An Operation-Based Metric for CPA Resistance*

J. Pan, J.I. den Hartog, and E.P. de Vink

Abstract Differential power analysis (DPA) is a strong attack upon cryptographic devices such as smartcards. Correlation power analysis (CPA) is a specific form of DPA where the Hamming-weight and the correlation coefficient are employed. In this paper we investigate the intrinsic vulnerability of the individual operations that are targeted in DPA attacks. We find that under the typical circumstances, there is a difference in resistance to the attack between the operations. We then provide a precise definition of CPA resistance and capture it in a simple yet effective metric to rank operations. The metric is validated with both simulations and experiments on actual hardware.

1 Introduction

Since the well-known work of Kocher *et al.* [7] and the research following [1, 2, 3, 8, 11, 12, 15], side-channel attacks and particularly Differential Power Analysis (DPA) have become a major security concern for the implementation of cryptographic algorithms on small devices such as smartcards. The side-channel exploited in DPA attacks is the power consumption of a cryptographic device that usually reveals some information about the data being processed. Unlike traditional cryptanalysis, a DPA attack targets a small part of the key at a time. This is possible because the power consumption of a cryptographic device at a point in time usually depends on only a few bits of the processed data.

Correlation power analysis (CPA) [3], as a specific form of DPA attack, employs the Hamming-weight model and the correlation coefficient. In this attack, the power consumption of the device is assumed to be linked to the Hamming-weight of the

Eindhoven University of Technology, Den Dolech 2, 5612 AZ, Eindhoven, the Netherlands. Corresponding author e-mail: j.pan@tue.nl.

* This work is funded by Sentinels-project PinpasJC TIF.6687.

Please use the following format when citing this chapter:

Pan, J., den Hartog, J.I. and de Vink, E.P., 2008, in IFIP International Federation for Information Processing, Volume 278; *Proceedings of the IFIP TC 11 23rd International Information Security Conference*; Sushil Jajodia, Pierangela Samarati, Stelvio Cimato; (Boston: Springer), pp. 429–443.

data. By looking at the correlation between the Hamming-weight of the predicted values and the actual power consumption, the hypothetical key values and the actual key of the device are compared.

In this paper, we study the resistance to CPA by examining the individual operations executed on the cryptographic device. The resistance of fundamental operations in the algorithm determines the basic resistance of the algorithm as a whole. Knowing which operation is the weakest, and how likely a CPA attack on this location is to succeed given a certain level of noise in the measurements, is an important starting point in defending the implementation of a cryptographic algorithm.

Our analysis of commonly used operations in smartcards indeed shows that there are differences in the success rate of CPA attacks (assuming there is some noise involved in the measurement of the power consumption, which is in practice always the case). We demonstrate this by simulated attacks and discuss the underlying statistics. The main goal of this paper is to provide an easy to use yet effective method for ranking operations based on their resistance to CPA attacks.

After formally defining the CPA resistance of operations, we introduce a metric which captures this resistance. The metric is simple to calculate as it is purely based on the correlation values for the different key candidates, using the operation and an estimated noise level of the target device. We show why it is reasonable to build our metric on these values and subsequently validate the metric. The validation is done for four operations with simulated attacks as well as experiments based on physical measurements in practice and then comparing the results with the ranking obtained from the metric. The validation shows that the metric can capture the CPA resistance of the operations on the device.

Prouff [13] investigated the problem of DPA vulnerability of S-boxes from a cryptographic point of view and defined the notion of a 'transparency order' of an S-box, meant to capture its resistance to a particular DPA attack [2]. The conclusion of [13], that some of the very properties which cryptographically enhance operations, weaken them on the other hand to power analysis, corresponds to our general observation (a point also made in [6]). Compared to the approach in [13], we define a metric that is simpler and more general in the type of operations that can be addressed. The notion of 'transparency order' is complex and requires rapidly increased computations regarding to the bit size. This is not really an issue for typical S-boxes, as cryptographic devices generally do not have the storage space to deal with large S-boxes, but it can become a practical problem for other types of operations. Individual operations were analyzed in [10] for DPA resistance as well, with both simulated and physical means, but not compared with respect to a metric. In the framework presented in [15], our approach fits in the class of strong implementation with an adequate leakage model and sufficient many queries.

The paper is organized as follows. Section 2 starts with a description of the simulated CPA attacks. Sections 3 and 4 demonstrate the simulation on some examples where noise is ignored and regarded, respectively. The essentials are then analyzed in Section 5, where a metric for ranking operations regarding to their resistance to CPA attacks is proposed and exampled. In Section 6, experimental results are shown validating the ranking given by our metric. The last section provides conclusion.

2 CPA Simulation

A comprehensive description of CPA simulation can be found in [11]. We here summarize the general technique with emphasis to the characteristics of our experiment.

2.1 Modeling of Power Consumption

For the modeling of the power consumption signals, we employ the decomposition pattern from [11] as shown below,

$$P_{total} = P_{data} + P_{op} + P_{noise} + P_{const} ,$$

where the total power consumption P_{total} at a single point in time can be decomposed into four disjoint components: the data-dependent consumption P_{data}, the operation-dependent consumption P_{op}, the electronic noise P_{noise} and the constant component P_{const}. The P_{data}, P_{op} and P_{noise} are the most important. The attacker can learn about confidential information by analyzing P_{data} and P_{op}. Electronic noise reflects the fluctuation that occurs when a fixed measurement is repeated. The bigger P_{noise} is, the more difficult the analysis is. The electronic noise in most cryptographic devices can be assumed to have a Gaussian distribution (see e.g. [9, 12]). In the presence of P_{const}, the expected value of P_{noise} equals zero. The standard deviation is, of course, specific to a device. We thus denote that $P_{noise} \sim \mathcal{N}(0, \sigma)$. The constant component P_{const} occurs independently of the operation performed and the data processed, and is therefore not relevant to CPA attacks.

