and finding an explicit formula for $f(Z_i^g)$ under the Hamming Weight power leakage model:

$$V_0(Z, M) = H(Z \oplus M) \qquad \text{and} \qquad V_1(M) = H(M). \tag{24}$$

By Lemma 21 in [16], for any b-bits random mask M, $E_m[H(Z \oplus M)H(M)|Z] = -\frac{1}{2}H(Z) + b/4$. Applying this formula to (24), we get

$$f(Z_i^g) = E_m(V_{m,i,0}^g V_{m,1}) = -\frac{1}{2}H(Z_i^g) + constant.$$

Therefore, under the Hamming Weight power leakage model, the CPCF attack maximizes

$$\widetilde{T}^g = -\frac{\delta_0\delta_1}{2n}\sum_{i=1}^{n}\widetilde{C}_i H(Z_i^g). \tag{25}$$

Remark 2: The above derivations for approximate ML attacks assume that the system parameters $(\boldsymbol{c}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma})$ are all known, and therefore the theoretically strongest attack can just plug these parameters into (18) with $E(L_{i,j}) = c_j + \varepsilon_j \frac{b}{2}$. The real applicable CPCF attack does not know these parameters and needs to estimate $E(L_{i,j}) = \sum_{i=1}^{n} l_{i,j}/n$ and $(\boldsymbol{c}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma})$ just based on the power data. We therefore consider such two attacks: (a) the theoretical strongest approximate ML attack and (b) the real second-order attack. The real attack (b) should be less powerful due to parameter estimation using finite power data. Similar to the previous work on DPA and CPA modeling, the second-order approximate ML attack (a) provides a theoretical bound for the real CPCF attack (b). We will derive the success rate formula for (a) only in Section 3.3, and will compare these two attacks in Section 4.2.

3.3 The Explicit Asymptotic Success Rate Formula for Second-order Attack

We now derive an explicit asymptotic success rate of the approximate second-order ML-attack, in terms of algorithm confusion coefficients and SNRs. For the test selecting $argmax_{k_g \in S} T^g$, the $\Delta(k_c, k_g)$ in Equation (2) becomes:

$$\Delta(k_c, k_g) = T^c - T^g = \frac{\delta_0 \delta_1}{n}(-\frac{1}{2}) \sum_{i=1}^{n} \widetilde{C}_i (H(Z_i^c) - H(Z_i^g)). \tag{26}$$

As explained in Section 2, the asymptotic success rate of the ML-attack is $\Phi_{N_k-1}(\sqrt{n}\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu})$ given in Equation (3). The mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$ are for the $(N_k - 1)$-dimensional vector according to (26). Then i-th element in $\boldsymbol{\mu}$ is

$$\mu_i = (\frac{1}{2})^3 (\delta_0 \delta_1)^2 \kappa(k_c, k_{g_i}). \tag{27}$$

For the ij-th element of the covariance $\boldsymbol{\Sigma}$, we keep the leading term and simplify it as:

$$\sigma_{ij} = (\frac{1}{2})^2 (\delta_0 \delta_1)^2 \kappa(k_c, k_{g_i}, k_{g_j}). \tag{28}$$

The detailed calculations for (27) and (28) are provided in Appendix A.

Therefore, under the power leakage model (15) and (24) for masked devices, the asymptotic success rate for the second-order ML-attack is given by:

$$SR = \Phi_{N_k-1}(\frac{\sqrt{n}\delta_0\delta_1}{4} \boldsymbol{K}^{-1/2} \boldsymbol{\kappa}). \tag{29}$$

Here definitions of the confusion vector $\boldsymbol{\kappa}$ and the confusion matrix $\boldsymbol{K}$ are exactly the same as those for CPA attack on unmasked devices with elements given in (9) and (10). Compared to the simplified formula for CPA in (12), the second-order attack involves the same algorithmic confusion properties ($\boldsymbol{\kappa}$ and $\boldsymbol{K}$), and the product of two SNRs ($\frac{\delta_0}{2}$ and $\frac{\delta_1}{2}$ at the two time points) introduced by the masking.

The success rate formula (29) provides a good approximation of the true success rates when the noise is high. To also approximate well for moderate noises ($\delta \leq 1$), we keep all the terms in elements of the variance matrix $\boldsymbol{\Sigma}$ without approximation (in Appendix B) and get the complete theoretical model:

$$\boldsymbol{\mu} = \frac{1}{8}\delta_0^2\delta_1^2\boldsymbol{\kappa};\ \ \boldsymbol{\Sigma} = \frac{1}{4}\delta_0^2\delta_1^2(1 + \frac{b}{4}\delta_0^2)(1 + \frac{b}{4}\delta_1^2)\boldsymbol{K} + \frac{1}{64}\delta_0^4\delta_1^4(8\boldsymbol{K}^* - 2b\boldsymbol{K} - \boldsymbol{\kappa}\boldsymbol{\kappa}^T), \tag{30}$$

where $\boldsymbol{K}^*$ is the higher-order confusion matrix in (11) for CPA attack on unmasked devices. Then the general success rate $\Phi_{N_k-1}(\sqrt{n}\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu})$ in Equation (3) can be calculated with this full variance formula (30). Numerical results in next section show that this complete SR model is very accurate in moderate to high noise situations.

