SCIENCE CHINA
Information Sciences

• LETTER •

February 2022, Vol. 65 129101:1–129101:3
https://doi.org/10.1007/s11432-019-2719-2

Correlation leakage analysis based on masking schemes

Jiawei ZHANG[1], Yongchuan NIU[1*] & An WANG[2*]

[1]*Data Communication Science and Technology Research Institute, Beijing,* 100191*, China;*
[2]*School of Computer Science, Beijing Institute of Technology, Beijing,* 100081*, China*

Received 25 June 2019/Revised 16 September 2019/Accepted 25 November 2019/Published online 13 May 2021

Citation Zhang J W, Niu Y C, Wang A. Correlation leakage analysis based on masking schemes. Sci China Inf Sci, 2022, 65(2): 129101, https://doi.org/10.1007/s11432-019-2719-2

Dear editor,

Masking is generally utilized to construct the first-order protection for cryptographic algorithms, but such protected designs are still susceptible to higher-order power analysis attacks. Second-order differential power analysis (DPA) [1–4] can break first-order masking countermeasures by combining the leakages of the two secret shares into a signal that is correlated with the target intermediate variable. The efficiency of the second-order DPA greatly depends on the combining function it employs and the leakage model it constructs, but the combining function is just an approximate representation of the leakage. Collision attack [5–7] is another extensively applied method for achieving an attack on masked implementations. Clavier et al. [5] utilized the reuse of masks to show the relationship among masked data in various substitution boxes (s-boxes) to perform the collision. Scalable collision attack [7] is a fault-tolerant collision attack that can keep more useful key-related information to increase the success rate. This study presents a correlation leakage model which is a precise characterization of the leakage in first-order masked implementations. It uses the correlation coefficient between the power consumptions of the two intermediate variables to describe the power leakage. Based on this leakage model, we further propose a novel second-order attack method, which is called correlation leakage analysis (CLA). No matter how large the noise is, the optimal correlation coefficient in CLA is always equal to 1.

Correlation leakage model. The proposed correlation leakage model illustrates the leakage of power consumptions of two intermediate variables, Z_1 and Z_2, that are processed during algorithm executions.

Definition 1 (Correlation leakage)**.** Assume that $L(Z_1)$ and $L(Z_2)$ denote the power consumptions of intermediate variables Z_1 and Z_2, respectively. The correlation leakage is defined by $\rho(L(Z_1), L(Z_2))$, i.e., the correlation coefficient between $L(Z_1)$ and $L(Z_2)$.

When targeting a masked implementation, we let Z_1 denote the sensitive variable $Z \oplus M$ processed at time t_1 and Z_2 denote the corresponding mask, M, processed at time t_2. Then the leakage model can be defined by Assumption 1 according to [4], where Z and M are assumed to be mutually independent and uniformly distributed over $\mathbb{F}_2^n$.

Assumption 1 (Leakage model)**.** The leakages $L(Z_1)$ and $L(Z_2)$ satisfy

$$\begin{cases} L(Z_1) = \delta_1 + \mathrm{HW}(Z \oplus M) + B_1, \\ L(Z_2) = \delta_2 + \mathrm{HW}(M) + B_2, \end{cases} \tag{1}$$

where $\mathrm{HW}(\cdot)$ denotes the Hamming weight function, and δ_1, δ_2, Z, M, B_1, and B_2 have the same definitions, as in [4].

Proposition 1. Let $L(Z_1)$ and $L(Z_2)$ satisfy (1), Then, for every $z \in Z$, we have the following equation:

$$\rho(L(Z_1), L(Z_2)|Z = z) = \frac{n - 2\mathrm{HW}(z)}{n + 4\sigma^2}, \tag{2}$$

where $Z = Z_1 \oplus Z_2$, Z is uniformly distributed over $\mathbb{F}_2^n$, and σ denotes the standard deviation of the Gaussian noise centered at zero.

The proof of Proposition 1 is given in Appendix A.

Moreover, Proposition 1 indicates that there is a precise mathematical relationship between the Hamming weight of the sensitive intermediate variable $Z_1 \oplus Z_2$ and the correlation coefficient $\rho(L(Z_1), L(Z_2))$. Additionally, this strong linear correlation makes $\rho(L(Z_1), L(Z_2))$ a better representation of the device leakage.

Based on the above analysis, the correlation leakage model is given as follows.

Definition 2 (Correlation leakage model)**.** Correlation leakage model, $\mathrm{CL}(\cdot)$, is defined as

$$\mathrm{CL}(Z) = \frac{n - 2\mathrm{HW}(Z)}{n + 4\sigma^2}, \tag{3}$$

where Z is uniformly distributed over $\mathbb{F}_2^n$, and σ denotes the standard deviation of the Gaussian noise centered at zero.