2.2 The Attack Strategy

Let f denote the operation under attack and let $f(d,k)$ be the output of performing f on input d and key k. Input d can in general be calculated by the attacker based on the input to the device. The key material k is often a small portion of the secret key of the device.

In an CPA attack one can distinguish three phases – measurement, prediction, and analysis. In the first phase, the power consumption of a device is physically measured while the device performs cryptographic operations. In second phase, we predict the power consumption P_{data} for hypothetical key values, by constructing the output $f(d,k)$ for chosen d. In the analysis phase, the predicted power consumption values are compared with the measurements. The result determines which key guess used for power prediction can be a candidate for the key of the device. In our work, the measurements in the first phase is simulated based on the power model in Section 2.1. This implies we are supposed to know the key of the device in this phase,

so that given an operation and its input message, the output can be computed based on the key. We will next present the attack strategy in more details.

Phase 1: measurement. We compute the output $f(d,k)$ for different input d using the key k from the device. For this purpose, we generate an input vector $\mathbf{d} = (d_1, d_2, \ldots, d_m)'$ such that $\mathbf{d}$ includes all possible values for d. This allows us to obtain maximal information about the operation and, subsequently, about the key. The computation results in a vector of output values $\mathbf{v} = (v_1, v_2, \ldots, v_m)'$. They are then mapped to power consumption values $\mathbf{h} = (h_1, h_2, \ldots, h_m)'$ using the Hamming-weight power model, which projects a value X to the number of bit set in it, here referred to as $HW(X)$. To model the P_{noise} while each value in $\mathbf{v}$ is computed on the device, we use a vector of noise values $\mathbf{p} = (p_1, p_2, \ldots, p_m)'$ sampled from normal distribution $\mathcal{N}(0,\sigma)$. Since we perform the analysis for specific operation individually, P_{op} is constant for each measurement and is thus captured by P_{const}. As stated before, component P_{const} is not relevant in determining the correct key value. Hence, by omitting P_{op} and P_{const} from our simulation, the power consumption of the device at the point in time when an output value v_i is handled, is modeled as $t_i = HW(f(d_i,k)) + p_i$, yielding a vector of simulated power consumption values $\mathbf{t} = (t_1, t_2, \ldots, t_m)'$.

Phase 2: prediction. In this phase, we compute $f(d_i,k_j)$ with the input vector $\mathbf{d}$ from Phase 1 and a key hypotheses vector $\mathbf{k} = (k_1, k_2, \ldots, k_n)$ containing all possible choices for k. The simulation of P_{data} when $f(d_i,k_j)$ is processed is similar to that in Phase 1, however, it now needs to be done for each $k_j \in \mathbf{k}$. This leads to a matrix $\mathbf{H}$ of power consumption values, where $h_{i,j} = HW(f(d_i,k_j))$. Because P_{data} is the only relevant power component for determining the key in a CPA attack, the matrix $\mathbf{H}$ is then the result of this phase. Since $\mathbf{k}$ contains all possible choices for k, the key of the device is then among $\mathbf{k}$. We refer to the index of this element as ck and the key of the device as k_{ck}. Column $\mathbf{h}_{ck}$ of $\mathbf{H}$ is correspondingly derived based on k_{ck}.

Phase 3: analysis. After having obtained the simulated power consumption data and the predicted power consumption data, we next compare them and determine the correct key value. The comparison is based on the correlation coefficient, which is commonly used to express the linear relationship of two random variables, defined as:

$$CC(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X) \cdot Var(Y)}} \cdot$$

Based on N samples for X and Y each, the value of $Cov(X,Y)$, $Var(X)$ and $CC(X,Y)$ can typically be assessed by the following estimators, respectively:

$$W(\mathbf{x},\mathbf{y}) = \frac{1}{N-1} \cdot \sum_{i=1}^{N} (x_i - \bar{x}) \cdot (y_i - \bar{y})$$

$$S^2(\mathbf{x}) = \frac{1}{N-1} \cdot \sum_{i=1}^{N} (x_i - \bar{x})^2$$

$$R(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{N} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N} (x_i - \bar{x})^2 \cdot \sum_{i=1}^{N} (y_i - \bar{y})^2}}$$

The correlation between $\mathbf{t}$ and each column of $\mathbf{H}$ is estimated by R, resulting in a vector $\mathbf{r} = (r_1, r_2, \ldots, r_n)$, where r_j compares the j-th column of $\mathbf{H}$ to $\mathbf{t}$. Recall that column $\mathbf{h}_{ck}$ has been processed with the key hypothesis k_{ck}, which has also been used to simulate $\mathbf{t}$. Therefore, column $\mathbf{h}_{ck}$ and $\mathbf{t}$ are assumed to be strongly related and the corresponding correlate coefficient r_{ck} is the highest in $\mathbf{r}$. Other values of $\mathbf{r}$ are expected to be lower because the other columns of $\mathbf{H}$ and $\mathbf{t}$ are less correlated. Following this line of reasoning, the index of the correct key hypothesis ck is revealed.

A minor point suppressed in the sequel is the following. If the power consumption increases with the Hamming-weight, k_{ck} has a positive correlation coefficient; otherwise it has a negative correlation coefficient. The linear dependency is determined by specific cryptographic device, which, if is unknown beforehand to the attacker, a brute-force analysis needs to be applied. Therefore, we consider both positive and negative correlation peaks as possible candidates. Consequently, the absolute values of the correlation coefficients $(|r_1|, |r_2|, \ldots, |r_n|)$ are taken as references for the analysis, instead of the actual values. Some wrong key hypotheses cause what are often referred to as 'ghost peaks' in context of CPA attacks. The presence of ghost peaks typically requires additional brute-force methods to identify the correct key; and the cost increases exponentially on the number of ghost peaks. Therefore, we say that the more ghost peaks there are, the more resistant an operation is to CPA attacks.