3.4 Extension to Higher-order Masking Devices and Other Power Leakage Models

We now consider the J-th order masking scheme with J shares of masking variables, M_1, M_2, ..., M_J. Each M_j takes value uniformly in the set $\mathcal{M}$. The previous results can be extended to this general J-th order masking setting. The $(J+1)$-th order attack combines the leakage of $V_0 = V_0(Z \underset{j=1}{\overset{J}{\oplus}} M_j)$ at time t_0 and the leakage of $V_1 = V_1(M_1)$, ..., $V_J = V_J(M_J)$ at other J times points t_1,...,t_J, respectively. Denote $\boldsymbol{M} = (M_1, ..., M_J)$. The leakage vector is $\boldsymbol{l}_i = (l_{i,0}, l_{i,1}, ..., l_{i,J})$. The Gaussian leakage model is now:

$$L_j = L(t_j) = c_j + \varepsilon_j V_j + \sigma_j r_j, \qquad j = 0, ..., J. \tag{31}$$

To discover the first key-dependent term in the Taylor series, the number of $(J+2)$ leading terms will be kept and the $(J+1)$-th order ML attack can be shown again to be equivalent to the centered product combination attack. Furthermore, we can get the general formula for the success rate. The mean $\boldsymbol{\mu}$ and the simplified variance $\boldsymbol{\Sigma}$ of $\Delta(k_c, k_g)$ in (26) has elements

$$\mu_i = (\frac{1}{2})^{2J+1}(\prod_{j=0}^{J} \delta_j)^2 \kappa(k_c, k_{g_i}); \qquad \sigma_{ij} = (\frac{1}{2})^{2J}(\prod_{j=0}^{J} \delta_j)^2 \kappa(k_c, k_{g_i}, k_{g_j}). \tag{32}$$

Therefore the asymptotic success rate for $(J+1)$-th order attack becomes:

$$SR = \Phi_{N_k - 1}(\frac{\sqrt{n} \prod_{j=0}^{J} \delta_j}{2^{J+1}} \boldsymbol{K}^{-1/2} \boldsymbol{\kappa}). \tag{33}$$

The detailed analysis for the $(J+1)$-th order attack is provided in Appendix A. Formula (33) shows that each time one more mask is applied, the entire system SNR (the factor inside function Φ_{N_k-1}) changes by $\frac{\delta}{2}$ (normally lower than 1) and therefore the attack success rate reduces. Comparing a J-th order masked device to an unmasked device, we assume the first-order attack on the unmasked device requires n measurement traces to achieve a certain success rate, then $(J+1)$-th order attack on the J-th order masked device needs measurements in the order of $n(\frac{2}{\delta})^{2J}$ assuming all the δ_i are the same as δ. It is clear from this expression that higher-order masking is more effective when the noise is high (small δ).

We derived the results above under the Gaussian noise assumption (15) and Hamming Weight leakage model (24). Some extensions to other leakage models are possible. For non-Hamming Weight leakage, the CPCF attack (17) maximize the correlation with a function $f(Z_i^g)$ that may be different from $H(Z_i^g)$. However, the $f(Z_i^g)$ is still a deterministic function whose explicit formula can be calculated from the given leakage model. For example, recent work [26] does so for linear regression leakage model. For Non-Gaussian noise, the success rate formulas (29) and (33) still holds for CPCF attack. Some more discussion on such extensions will be provided in an extended version on eprint.iacr.org. Full extensions to other power leakage model and other masking schemes remains an open topic.

4 Numerical Results

In this section, we verify the derived statistical model for second-order DPA attacks on realistic measurement data, and also run numerical simulations on synthetic data for second-order and higher-order attacks.

4.1 Empirical Success Rates on Measurements from a Physical Implementation

We first verify the analytical results of Section 3 on real measurement data of a masked AES implementation on an SASEBO-GII board with a Virtex-5 FPGA. The SASEBO board implements the boolean-masked AES algorithm according to the scheme described in [29]. A 128-bit random mask sequence is obtained from a set of linear shift registers [30], then XORed with the input plaintext before the AES

AddRoundkey operation. The AES SBox module implementations are modified to keep all intermediate states masked. The overhead of such masking is large, with 50% more slices and 67% more power consumption than the unprotected AES implementation on the same FPGA board.

We collect $N = 1,400,000$ power traces with 3125 points for each one. The two leakage points are at the time points with the highest correlation between the power measurements and $H(M)$ and $H(Z \oplus M)$, respectively. The first leakage point leaks the Hamming Weight of the random mask M, while the second leakage point leaks the Hamming distance of the first byte of SBOX output in the last round of AES. We find them at the 581^{th} and 2873^{th} points, with their SNRs 0.0926 and 0.0955. To obtain the empirical success rate we repeatedly sample n traces from the total number of $N = 1,400,000$ traces. We conduct the second-order DPA attack on the sampled n traces with $1,000$ trials and calculate the empirical success rate for each selected n. We plot the empirical success rate versus number of traces in Fig. 1. To draw the theoretical success rate curve, we just use 10,000 traces to find the SNRs at the two points, and then plug them into formula (30) once, without complex experimental trials over millions of traces. Fig. 1 shows the two curves, *Empirical SR* and *Theoretical SR*, track each other very well, verifying that our theoretical success rate formula predicts the empirical success rates accurately. The analytical formula depicts the relation between the attack success rate and the number of traces, without collecting millions of traces and running statistical analysis to empirically calculate the SR. Such formula will be very useful for efficient countermeasure design evaluation before real implementation.

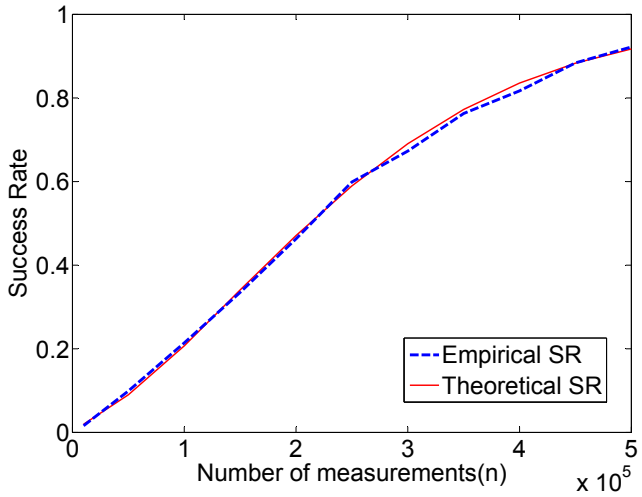

Fig. 1. Theoretical and empirical success rates of the second-order attack on a masked AES implementation.

We also check the noise distribution in the measured power traces, and find that for the Virtex-5 FPGA chip on SASEBO-GII board under 65 nm technology, the power model is indeed linear. Fig. 2 shows the average power and distribution for each group of power traces (with different Hamming distances) at the two time points.