Besides, when the noise is close to zero, we can obtain the following idealized model:

$$\mathrm{CL}_{\sigma=0}(Z) = \frac{n - 2\mathrm{HW}(Z)}{n}. \tag{4}$$

* Corresponding author (email: niuyongchuan@hotmail.com, wanganl@bit.edu.cn)

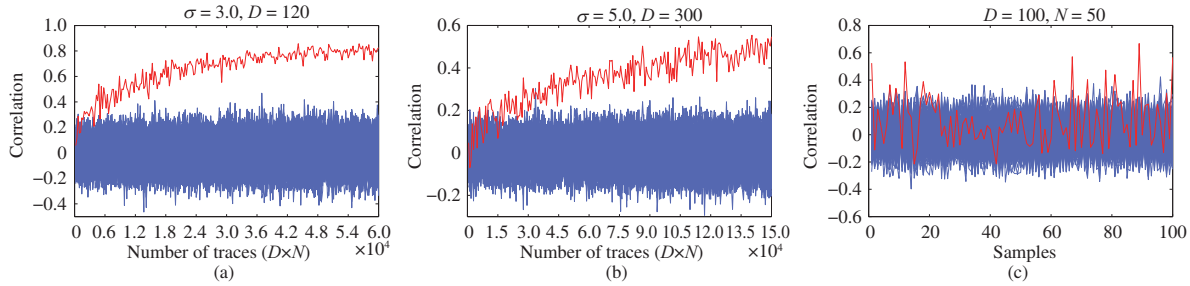

Figure 1 (a) (b) The simulated results of CLA over the number of traces based on the Gaussian noise with different standard deviations; (c) practical results of CLA using 5000 traces. The correct and incorrect key hypotheses are plotted in red and blue, respectively.

Remarkably, although the correlation leakage model is defined based on the Hamming weight model, it is still true for Hamming distance model since assumption 1 also holds under Hamming distance model according to [4].

Correlation leakage analysis. Based on the good properties of the correlation leakage model, we further propose CLA, whose general attack strategy can be stated as follows.

Step 1: Choosing two intermediate variables of the executed algorithm. The first step of the CLA is to choose two intermediate variables, which are denoted by $z \oplus m$ and m, of the cryptographic algorithm that is executed by the attacked device, based on Assumption 1. Every time the encryption executes, these two intermediate variables randomly differ, while their XOR result is only related to a small part of the plaintext and a small part of the key, which are denoted by d and k, respectively. We refer to this XOR result, z, as a function $f(d, k)$.

Step 2: Calculating correlation curves. We randomly select D plaintexts and refer to the target data blocks that are involved in calculating the intermediate results as a vector $\boldsymbol{d} = (d_1, d_2, \ldots, d_D)'$, where d_i denotes the target data block of the i-th plaintext. For each d_i, we encrypt it N times and obtain two power trace matrices of size $N \times L$, written as $\boldsymbol{T}_i$ and $\boldsymbol{T}_i^*$, which respectively correspond to the processing of the two intermediate variables chosen in step 1, where L denotes the length of the power trace corresponding to each intermediate variable. For each column of the matrix, $\boldsymbol{T}_i$ and each column of the matrix, $\boldsymbol{T}_i^*$, denoted by $\boldsymbol{\tau}_a$, $a \in [1, L]$ and $\boldsymbol{\tau}_b^*$, $b \in [1, L]$, their correlation coefficients are calculated resulting in a correlation vector $\boldsymbol{v}_i^{'} = (v_{i,1}, v_{i,2}, \ldots, v_{i,L^2})$, where $v_{i,(a-1)*L+b}$ denotes the correlation coefficient between $\boldsymbol{\tau}_a$ and $\boldsymbol{\tau}_b^*$. One data block results in one correlation vector, and hence, a correlation matrix, $\boldsymbol{V}$, of size $D \times L^2$ is generated for $\boldsymbol{d}$.

Step 3: Calculating the hypothetical intermediate values. All possible choices of k can be written as vector $\boldsymbol{k} = (k_1, k_2, \ldots, k_K)$, where K denotes the total number of all the possible choices for k. Given data vector, $\boldsymbol{d}$, and the key hypotheses, $\boldsymbol{k}$, the corresponding hypothetical intermediate values, $f(d, k)$, can be easily calculated for all D data blocks and for all K key hypotheses. Besides, a matrix $\boldsymbol{U}$ of size $D \times K$ is obtained, where $u_{i,j} = f(d_i, k_j)$, $i \in [1, D]$, $j \in [1, K]$.

Step 4: Mapping matrix $\boldsymbol{U}$ to hypothetical correlation matrix $\boldsymbol{R}$. Based on the correspondence in (4), matrix $\boldsymbol{U}$ is mapped to hypothetical correlation matrix $\boldsymbol{R}$ under the Hamming weight model.