2.3 Demonstration

In order to demonstrate the attack simulation, we take as examples four operations that are typically targeted in DPA attacks for software implementations of AES [4], TEA [16] and Edon [5]. The operations are: exclusive-or, modular addition, modular multiplication, and AES AddRoundKey plus SubBytes. In this paper, we refer to them as operations `XOR`, `ADD`, `MUL`, and `AES`, respectively. To achieve a fair comparison between the operations, they are all carried out with 8-bit data.

A note for `MUL` (see [5]) is that to avoid multiplications by zero, the inputs are mapped from $\mathbb{Z}_{255}$ to $\mathbb{Z}_{256}^*$ by a function g and the output is projected from $\mathbb{Z}_{256}^*$ back to $\mathbb{Z}_{255}$ by the inverted function g' after modular multiplication. The f-function is then: $f(d,k) = g'(g(d) \times g(k) \bmod 257)$.

3 Idealized Simulation

This section provides examples of CPA attacks for the idealized case, where the electronic noise component P_{noise} is omitted in the simulation and only the data-dependent component P_{data} is taken into account in the measurements. Clearly, this situation never occurs in practice. However, it is useful for better understanding the dependency between the processed data and the power consumption of the device. Technically, column $\mathbf{h}_{ck}$ of $\mathbf{H}$ and vector $\mathbf{t}$ now contain the same values, which results in the maximum correlation coefficient value 1 for r_{ck} for all operations.

We have performed this simulated attacks on every operation. The resulting correlation coefficients are plotted in Fig. 1. Note that for operations that are bijective, which is the case for our examples, the frequency distribution of the correlation coefficients is subject to the operation only, independently from the choice of the correct key. Based on the results in Fig. 1, we will next analyze the characteristics of the operations individually.

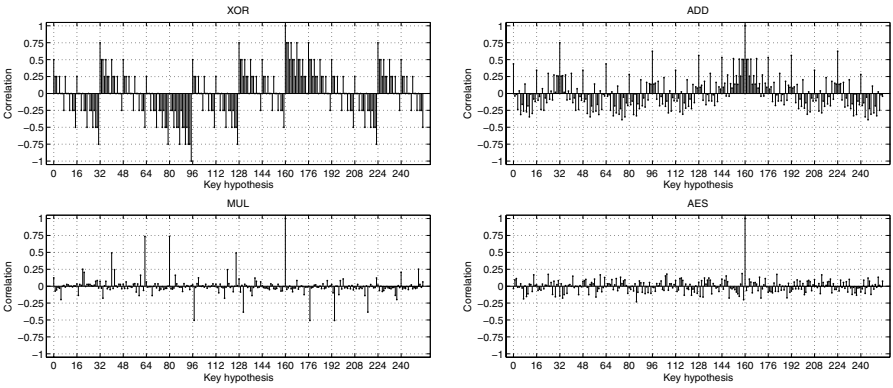

Fig. 1 Correlation coefficients for all key hypotheses when $k_{ck} = 160$.

XOR. The correlation coefficients for XOR are evaluated to 1 for the correct key hypothesis k_{ck} and to -1 for its bitwise inverted value $\neg k_{ck}$; hence, they are both considered as possible key candidates in this case. Ghost peaks occur at key hypotheses that differ by 1 bit from k_{ck} or $\neg k_{ck}$. The subsequent peaks correspond to key hypotheses that differ by 2 and 3 bits from k_{ck} or $\neg k_{ck}$. Those that differ by 4 bits from $\neg k_{ck}$ are not correlated and hence lead to zero correlation.

ADD. Operation ADD is similar to XOR except for the bit carry propagation. The wrong key hypotheses that cause ghost peaks can be ranked as: $k_{ck} \pm 2^7$, $k_{ck} \pm 2^6$, …, $k_{ck} \pm 2^0$, $k_{ck} \pm 2^7 \pm 2^6$, $k_{ck} \pm 2^7 \pm 2^5$, …, $k_{ck} \pm 2^7 \pm 2^0$, …. For instance, the output of $f(d, k_{ck} \pm 2^7)$ differs by one bit from $f(d, k_{ck})$ for any input d; and $f(d, k_{ck} \pm 2^6)$ differs from $f(d, k_{ck})$ for $2^8/2$ values by one bit, for $2^8/4$ values by zero bit and for $2^8/4$ values by two bits.

MUL. A few wrong key hypotheses show ghost peaks here. Employing the method in [10], we summarize the correlated key hypotheses in four sequences $K_{1,i}$, $K_{2,i}$, $K_{3,i}$ and $K_{4,i}$ as follows:

$$K_{1,i} = g'(2^i \cdot g(k_{ck}) \bmod 257)\ ;\ K_{2,i} = g'(257 - g(K_{1,i}))\ ;$$
$$K_{3,0} = k_{ck}\ ,\ K_{3,i+1} = g'(\tfrac{g(K_{3,i})}{2}) \text{ for } g(K_{3,i}) \text{ even}\ ,$$
$$K_{3,i+1} = g'(\tfrac{257-g(K_{3,i})}{2}) \text{ for } g(K_{3,i}) \text{ odd}\ ;\ K_{4,i} = g'(257 - g(K_{3,i}))\ ,$$

where $i = 0, 1, \ldots, 8$. To give an example, the key hypotheses that cause the peaks in Fig. 1 are: $k_{ck} = 160$; $K_{1,i} = \{\,160, 63, 126, 252, \ldots\,\}$; $K_{2,i} = \{\,97, 194, 131, 5, \ldots\,\}$; $K_{3,i} = \{\,160, 80, 40, 20, \ldots\,\}$; and $K_{4,i} = \{\,97, 177, 217, 237, \ldots\,\}$.