4.2 Success Rates on Synthetic Datasets

We further verify the analytic success rate formula on synthetic datasets generated from the Hamming weight model (15), to evaluate the effect of system parameters, SNRs, on side-channel attacks and validate our approximate ML-attacks with the centered-product attacks. For simplicity, we take $c_0 = c_1 = 1$ and signal strengths $\varepsilon_0 = \varepsilon_1 = 1$ in the simulation. The range of noises, standard deviations σ_0 and σ_1, is $\{1, 5\}$. That is, the SNRs δ_0 and δ_1 take values in the range of $\{0.2, 1\}$. These settings are similar to those in previous work [16].

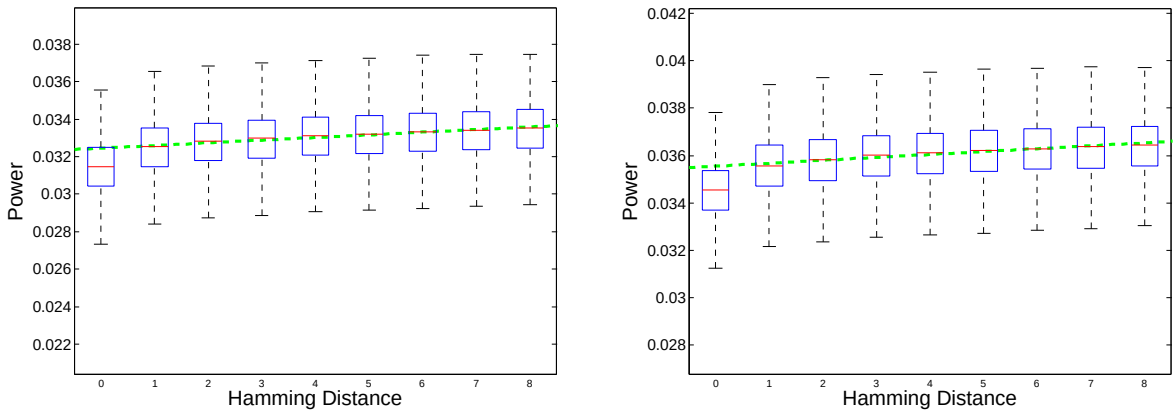

Fig. 2. The linear power model with Gaussian distribution noises at the two time points

For each set of generated power leakages, we apply two attacks as discussed in Remark 2: (a) the theoretical strongest approximate ML-attack that assumes all the parameters $(\boldsymbol{c}, \boldsymbol{\varepsilon}, \boldsymbol{\sigma})$ known (as a system designer); (b) the real second-order attack with CPCF that only works on the power data (as an attacker). We plot the success rate versus number of measurements for different SNRs δ_0 and δ_1 values in Fig. 3, for the two attacks and the theoretical model. $10,000$ simulation trials were run to compute the empirical success rates of the attacks. We can see that the theoretical success rate curve fits the empirical results well when SNRs are small. In addition, attacks (a) and (b) match very well, showing the equivalence between our approximate second-order ML attack and the second-order attack based on the CPCF. In each graph of Fig. 3, when one SNR increases, the attack requires less measurements for the same success rate. When the SNRs are big, $\delta_0 = \delta_1 = 1$, the three curves diverge for small n but still converge for large n values. This confirms that our asymptotic analysis works for big sample size n under very noisy situations (small SNRs). In reality, SNRs are small and would not be as high as 1.

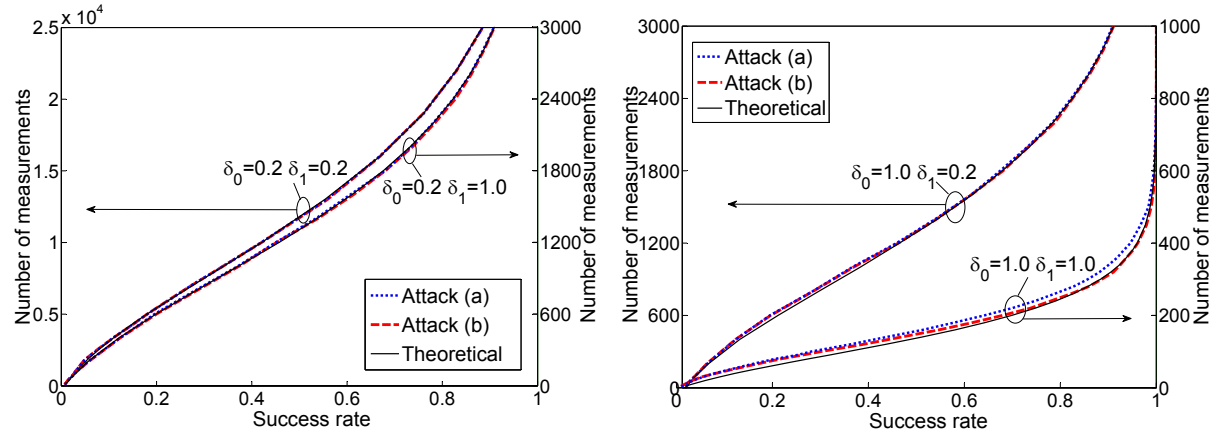

Fig. 3. The empirical and theoretical success rates of second order attacks (a) known parameter ML-attack and (b) the CPCF attack on masked AES SBox.

Finding the success rate of an attack based on simulated power data can be as time-consuming as on real measurements, especially when the number of traces is large. For example, the $10,000$ simulations

to create the success rate curve for $\delta_0 = \delta_1 = 0.2$ in Fig. 3 took about 11 hours on our workstation while the success rate curve using the explicit formula is produced within seconds. Hence, the analytic success rate formula would be very efficient and insightful for a secure system designer to evaluate any implementation.

The theoretical success rate model also helps us better understand the effect of masking on the security against SCAs. With masking, the number of measurements for the masked device should increase to be $(\frac{2}{\delta})^2$ times the number of measurements for the unmasked device to achieve the same success rate. For example, in noisy situations when $\delta = 0.1$, that is 400 times. We see that while the second-order attack can break first-order masking, it is computationally much harder to conduct due to significantly reduced information leakage. Moreover, the leakage reduction is much more pronounced in noisy situations with small SNRs. Hence the security benefit of masking is greater when it is combined with other countermeasures that aim to increase the noise and reduce the SNR.