Step 5: Comparing hypothetical correlation with real correlation. We calculate the correlation coefficients between each column $\boldsymbol{r}_a$ of matrix $\boldsymbol{R}$ and each column $\boldsymbol{v}_b$ of matrix $\boldsymbol{V}$, and a matrix $\boldsymbol{C}$ of size $K \times L^2$ is obtained, whose element $c_{a,b}$ is the correlation coefficient between $\boldsymbol{r}_a$, $a \in [1, K]$ and $\boldsymbol{v}_b$, $b \in [1, L^2]$. The line index, ck, corresponding to the highest correlation coefficient shows the maximum possible correct key.

Experiments and results. An advanced encryption standard with 128-bit key (AES-128) is taken as an example to show how the CLA attack works on masked linear layers. For most of the masked implementations of block ciphers, the on-the-fly computation of mask compensation for linear layers is generally adopted since there is insufficient memory to store all the precomputed random masks. Without loss of generality, we select the first operation of MixColumns $2 \cdot (\mathrm{Sbox}(p_{i,a} \oplus k_a) \oplus m^{'})$ and its corresponding mask compensation $2 \cdot m^{'}$ as the attack points. Then we have

$$\begin{aligned} f(d_i, k) &= 2 \cdot (\mathrm{Sbox}(p_{i,a} \oplus k_a) \oplus m^{'}) \oplus (2 \cdot m^{'}) \\ &= 2 \cdot ((\mathrm{Sbox}(p_{i,a} \oplus k_a) \oplus m^{'}) \oplus m^{'}) \\ &= 2 \cdot \mathrm{Sbox}(p_{i,a} \oplus k_a), \end{aligned} \tag{5}$$

where $p_{i,a}$ denotes the a-th byte of the i-th plaintext, and k_a denotes the a-th byte of the key.

Additionally, in the conducted simulated experiments, traces are generated by the Hamming weight of the intermediate variable involved in the computation plus a centered Gaussian noise with standard deviation, σ. Meanwhile, the attack results are shown in Figure 1(a) and (b). In the case that $\sigma = 3$, 120 plaintexts were randomly selected, and the right key hypothesis can be distinguished from wrong guesses after about 50 encryption operations for each plaintext, as depicted in Figure 1(a), while 120 encryptions for 300 plaintexts were required when $\sigma = 5$, as demonstrated in Figure 1(b).

In the conducted practical experiments, an 8-bit microcontroller STC89C52 was used as the hardware platform to perform the CLA attack in a realistic environment. We measured the power consumption of the first encryption round and identified the two target segments with a visual inspection. These two segments correspond to the intervals that likely contain the operations of the masked data and mask compensation. In our conducted experiments, 100 plaintexts were randomly selected, and 50 encryptions for each plaintext were executed. The corresponding attack result is shown in Figure 1(c); we observe that there are more than one high correlation peaks since all samples that are related to the processing of the two attacked intermediate variables are strongly correlated.

Conclusion. In this study, a novel power leakage model called correlation leakage model was presented, which utilizes the correlation coefficient between the leakages of inter-

mediate variables to represent the power leakage. By employing mathematical reasoning, the exact formula of this model was given, in which the relationship between the correlation leakage and the sensitive intermediate variable was clearly observed. Based on this leakage model, we proposed a new type of second-order attack, CLA. This CLA can break the first-order masked implementations of cryptographic algorithms; it is applicable to all the cases that can be attacked by second-order analysis. Both the simulated and practical experiments verified the effectiveness and good performance of the CLA attacks.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61872040, U1836101), National Cryptography Development Fund (Grant No. MMJJ20170201), Foundation of Science and Technology on Information Assurance Laboratory (Grant No. KJ-17-009), and Henan Key Laboratory of Network Cryptography Technology (Grant No. LNCT2019-A02).

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Chari S, Jutla C S, Rao J R, et al. Towards sound approaches to counteract power-analysis attacks. In: Proceedings of Annual International Cryptology Conference, Santa Barbara, 1999. 398–412
2 Messerges T S. Using second-order power analysis to attack DPA resistant software. In: Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems, Worcester, 2000. 238–251
3 Joye M, Paillier P, Schoenmakers B. On second-order differential power analysis. In: Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems, Edinburgh, 2005. 293–308
4 Prouff E, Rivain M, Bevan R. Statistical analysis of second order differential power analysis. IEEE Trans Comput, 2009, 58: 799–811
5 Clavier C, Feix B, Gagnerot G, et al. Improved collision-correlation power analysis on first order protected AES. In: Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems, Nara, 2011. 49–62
6 Wang A, Zhang Y, Tian W, et al. Right or wrong collision rate analysis without profiling: full-automatic collision fault attack. Sci China Inf Sci, 2018, 61: 032101
7 Niu Y, Zhang J, Wang A, et al. An efficient collision power attack on AES encryption in edge computing. IEEE Access, 2019, 7: 18734–18748