AES. In contrary to the other operations, no ghost peak occurs for `AES`. This is due to the fact that the AES S-box has been well chosen regarding to the non-linearity criterion. Although it is an advantage to resist linear cryptanalysis, this optimization allows CPA attacks to succeed easily.

4 Simulation with Noise

We now discuss more realistic DPA attacks where electronic noise is involved. A notion for the failure of CPA attacks is introduced, and experiments using the simulated CPA attacks based on this notion are presented.

Again, CPA selects possible key candidates according to the absolute values of the obtained correlation coefficients. A straightforward CPA chooses only the most significant correlation peak as the candidate. Due to the noise, the highest peak may not exactly occur at the correct key hypothesis and thus a wrong candidate could be returned. In this case, the attack is deemed to be failed. Intuitively, an attack can easily fail this way when the noise is high. As stated in Section 2.1, the influence of the electronic noise on the measurement is typically characterized by its standard deviation σ. The greater σ is, the higher P_{noise} is. Given the standard deviation σ for the noise P_{noise}, we refer to the resulting correlation coefficients based on P_{noise} as $(r_1^\sigma, r_2^\sigma, \ldots, r_n^\sigma)$. Accordingly, the correlation values obtained in the idealized case (see Section 3) are denoted as $(r_1^0, r_2^0, \ldots, r_n^0)$. Using these notations, we define the difference between the absolute correlation values for k_{ck} and k_j, for some σ, as follows:

$$\delta_j^\sigma = |r_{ck}^\sigma| - |r_j^\sigma|\ \cdot \tag{1}$$

We introduce notion $\mathscr{F}(\sigma)$ for the event that CPA fails when the standard deviation of noise equals σ. Notion $\mathscr{F}(\sigma)$ can then be formulated as a set of boolean outcomes that if there is any ghost peaks higher than the peak at k_{ck}:

$$\mathscr{F}(\sigma) = \{\,\delta_j^\sigma < 0 \mid 1 \le j \le n\,\}\ \cdot$$

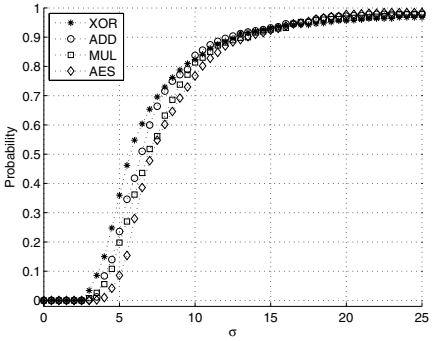

Fig. 2 *Prob*($\mathscr{F}(\sigma)$) for all operations.

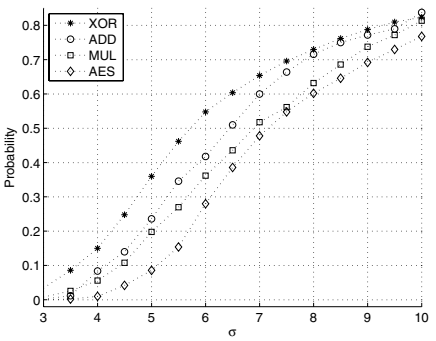

Fig. 3 A zoom of Fig. 2.

Next, we perform simulated CPA attacks according to the strategy presented in Section 2. In contrast to the idealized case, the noise values in vector $\mathbf{p}$ are now added to the power consumption values in $\mathbf{t}$. To model different levels of electronic noise, we generate a set of possible values $\{0.5, 1, 1.5, \ldots, 13\}$ for σ. In order to deliver a promising assessment for the probability that an attack fails, which is referred to as *Prob*($\mathscr{F}(\sigma)$), we have repeated each experiment for 500 times[2] with the same input d and k and different values for p. The number of times that the attacks fail is recorded for each choice of σ. Dividing these numbers individually by 500 yields the probabilities *Prob*($\mathscr{F}(\sigma)$) for each value of σ.

The above experiment has been performed to all the operations and the results are plotted in Fig. 2. The overall rising of the probabilities agrees that the attacks fails more often for all the cases as the noise is turned up. As the figure presents, after an initial leveling down at 0%, the failure rates start growing for all the cases. The increase is the most rapid for `XOR`. When $\sigma = 4$, for instance, the probability for `XOR` is close to twice as much as that for `ADD` and almost three times as much as that for `MUL`; and, it exceeds the one for `AES` by a ratio of 1.5 to 0. The ranking of the operations by their failure probabilities stays this way until σ hits 10. After that, they all converge to 1. As can be seen, in majority of the cases, the failure rates remain in the order: `XOR` > `ADD` > `MUL` > `AES`, which is, hence, taken as the ranking of the operations for the resistance to CPA attacks in our experiment.

5 A Metric for Resistance to CPA attacks

Above, we have shown the results of CPA simulation for both the noise-free case and the noise-involved case. In this section, we analyze the probability of CPA failure in Section 4 and show it primarily depends on the correlation values obtained in the idealized case in Section 3. We then integrate the results from both sections

[2] The number 500 is chosen as a trade-off of precision and cost of experiment.

and propose a metric for evaluating the resistance of an operation to CPA attacks, based on correlation values in the idealized simulation. Additionally, we show by examples how this metric can be used to rank operations.