4.3 Extension to Higher-order Attacks and Other Power Models

The general formula for higher-order attacks is given in Section 3.4. As mentioned at the end of Section 3.4, the success rate formula also hold for other non-Gaussian noises. Here we numerically study these extended SR formula. Firstly, we generate power data from higher-order mask model (31) with $J = 2$ and SNRs $\delta_0 = \delta_1 = \delta_2 = 0.2$. Fig. 4 shows the success rates of the corresponding third-order approximate ML attack again fits the theoretical success rate formula very well. Compared to the second-order results in Fig. 3, the number of measurements needed for the third-order attack increases to 100 times of that needed for the second-order attack to achieve the same success rate under the same SNRs ($\delta = 0.2$). Secondly, we generate synthetic power data from model (15) with Laplace noises instead. That is, the noises r_0 and r_1 both come from the probability density function $p(x) = e^{-\sqrt{2}|x|}/\sqrt{2}$. We set the SNRs $\delta_0 = 0.0955$ and $\delta_1 = 0.0926$ the same as the SNRs observed in our real measurements. The success rate curves for the second-order attack are shown in Fig. 5. We can see that the theoretical success rate formula fits the empirical success rates equally well for Laplace noises. The plot is very similar to the plot of success curves under Gaussian noises with the same SNRs.

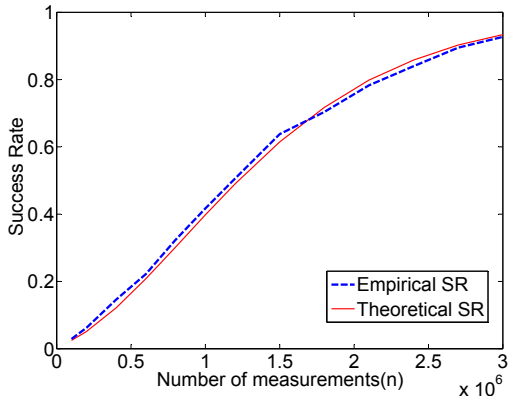

Fig. 4. The success rates of the third-order attack on simulated data with all three SNRs $\delta_0 = \delta_1 = \delta_2 = 0.2$.

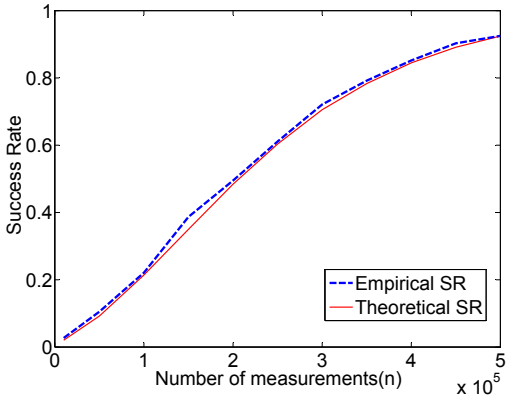

Fig. 5. The success rates of the second-order attack on simulated data with noises from the Laplace distribution ($\delta_0 = 0.0955, \delta_1 = 0.0926$).

5 Discussions and conclusions

Various other combination functions have been proposed in literature. Joye et al. suggested raising the absolute difference combining to a power α in [11]. In [15], Oswald et al. proposed a combination function

based on the sine function. There has not been any theoretical result indicating the optimal combination function. Prouff et al. proved in [16] that the CPCF is optimal among all attacks using the product combination functions in noisy situations. We prove for the first time that the most powerful SCA, ML-attack, is equivalent to the CPCF attack under very noisy situations. This gives a formal proof that the centered product combination function based attack is indeed optimal among all possible second-order and higher-order attacks on masked devices.

References

1. E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," *Proc. Int. Wksp on Cryptographic Hardware & Embedded Systems*, pp. 135–152, 2004.
2. B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis," in *Proc. Int. Wksp on Cryptographic Hardware & Embedded Systems*, 2008, pp. 426–442.
3. S. Chari, J. Rao, and P. Rohatgi, "Template attacks," in *Proc. Int. Wksp on Cryptographic Hardware & Embedded Systems*, 2003, pp. 51–62.
4. B. Gierlichs, K. Lemke-Rust, and C. Paar, "Templates vs. stochastic methods: A performance analysis for side channel cryptanalysis," in *Proc. Int. Wksp on Cryptographic Hardware & Embedded Systems*, 2006.
5. P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Int. Cryptology Conf. on Advances in Cryptology*, 1999, pp. 388–397.
6. S. Chari, C. Jutla, J. Rao, and P. Rohatgi, "A cautionary note regarding evaluation of AES candidates on smart-cards," in *Second Advanced Encryption Standard Candidate Conf.*, 1999, pp. 22–23.
7. J. Blömer, J. Guajardo, and V. Krummel, "Provably secure masking of AES," in *Selected Areas in Cryptography*, 2005, pp. 69–83.
8. E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen, "A side-channel analysis resistant description of the AES S-box," in *Fast Software Encryption*, 2005, pp. 413–423.
9. D. Canright and L. Batina, "A very compact perfectly masked S-box for AES," in *Applied Cryptography & Network Security*, 2008, pp. 446–459.
10. T. Messerges, "Using second-order power analysis to attack DPA resistant software," in *Proc. Int. Wksp on Cryptographic Hardware & Embedded Systems*, 2000, pp. 27–78.
11. M. Joye, P. Paillier, and B. Schoenmakers, "On second-order differential power analysis," *Proc. Int. Wksp on Cryptographic Hardware & Embedded Systems*, pp. 293–308, 2005.
12. S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counter power analysis attacks," in *Proc. Crypto*, Aug. 1999, pp. 398–412.
13. K. Schramm and C. Paar, "Higher order masking of the AES," *Topics in Cryptology–CT-RSA 2006*, pp. 208–225, 2006.
14. B. Gierlichs, L. Batina, B. Preneel, and I. Verbauwhede, "Revisiting higher-order DPA attacks," in *Topics in Cryptology-CT-RSA 2010*, 2010, pp. 221–234.
15. E. Oswald, S. Mangard, C. Herbst, and S. Tillich, "Practical second-order DPA attacks for masked smart card implementations of block ciphers," in *RSA Conference Cryptographers Track*, 2006, pp. 192–207.
16. E. Prouff, M. Rivain, and R. Bevan, "Statistical analysis of second order differential power analysis." *IEEE Trans.on Computers*, pp. 799 – 811, 2009.
17. F.-X. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, and S. Manyard, "The world is not enough: Another look on second-order DPA," in *Advances in cryptology - AsiaCrypt 2010*, 2010, pp. 112 – 129.
18. E. Prouff and M. Rivain, "Masking against side-channel attacks: A formal security proof," in *Advances in Cryptology–EUROCRYPT 2013*. Springer, 2013, pp. 142–159.
19. Y. Fei, Q. Luo, and A. A. Ding, "A statistical model for DPA with novel algorithmic confusion analysis," in *Proc. Int. Wksp on Cryptographic Hardware & Embedded Systems*, Sept. 2012.
20. Q. Luo and Y. Fei, "Algorithmic collision analysis for evaluating cryptographic systems and side-channel attacks," in *IEEE Int. Symp. Hardware Oriented Security & Trust*, June 2011, pp. 75–80.
21. F.-X. Standaert, T. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Advances in Cryptology - EUROCRYPT*, 2009, pp. 443–461.
22. H. Fischer, *A history of the Central Limit Theorem: From classical to modern probability theory.* Springer, 2011.
23. M. Rivain, "On the exact success rate of side channel analysis in the gaussian model," in *Selected Areas in Cryptography*, ser. Lecture Notes in Computer Science, R. Avanzi, L. Keliher, and F. Sica, Eds. Springer Berlin Heidelberg, 2009, vol. 5381, pp. 165–183.