5.1 Reasoning about $Prob(\delta_j^\sigma < 0)$

As previously assumed, a CPA attack fails when at least one ghost peak is higher than the resulting correlation by the correct key hypothesis. The difference δ_j^σ defined in (1) is modeled as in (2). Random variables H_j and P_{noise}^σ are used to denote respectively the hypothetical power consumption values in column j of $\mathbf{H}$ and the simulated noise values in $\mathbf{p}$ with a standard deviation of σ.

$$\begin{aligned}
\delta_j^\sigma &= |CC(H_{ck}, H_{ck} + P_{noise}^\sigma)| - |CC(H_j, H_{ck} + P_{noise}^\sigma)| \\
&= \frac{1}{\sqrt{Var(H_{ck} + P_{noise}^\sigma)}} \cdot \Big(\sqrt{Var(H_{ck})} \cdot \big(|CC(H_{ck}, H_{ck})| - |CC(H_j, H_{ck})| \big) \\
&\qquad + \big(\frac{\pm |Cov(H_{ck}, P_{noise}^\sigma)|}{\sqrt{Var(H_{ck})}} - \frac{\pm |Cov(H_j, P_{noise}^\sigma)|}{\sqrt{Var(H_j)}} \big) \Big) \\
&\approx \frac{1}{\sqrt{S^2(\mathbf{h}_{ck}) + \sigma^2}} \cdot \Big(\sqrt{S^2(\mathbf{h}_{ck})} \cdot \big(1 - |r_j^0| \big) + \big(\frac{\pm |W(\mathbf{h}_{ck}, \mathbf{p})| \mp |W(\mathbf{h}_j, \mathbf{p})|}{\sqrt{S^2(\mathbf{h}_{ck})}} \big) \Big) \cdot
\end{aligned} \tag{2}$$

One assumption underlying the deduction in (2) is that the attacked operation is assumed to be balanced [13], which is the case for most of the operations used in cryptographic algorithms. So that, given uniformly distributed random input and key, the output of the operation is also uniformly distributed. This assumption yields that the variances of the Hamming-weight of the outputs for different key hypotheses are constant when all input values are used, i.e., $Var(H_i) = Var(H_j)$ for any $1 \leq i, j \leq n$.

In Eq. (2), the variables are in the end substituted by their estimators. The exact value for $Prob(\delta_j^\sigma < 0)$ is difficult to derive analytically based on this model, requiring statistical methods out of the scope of this paper. However, we have discovered some interesting properties for the probability $Prob(\delta_j^\sigma < 0)$ based on (2). Since $Var(H_{ck})$ is constant, $S^2(\mathbf{h}_{ck})$ tends to constant when all possible input values are used. Because P_{noise}^σ and H_j are independent, by the Central Limit Theorem [14], when all possible values for input are used, the distribution of $W(\mathbf{h}_j, \mathbf{p})$ is approximately normal with expectation zero and some variance $Var[W(\mathbf{h}_j, \mathbf{p})]$, which increases on σ. Therefore, considering two arbitrary key hypotheses k_i and k_j, we can assume that $W(\mathbf{h}_i, \mathbf{p})$ and $W(\mathbf{h}_j, \mathbf{p})$ have nearly the same distribution for a fixed σ. Consequently, we can assume that when all inputs are used, δ_j^σ can be seen as a function of $|r_j^0|$. Hence, the probability $Prob(\delta_j^\sigma < 0)$ depends only on $|r_j^0|$ for some

fixed σ and the relation of $Prob(\delta_j^\sigma < 0)$ between different key hypothesis can be approximated by their relation for $|r_j^0|$. As shown in (3), for two key hypotheses k_i and k_j where $i \neq j$, their probabilities of resulting a higher correlation peak than that by k_{ck}, are approximately equal if $|r_i^0| = |r_j^0|$ and have the same relation as $|r_i^0|$ and $|r_j^0|$ if otherwise. Note that when $|r_i^0|$ is smaller than but very close to $|r_j^0|$, the probability $Prob(\delta_i^\sigma < 0)$ can be approximately equal to and not necessarily smaller than $Prob(\delta_j^\sigma < 0)$.

$$\begin{aligned} |r_i^0| = |r_j^0| &\implies Prob(\delta_i^\sigma < 0) \approx Prob(\delta_j^\sigma < 0) \\ |r_i^0| < |r_j^0| &\implies Prob(\delta_i^\sigma < 0) < Prob(\delta_j^\sigma < 0) \cdot \end{aligned} \tag{3}$$

When two balanced operations op_1 and op_2, both processing with b-bit data, are considered, we have that $Var(H_i^{op_1}) = Var(H_j^{op_2}) = b/4$ for any i and j, when all input values are used. As P_{noise}^σ is also independent of the operation, similar arguments as previously regarding $W(\mathbf{h}_j, \mathbf{p})$ can also be applied here. Hence, we can conclude that the properties in (3) hold independently of operations as long as they are carried out with data of the same sized.

5.2 Assessing $Prob(\delta_j^\sigma < 0)$ and $Prob(\mathscr{F}(\sigma))$

Based on the relation between $|r_j^0|$ and $Prob(\delta_j^\sigma < 0)$ as in (3), we define a function $h(r, \sigma)$ which takes non-negative inputs r and σ, and returns the probability that a key hypothesis with correlation coefficient $\pm r$ in the noise-free simulation, becomes the key candidate in an attack when the standard deviation of noise equals σ. Thus, the probability $Prob(\delta_j^\sigma < 0)$ can be expressed as $h(|r_j^0|, \sigma)$.