24. Y. Fei, A. A. Ding, J. Lao, and L. Zhang, "A statistics-based fundamental model for side-channel attack analysis," Cryptology ePrint Archive, Report 2014/152, 2014, http://eprint.iacr.org/.
25. A. Thillard, E. Prouff, and T. Roche, "Success through confidence: Evaluating the effectiveness of a side-channel attack," in *Proc. Int. Wksp on Cryptographic Hardware & Embedded Systems*, 2013, pp. 21–36.
26. G. Dabosville, J. Doget, and E. Prouff, "A new second-order side channel attack based on linear regression," *Computers, IEEE Transactions on*, vol. 62, no. 8, pp. 1629–1640, 2013.
27. A. Runnalls, "Kullback-leibler approach to gaussian mixture reduction," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 43, no. 3, pp. 989–999, July 2007.
28. E. Seneta, "A tricentenary history of the law of large numbers," *Bernoulli*, vol. 19, no. 4, pp. 1088–1121, 09 2013.
29. M.-L. Akkar and C. Giraud, "An implementation of DES and AES, secure against some attacks," in *Proc. Int. Wksp on Cryptographic Hardware & Embedded Systems*, 2001, pp. 309–318.
30. T. G. Lewis and W. H. Payne, "Generalized feedback shift register pseudorandom number algorithm," *Journal of the ACM (JACM)*, vol. 20, no. 3, pp. 456–468, 1973.

Appendix

A Derivations for the $(J+1)$-th order ML-attack on Masked Devices

For the $(J+1)$-th order ML-attack, the attack statistic from Equation (16) becomes

$$T^g_{ML} = \frac{1}{n}\sum_{i=1}^{n} \log\{E_{\boldsymbol{m}}[\phi(r^g_{\boldsymbol{m},i,0}) \prod_{j=1}^{J} \phi(r_{\boldsymbol{m},i,j})]\} \tag{34}$$

where $\boldsymbol{m} = (m_{(1)}, ..., m_{(J)})$, $r^g_{\boldsymbol{m},i,0} = \frac{l_{i,0} - c_0 - \varepsilon_0 V^g_{\boldsymbol{m},i,0}}{\sigma_0} = r_{i,0} + \delta_0(V_{i,0} - V^g_{\boldsymbol{m},i,0})$ and $r_{\boldsymbol{m},i,j} = \frac{l_{i,j} - c_j - \varepsilon_j V_{\boldsymbol{m},j}}{\sigma_j} = r_{i,j} + \delta_j(V_{i,j} - V_{\boldsymbol{m},j})$, $j = 0, ..., J$. Here $V^g_{\boldsymbol{m},i,0} = V_0(Z^g_i \oplus^J_{j=1} m_{(j)})$, $V_{\boldsymbol{m},j} = V_j(m_{(j)})$, and $\delta_j = \varepsilon_j/\sigma_j$ denotes SNR for $j = 0, ..., J$. The $E_{\boldsymbol{m}}$ takes expectation over all possible values of the vector $\boldsymbol{m} = (m_{(1)}, ..., m_{(J)})$. That is, each element $m_{(j)}$ iterate over $\mathcal{M}$.

For the $(J+1)$-th order attack, we need to take $J+2$ terms in the Taylor expansion of (34). So the leading key-sensitive term (22) before becomes

$$T^g = \frac{1}{n}\sum_{i=1}^{n} B^g_i = \frac{1}{n}\sum_{i=1}^{n} (-\frac{1}{2})^{J+1} E_{\boldsymbol{m}}[A^g_{\boldsymbol{m},i} \prod_{j=1}^{J} A_{\boldsymbol{m},i,j}], \tag{35}$$

where $A^g_{\boldsymbol{m},i}$ and $A_{\boldsymbol{m},i,j}$ in (23) now become

$$\begin{array}{l} A^g_{\boldsymbol{m},i} \;= A_{i,0} - 2\delta_0 \tilde{l}_{i,0} V^g_{\boldsymbol{m},i,0} + \delta_0^2 V^g_{\boldsymbol{m},i,0}[V^g_{\boldsymbol{m},i,0} - E(V_{i,0})]; \\ A_{\boldsymbol{m},i,j} = A_{i,j} - 2\delta_j \tilde{l}_{i,j} V_{\boldsymbol{m},j} + \delta_j^2 V_{\boldsymbol{m},j}[V_{\boldsymbol{m},j} - E(V_{i,j})] \qquad j = 1, ..., J, \end{array} \tag{36}$$

Using (36), we find the leading key-sensitive term in $E_{\boldsymbol{m}}[A^g_{\boldsymbol{m},i} \prod^J_{j=1} A_{\boldsymbol{m},i,j}]$ relates to the centered product combination as

$$E_{\boldsymbol{m}}(\{-2\delta_0 \tilde{l}_{i,0} V^g_{\boldsymbol{m},i,0}\} \prod^J_{j=1} \{-2\delta_j \tilde{l}_{i,j} V_{\boldsymbol{m},j}\}) = (-2)^{J+1} (\prod^J_{j=0} \delta_j) \widetilde{C}_i f(Z^g_i).$$

Here $\widetilde{C}_i = \prod^J_{j=0} \tilde{l}_{i,j}$ and $f(Z^g_i) = E_{\boldsymbol{m}}(V^g_{\boldsymbol{m},i,0} \prod^J_{j=1} V_{\boldsymbol{m},j})$. Plug this into (35),

$$T^g = B^* + \widetilde{T}^g + o(\prod_{j=0}^{J} \delta_j), \qquad \text{with } \widetilde{T}^g = \frac{(\prod^J_{j=0} \delta_j)}{n} \sum_{i=1}^{n} \widetilde{C}_i f(Z^g_i). \tag{37}$$

The $f(Z^g_i)$ has an explicit formula as $E_{\boldsymbol{m}}(V^g_{\boldsymbol{m},i,0} \prod^J_{j=1} V_{\boldsymbol{m},j})$ under the Hamming Weight power leakage model:

$$V_0(Z, \boldsymbol{M}) = H(Z \overset{J}{\underset{j=1}{\oplus}} M_j) \qquad \text{and} \qquad V_j(\boldsymbol{M}) = H(M_j), \qquad j = 1, ..., J. \tag{38}$$

By Lemma 21 in [16], for any b-bits random mask M, $E[H(Z \oplus M)H(M)|Z] = -\frac{1}{2}H(Z) + b/4$. Apply this formula once, we get

$$E_{m_{(1)}}(V^g_{\boldsymbol{m},i,0}V_{\boldsymbol{m},1}) = E_{m_{(1)}}[H(Z_i^g \overset{J}{\underset{j=1}{\oplus}} m_{(j)})H(m_{(1)})] = -\frac{1}{2}H(Z_i^g \overset{J}{\underset{j=2}{\oplus}} m_{(j)}) + b/4.$$

Repeatedly apply the formula another $J-1$ times, we get that

$$E_{\boldsymbol{m}}(V^g_{\boldsymbol{m},i,0}\prod_{j=1}^{J} V_{\boldsymbol{m},j}) = (-\frac{1}{2})^J H(Z_i^g) + constant.$$

Hence under the Hamming Weight power leakage model, the centered product combination function attack maximizes

$$\widetilde{T}^g = (-\frac{1}{2})^J \frac{(\prod_{j=0}^{J}\delta_j)}{n}\sum_{i=1}^{n}\widetilde{C}_i H(Z_i^g). \tag{39}$$

We now can calculate $\boldsymbol{u}$ and $\boldsymbol{\Sigma}$ in the success Rate (3) for the higher order attack. From (35), $\Delta(k_c, k_g) = T^c - T^g = \frac{1}{n}\sum_{i=1}^{n}(B_i^c - B_i^g)$. So as in Section 2, $\boldsymbol{u}$ and $\boldsymbol{\Sigma}$ are the mean and variance of $\tilde{\Delta}_1 = (B_1^c - B_1^{g_1}, \ldots, B_1^c - B_1^{g_{N_k-1}})^T$. Using (37) and (39), we get

$$B_1^c - B_1^g = (-\frac{1}{2})^J(\prod_{j=0}^{J}\delta_j)[\prod_{j=0}^{J}\widetilde{l}_{1,j}][H(Z_1^c) - H(Z_1^g)]. \tag{40}$$

Recall $\widetilde{l}_{1,j} = r_{1,j} + \delta_j[V_{1,j} - E(V_{1,j})]$. Since $E(r_{1,j}) = 0$ and $r_{1,j}$'s are independent of the $V_{i,j}$'s. We find

$$E(B_1^c - B_1^g) = (-\frac{1}{2})^J(\prod_{j=0}^{J}\delta_j)^2 E\{\prod_{j=0}^{J}[V_{1,j} - E(V_{1,j})][H(Z_1^c) - H(Z_1^g)]\}.$$

Recall that $V_{1,j} = H(m_{1,j})$'s are Hamming weights of the random masks for $j = 1, ..., J$. Using formula (46) in Appendix C, taking expectation over the masks, $E\{\prod_{j=0}^{J}[V_{1,j} - E(V_{1,j})][H(Z_1^c) - H(Z_1^g)]\} = E\{(-1/2)^J[H(Z_1^c) - b/2][H(Z_1^c) - H(Z_1^g)]\}$. Here b is the bit length of Z_1^c. Since $E[H(Z_1^c) - H(Z_1^g)] = b/2 - b/2 = 0$, we get

$$\begin{aligned}
E(B_1^c - B_1^g) &= (-\tfrac{1}{2})^{2J}(\textstyle\prod_{j=0}^{J}\delta_j)^2 E\{H(Z_1^c)[H(Z_1^c) - H(Z_1^g)]\} \\
&= (\tfrac{1}{2})^{2J}(\textstyle\prod_{j=0}^{J}\delta_j)^2 E\{[H(Z_1^c)]^2 - H(Z_1^c)H(Z_1^g)\} \\
&= (\tfrac{1}{2})^{2J}(\textstyle\prod_{j=0}^{J}\delta_j)^2 \tfrac{1}{2} E\{[H(Z_1^c)]^2 + [H(Z_1^g)]^2 - 2H(Z_1^c)H(Z_1^g)\} \\
&= (\tfrac{1}{2})^{2J+1}(\textstyle\prod_{j=0}^{J}\delta_j)^2 E\{[H(Z_1^g) - H(Z_1^c)]^2\} \\
&= (\tfrac{1}{2})^{2J+1}(\textstyle\prod_{j=0}^{J}\delta_j)^2 \kappa(k_c, k_g).
\end{aligned}$$