We estimate the function $h(r, \sigma)$ by applying the CPA simulation performed in Section 4 on the four demonstrated operations with 500 repetitions each. Unlike the previous experiment, we have now recorded the number of times that each hypothesis k_j results in a negative δ_j^σ, i.e. when the absolute value of the correlation by k_j is higher than that by k_{ck}, for $\sigma = 0.5, 1, 1.5, \ldots, 13$. The ratios of these numbers to 500 are then the estimation of $h(r, \sigma)$. Note that this assessment for $h(r, \sigma)$ covers only a part of the possible values for r, which however, as we will show later, is sufficient to capture the characteristics of $h(r, \sigma)$.

The results of this experiment agrees with our analyzed properties about relations between r and $h(r, \sigma)$ as in (3). Due to the lack of space, we do not show the result for each individual experiment. In order to give a clear illustration of $h(r, \sigma)$, we plot the averaged results in Fig. 4 for a few representative values of r. Figure 5 gives an example when $\sigma = 7$, which is the column of Fig. 4 where $\sigma = 7$. The plottings indicate that $h(r, \sigma)$ monotonically increases on σ for every r, and grows rapidly on r in most of the cases for σ.

We now discuss how to reason about $Prob(\mathscr{F}^\sigma)$ using $h(r, \sigma)$. An assessment of $Prob(\mathscr{F}^\sigma)$ can be derived as in (4), where every step of approximation is la-

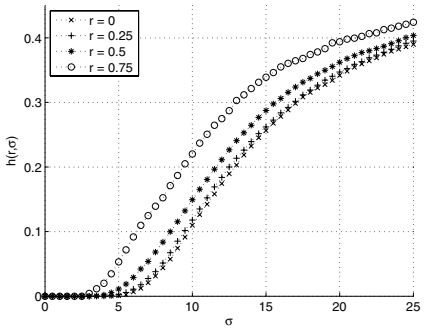

Fig. 4 Function $h(r,\sigma)$.

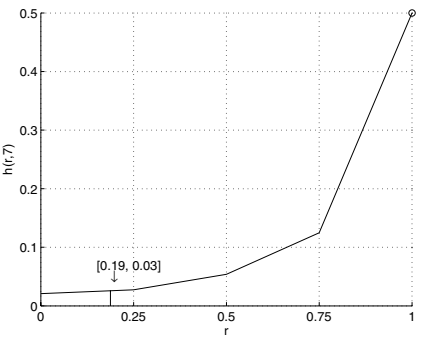

Fig. 5 Function $h(7,\sigma)$.

beled with the amount of errors that this approximation causes. That is, for $X \overset{\varepsilon}{\approx} Y$, $\varepsilon = Y - X$. Therefore, a positive ε represents overestimating and a negative one represents underestimating. We will later investigate into the precision of each individual and the overall approximations.

$$\begin{aligned} Prob(\mathscr{F}^{\sigma}) &= Prob(\{\,\delta_j^{\sigma} < 0 \mid 1 \le j \le n\,\}) \\ &\overset{\varepsilon_1}{\approx} \sum_{j=1}^{n} Prob(\delta_j^{\sigma} < 0) \;=\; \sum_{j=1}^{n} h(|r_j^0|,\sigma) \\ &\overset{\varepsilon_2}{\approx} \sum_{j=1}^{n} h(|r_j^0|,\sigma) \qquad \text{for } 1 > |r_j^0| \ge \rho \\ &\overset{\varepsilon_3}{\approx} \frac{h(1,\sigma)+h(\rho,\sigma)}{2} \cdot \#\{\, j \mid 1 > |r_j^0| > \rho \,\} \cdot \end{aligned} \tag{4}$$

For ε_1, outcomes $\{\delta_j^{\sigma} < 0\}$ for all j are not mutually exclusive. Hence, this error is related to the dependency between the outcomes, which is unknown. However, we can give a translation of ε_1 in the context of CPA attacks. Let us consider a stronger attacker who always takes the highest N correlation peaks from an attack, and later determines the correct key by brute-force methods in the order of the height of the peaks. In this case, the expected maximum number of trials that the attacker performs for each operation under attack before he finds the correct key hypothesis (or gives up) can be computed by $E[min(\#\{\, j \mid \delta_j^{\sigma} < 0\}, N)]$. Intuitively, the attacker can tolerate N wrong key candidates at maximum and a CPA attack can succeed if $|r_{ck}^{\sigma}|$ is ranked in the top N correlation peaks. Using this notation, the probability $Prob(\{\delta_j^{\sigma} < 0\})$, on the left hand side of ε_1, is equal to $E[min(\#\{\, j \mid \delta_j^{\sigma} < 0\,\}, 1)]$ and can be interpreted as the expectation of the number of trials that an attacker performs on brute-force, when he takes only one key candidate in an attack. On the other hand, the sum of $Prob(\delta_j^{\sigma} < 0)$ equals, by definition, the expectation of the total number of wrong candidates a CPA attack returns, i.e. $E[\#\{\, j \mid \delta_j^{\sigma} < 0\,\}]$, which is no less than $E[min(\#\{\, j \mid \delta_j^{\sigma} < 0\,\}, 1)]$. Therefore, ε_1 is non-negative and increases

Table 1 The value of ρ for selected σ.

σ	–2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	8	9	10	11	12	13–
ρ	$\frac{13}{16}$	$\frac{12}{16}$	$\frac{11}{16}$	$\frac{8}{16}$	$\frac{7}{16}$	$\frac{6}{16}$	$\frac{11}{32}$	$\frac{5}{16}$	$\frac{4}{16}$	$\frac{3}{16}$	$\frac{5}{32}$	$\frac{2}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{128}$	0

on σ. For example, when σ is small, noise influences less and the correlation peak for the correct key hypothesis is likely to be the highest so that ε_1 is small; and when σ is big, noise influences more and $|r_{ck}^{\sigma}|$ is unlikely to be the highest resulting a big ε_1.