Here the confusion coefficient $\kappa(k_c, k_g)$ is exactly the same as the confusion coefficient for unmasked device defined in (9). Thus we arrive at the first formula in equation (32)

$$\mu_i = E[B_1^c - B_1^{g_i}] = (\frac{1}{2})^{2J+1}(\prod_{j=0}^{J}\delta_j)^2\kappa(k_c, k_{g_i}).$$

The ij-th element in the variance $\boldsymbol{\Sigma} = Var(\tilde{\Delta}_1)$ is

$$\sigma_{ij} = Cov(B_1^c - B_1^{g_i}, B_1^c - B_1^{g_j}) = E[(B_1^c - B_1^{g_i})(B_1^c - B_1^{g_j})] - \mu_i\mu_j.$$

Since $E(r_{1,j}) = 0$ and $E(r_{1,j}^2) = 1$ for $j = 0, ..., J$, using equation (40), we have the leading term in σ_{ij} as in the second formula in equation (32)

$$(-\frac{1}{2})^{2J}(\prod_{j=0}^{J}\delta_j)^2 E(\prod_{j=0}^{J}r_{1,j}^2)E\{[H(Z_1^{g_i}) - H(Z_1^c)][H(Z_1^{g_j}) - H(Z_1^c)]\} = (\frac{1}{2})^{2J}(\prod_{j=0}^{J}\delta_j)^2\kappa(k_c, k_{g_i}, k_{g_j}).$$

Here the three-way confusion coefficient $\kappa(k_c, k_{g_i}, k_{g_j})$ is exactly the same as those defined for unmasked device in (10).

Taking $J = 1$, (32) becomes (27) and (28).

B More Accurate Formula for the Covariance Matrix $\boldsymbol{\Sigma}$ in the Second-order ML Attack on Masked Devices

Equation (28) only calculate the leading term in σ_{ij}. We can calculate all the terms to get more accurate formula. Keeping all terms in $\tilde{l}_{1,0}$ and $\tilde{l}_{1,1}$ we have

$$B_1^c - B_1^g = -\frac{1}{2}\delta_0\delta_1[r_{1,0} + \delta_0(V_{1,0} - \frac{b}{2})][r_{1,1} + \delta_1(V_{1,1} - \frac{b}{2})][H(Z_1^c) - H(Z_1^g)].$$

Then, since $E(r_{1,0}) = E(r_{1,1}) = 0$ and $E(r_{1,0}^2) = E(r_{1,1}^2) = 1$, we have

$$\begin{array}{l} \sigma_{ij} = \frac{1}{4}\delta_0^2\delta_1^2 E\{[H(Z_1^{g_i}) - H(Z_1^c)][H(Z_1^{g_j}) - H(Z_1^c)]\} \\ \quad +\frac{1}{4}\delta_0^2\delta_1^4 E\{(V_{1,1} - \frac{b}{2})^2[H(Z_1^{g_i}) - H(Z_1^c)][H(Z_1^{g_j}) - H(Z_1^c)]\} \\ \quad +\frac{1}{4}\delta_0^4\delta_1^2 E\{(V_{1,0} - \frac{b}{2})^2[H(Z_1^{g_i}) - H(Z_1^c)][H(Z_1^{g_j}) - H(Z_1^c)]\} \\ \quad +\frac{1}{4}\delta_0^4\delta_1^4 E\{(V_{1,0} - \frac{b}{2})^2(V_{1,1} - \frac{b}{2})^2[H(Z_1^{g_i}) - H(Z_1^c)][H(Z_1^{g_j}) - H(Z_1^c)]\} - \mu_i\mu_j. \end{array} \tag{41}$$

Using (44) and (47) in Appendix C, this simplifies to

$$\begin{array}{rl} \sigma_{ij} = & \frac{1}{4}\delta_0^2\delta_1^2\kappa(k_c, k_{g_i}, k_{g_j}) + \frac{1}{4}\delta_0^2\delta_1^4\frac{b}{4}\kappa(k_c, k_{g_i}, k_{g_j}) + \frac{1}{4}\delta_0^4\delta_1^2\frac{b}{4}\kappa(k_c, k_{g_i}, k_{g_j}) \\ & +\frac{1}{4}\delta_0^4\delta_1^4[\frac{1}{2}\kappa^*(k_c, k_{g_i}, k_{g_j}) + \frac{b^2-2b}{16}\kappa(k_c, k_{g_i}, k_{g_j})] - \frac{1}{64}\delta_0^4\delta_1^4\kappa(k_c, k_{g_i})\kappa(k_c, k_{g_j}) \\ = & \frac{1}{4}\delta_0^2\delta_1^2(1 + \frac{b}{4}\delta_0^2)(1 + \frac{b}{4}\delta_1^2)\kappa(k_c, k_{g_i}, k_{g_j}) \\ & +\frac{1}{64}\delta_0^4\delta_1^4[8\kappa^*(k_c, k_{g_i}, k_{g_j}) - 2b\kappa(k_c, k_{g_i}, k_{g_j}) - \kappa(k_c, k_{g_i})\kappa(k_c, k_{g_j})]. \end{array} \tag{42}$$

Here the three-way confusion coefficients $\kappa(k_c, k_{g_i}, k_{g_j})$ and $\kappa^*(k_c, k_{g_i}, k_{g_j})$ are the same as those defined for unmasked device in (10) and (11).

Thus we get the formula (30).