The second estimation ignores the case when r is smaller than some threshold ρ. Hence, $\varepsilon_2 = -\sum_{j=1}^{n} h(|r_j^0|, \sigma) \leq 0$, for $r < \rho$. Figure 5 shows an example when $\rho = 0.19$. Although $h(r, \sigma)$ can be small when r is small, the correlation values resulted from an attack can very likely be close to zero, referring to the demonstrations in Fig. 1. Therefore, the sum of $h(r, \sigma)$ for $r < \rho$ is not necessarily small. Generally speaking, ε_2 decreases on σ and ρ for all operations. When σ grows, $h(r, \sigma)$ rises for every r and the sum of $h(r, \sigma)$ increases for $r < \rho$; when ρ increases more cases for r will be ignored by this approximation.

By making the third approximation in (4), we are actually assuming that r is equally distributed for $r \geq \rho$. This may in practice not be the case for an operation. The value of ε_3 depends on the distribution of r for $r \geq \rho$, which is subject to the operation. Taking Fig. 5 as an example, at interval $r = [\rho, 1)$, ε_3 is positive if the distribution of r is denser in the area close to ρ and is negative if the distribution of r is denser in the area close to 1. The amount ε_3 approaches zero when σ increases, whereas its relation to ρ requires more details on the distribution of r.

In summary of the previous analysis, ε_1 is non-negative and increases on σ, whereas ε_2 is non-positive and decreases on σ and ρ, somehow compensating ε_1. Amount of ε_3 is uncertain, depending on specific operations. Therefore, we can claim that our approximation for $Prob(\mathscr{F}^{\sigma})$ in (4) is not too rough and can be very close to the true value if the threshold ρ is well chosen. Deriving a function for ρ based on σ, however, requires information that is unknown to us, such as the distribution of correlation coefficient for a random operation and the exact value for $h(r, \sigma)$. Nonetheless, the values for ρ can be assessed based on our experimental results for $h(r, \sigma)$ and $Prob(\mathscr{F}^{\sigma})$ (see Section 4). The approximated ρ for selected values of σ is shown in Table 1.

5.3 A Metric for CPA Resistance

Previous section shows that given an estimated standard deviation σ for noise P_{noise}, one can find a value for ρ in Table 1 such that $Prob(\mathscr{F}(\sigma))$ can be assessed using (4). The resulting formula indicates that when σ and ρ are fixed, the probability that a CPA fails is proportional to the number of correlation peaks that are smaller

Table 2 Metrics for the operations for selected values of σ.

σ	–2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	8	9	10	11	12	13–
XOR	0	16	16	72	72	72	72	72	184	184	184	184	184	184	184	254
ADD	0	1	1	15	17	19	27	31	71	99	125	157	237	247	255	255
MUL	0	0	2	5	7	9	9	9	11	17	21	29	115	177	209	255
AES	0	0	0	0	0	0	0	0	0	4	17	35	188	223	241	255

than 1 and higher than or equal to ρ in the results of the idealized attack simulation. Intuitively, the more high correlation peaks an operation results in from a noise-free CPA attack, the more wrong key hypotheses are correlated to the correct one and the more likely the real CPA attack on this operation, where the noise fluctuates the power consumption measurements, is going to fail.

Therefore, we can deliver a metric for the resistance of an operation to CPA attack:

Definition 1. Given correlation coefficient values $\mathbf{r}^0 = (r_1^0, r_2^0, \ldots, r_n^0)$ obtained from the idealized CPA simulation on an operation, a metric for its resistance to CPA attacks where the electronic noise has a standard deviation approximately equal to σ, is the number of elements in $\mathbf{r}^0$ whose absolute values fall into interval $[\rho, 1)$, i.e.,

$$\#\{\, j \mid 1 > |r_j^0| \geq \rho, 1 \leq j \leq n \,\},$$

where, knowing σ, the threshold ρ can be obtained from Table 1.

Using Definition 1, we have calculated the metrics in Table 2 for the operations used in demonstration. It shows that the ranking of the failure rates for the operations previously obtained by attack simulations (see Section 4) is now well captured by the metrics of those operations in Table 2.

6 Validating the Metric

In this section, we discuss executing the exemplary operations on a Atmel AVR microcontroller with all switchable countermeasures off. Nowadays, most cryptographic devices available on the market come with countermeasures against side-channel analysis. Information leakage in the Atmel AVR microcontroller reports similarity to that in cryptographic devices such as smartcards, but with a customizable setting for the countermeasures. Hence, it is typically seen as a good representative for predicting the leakage of cryptographic devices in the worst scenario at an earlier stage.

Usually for a practical CPA, one needs to record the power consumption values for a large number of time samples during execution. Therefore, each input results in a power trace consisting of numerous power consumption values each of which

corresponds to a single time sample. However, only the time sample at which the output of the attacked operation is processed is relevant to an CPA attack. We denote this point of time using τ. Hence, we firstly apply the CPA attack using the complete power traces for all input and then identify this concerned time sample τ based on the resulting correlation values. The power consumption values at τ then corresponds to the power consumption vector $\mathbf{t}$ in our simulation (see Section 2). Accordingly, the correlation values at τ are then captured in $\mathbf{r}$.

We have performed the attack for a number of traces (input). In general, the more resistant an operation is to CPA attacks, the more traces is needed to obtain a clear correlation peak. The resulting correlation values at τ are shown in Fig. 6. The correct key hypothesis is plotted in black. The plotting for `XOR` is vertically symmetric every key hypothesis and its bitwise inverted value report exactly negated correlation values. Although the correct key hypothesis always results in the highest correlation coefficient after about 250 traces, ghost peaks are still very significant. For `ADD`, only one ghost peak, which is caused by the key hypothesis $k_{ck} \pm 2^7$, remains very high after 400 traces. The results for `MUL` shows that the peak for k_{ck} becomes clear after about 250 traces, followed by a few of ghost peaks. In case of `AES`, the peak at k_{ck} stands out very obviously after only about 50 traces. The results show that the physical measurements from these experiments are in conformance with our previous simulation results, thereby validating the ranking and the metric of the operations in Section 5.

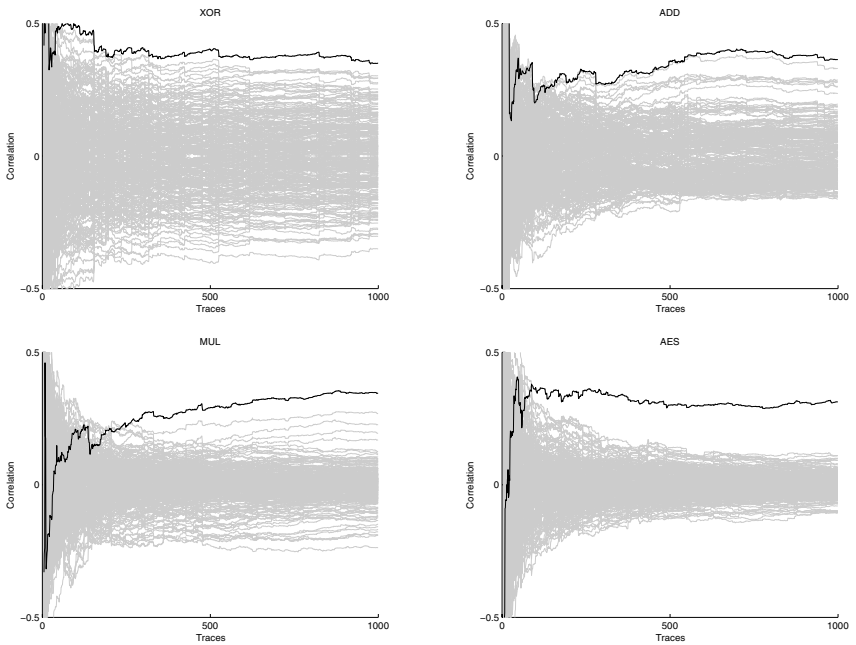

Fig. 6 The correlation values for different number of traces at τ.

7 Conclusion

In this paper, we analyze the resistance to CPA attacks for fix-sized primitive operations. By studying the results from simulated CPA attacks on a few operations that carry out 8-bit data, we provide a model for the resistance to CPA attacks. Based on this reasoning, we propose a convenient metric for measuring the resistance of an operation to the attacks and argue its validity. By demonstration, we show how this metric can be employed to rank operations with respect to their CPA resistance. Additionally, physical attacks are applied on the operations in practice on a Atmel AVR micro-controller and the results of agree well with the ranking metric proposed.

Acknowledgements We thank Jaap de Vos and Lex Schoonen for experimental support and stimulating interaction.

References

1. M.-L. Akkar, R. Bevan, P. Dischamp, and D. Moyart. Power analysis, what is now possible... In *ASIACRYPT*, pages 489–502. Springer, 2000.
2. R. Bevan and E. Knudsen. Ways to enhance differential power analysis. In *ICISC*, pages 327–342. Springer, 2002.
3. E. Brier, C. Clavier, and F. Olivier. Correlation power analysis with a leakage model. In *CHES*, pages 16–29. Springer, 2004.
4. J. Daemen and V. Rijmen. *The Design of Rijndael*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.
5. D. Gligoroski. Stream cipher based on quasigroup string transformations in $\mathbb{Z}_p^*$. *Computing Research Repository (CoRR)*, cs.CR/0403043, 2004.
6. S. Guilley, P. Hoogvorst, and R. Pacalet. Differential power analysis model and some results. In *CARDIS*, pages 127–142. Kluwer, 2004.
7. P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *CRYPTO*, pages 388–397. Springer, 1999.
8. T.-H. Le, J. Clediere, C. Serviere, and J.-L. Lacoume. A proposition for correlation power analysis enhancement. In *CHES*, pages 174–186. Springer, 2006.
9. T.-H. Le, J. Clediere, C. Serviere, and J.-L. Lacoume. How can signal processing benefit side channel attacks? *IEEE Workshop on Signal Processing Applications for Public Security and Forensics*, pages 1–7, April 2007.
10. K. Lemke, K. Schramm, and C. Paar. DPA on n-bit sized boolean and arithmetic operations and its application to IDEA, RC6, and the HMAC-construction. In M. Joye and J-J. Quisquater, editors, *CHES*, pages 205–219. Springer, 2004.
11. S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Advances in Information Security. Springer, 2007.
12. T. Messerges, E. Dabbish, and R. Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Computers*, 51(5):541–552, 2002.
13. E. Prouff. DPA attacks and S-boxes. In *FSE*, pages 424–441. Springer, 2005.
14. A. Siegel. *Statistics and data analysis: an introduction*. John Wiley & Sons,, 1988.
15. F.-X. Standaert, E. Peeters, C. Archambeau, and J.-J. Quisquater. Towards security limits in side-channel attacks. In *CHES*, pages 30–45. Springer, 2006.
16. D. Wheeler and R. Needham. TEA, a tiny encryption algorithm. In *FSE*, pages 363–366. Springer, 1994.