C Formulas for Eliminating $\boldsymbol{E_m}$ with Hamming Weight Power Models on Masked Devices

Here we list some formulas used for calculation in the above derivations. We shall consider the b-bit mask M similarly as in Prouff et al [16]. We are going to consider quantities involving $H(Z \oplus M)$ and $H(M)$ for a fixed Z. By Lemma 20 in [16],

$$E[H(M)] = \frac{b}{2}; \qquad E[H(M)^2] = \frac{b^2 + b}{4}. \tag{43}$$

Therefore,

$$E[(H(M) - \frac{b}{2})^2] = E[H(M)^2] - bE[H(M)] + \frac{b^2}{4} = \frac{b}{4}. \tag{44}$$

By Lemma 21 in [16],

$$E[H(M)H(Z \oplus M)|Z] = -\frac{1}{2}H(Z) + \frac{b^2 + b}{4}. \tag{45}$$

Combine (43) and (45), we arrive at

$$E\{[H(M) - \frac{b}{2}][H(Z \oplus M) - \frac{b}{2}]|Z\} = -\frac{1}{2}H(Z) + \frac{b^2 + b}{4} - (\frac{b}{2})^2 = -\frac{1}{2}H(Z) + \frac{b}{4} = -\frac{1}{2}[H(Z) - \frac{b}{2}].$$

Hence for fixed Z and J random masks M_1, ..., M_J, we have

$$E\{[H(Z \underset{j=1}{\overset{J}{\oplus}} M_j) - \frac{b}{2}] \prod_{j=1}^{J} [H(M_j) - \frac{b}{2}]\} = -\frac{1}{2} E\{[H(Z \underset{j=1}{\overset{J-1}{\oplus}} M_j) - \frac{b}{2}] \prod_{j=1}^{J-1} [H(M_j) - \frac{b}{2}]\} = ... = (-\frac{1}{2})^J [H(Z) - \frac{b}{2}] \tag{46}$$

The last result required in earlier derivation is

$$E\{[H(Z \oplus M) - \frac{b}{2}]^2 [H(M) - \frac{b}{2}]^2 | Z\} = \frac{1}{2}[H(Z) - \frac{b}{2}]^2 + \frac{b^2 - 2b}{16}. \tag{47}$$

This equation (47) follows from (43), (45) above, and (48), (49) below.

$$E[H(M)H(Z \oplus M)^2 | Z] = E[H(M)^2 H(Z \oplus M) | Z] = -\frac{b}{2} H(Z) + \frac{b^2(b+3)}{8}; \tag{48}$$

$$E[H(M)^2 H(Z \oplus M)^2 | Z] = \frac{1}{2} H(Z)^2 - \frac{b^2 + b}{2} H(Z) + \frac{b(b^3 + 6b^2 + 3b - 2)}{16}. \tag{49}$$

Equations (48) and (49) come from straight calculation using formulas (43),(50),(57),(58), (59) and (60). The above calculation used the formulas below which are derived similar to those in in [16]. First, let $\wedge$ denote the bit-wise multiplication. Then the following formula is the property 2 in [16]:

$$H(Z \oplus M) = H(Z) + H(M) - 2H(Z \wedge M). \tag{50}$$

Let $Z_{(i)}$ denotes the ith bit of Z. We derive the following formulas, using the fact that $E(M_{(i)} M_{(j)}) = E(M_{(i)}) = \frac{1}{2}$ when $i = j$ and $E(M_{(i)} M_{(j)}) = \frac{1}{4}$ by independence when $i \neq j$.

$$E[H(Z \wedge M)] = \frac{1}{2} H(Z), \tag{51}$$

since $E[H(Z \wedge M)] = E[\sum_{i=1}^{b} Z_{(i)} M_{(i)}] = \sum_i Z_{(i)} \frac{1}{2}$.

$$E[H(M)H(Z \wedge M)] = \frac{b+1}{4} H(Z), \tag{52}$$

since $E[H(M)H(Z \wedge M)] = \sum_{i=1}^{b} Z_{(i)} \sum_{j=1}^{b} E[M_{(i)} M_{(j)}] = \sum_i Z_{(i)} [\frac{1}{2} + (b-1)\frac{1}{4}]$.

$$E[H(Z \wedge M)H(Z^g \wedge M)] = \frac{1}{4} H(Z)H(Z^g) + \frac{1}{4} H(Z \wedge Z^g), \tag{53}$$

since $E[H(Z \wedge M)H(Z^g \wedge M)] = \sum_{i,j} Z_{(i)} Z^g_{(j)} E[M_{(i)} M_{(j)}]$ which becomes

$$\sum_{i=j} Z_{(i)} Z^g_{(j)} \frac{1}{2} + \sum_{i \neq j} Z_{(i)} Z^g_{(j)} \frac{1}{4} = \sum_{i,j} Z_{(i)} Z^g_{(j)} \frac{1}{4} + \sum_i Z_{(i)} Z^g_{(i)} \frac{1}{4}.$$

We get the following two formulas similarly as (51), (52) and (53) above, with the detailed calculation omitted for space.

$$E[H(M)H(Z \wedge M)H(Z^g \wedge M)] = \frac{b+2}{8} H(Z)H(Z^g) + \frac{b}{8} H(Z \wedge Z^g), \tag{54}$$

$$E[(H(M))^2 H(Z \wedge M)H(Z^g \wedge M)] = \frac{b^2 + 5b + 2}{16} H(Z)H(Z^g) + \frac{b^2 + b - 2}{16} H(Z \wedge Z^g). \tag{55}$$

Taking $Z^g = Z$ in (54) and (55), we get

$$E[H(M)H(Z \wedge M)^2] = \frac{b+2}{8} H(Z)^2 + \frac{b}{8} H(Z), \tag{56}$$

$$E[(H(M))^2 H(Z \wedge M)^2] = \frac{b^2 + 5b + 2}{16} H(Z)^2 + \frac{b^2 + b - 2}{16} H(Z). \tag{57}$$

Taking Z^g to have every bit equals to one in (54) and (55), we get

$$E[H(M)^2 H(Z \wedge M)] = \frac{b(b+3)}{8} H(Z), \tag{58}$$

$$E[(H(M))^3 H(Z \wedge M)] = \frac{b^3 + 6b^2 + 3b - 2}{16} H(Z). \tag{59}$$

Taking Z to have every bit equals to one in (58) and (59), we get

$$E[H(M)]^3 = \frac{b^2(b+3)}{8}; \qquad E[H(M)]^4 = \frac{b(b^3 + 6b^2 + 3b - 2)}{16}. \tag{60}$$