SCIENCE CHINA
Information Sciences

• HIGHLIGHT •

June 2017, Vol. 60 068101:1–068101:3
doi: 10.1007/s11432-016-0398-y

Practical two-dimensional correlation power analysis and its backward fault-tolerance

An WANG[1,2], Wenjing HU[3], Weina TIAN[4], Guoshuang ZHANG[5] & Liehuang ZHU[1*]

[1]*School of Computer Science, Beijing Institute of Technology, Beijing* 100081*, China;*
[2]*State Key Laboratory of Cryptology, P.O. Box* 5159*, Beijing* 100878*, China;*
[3]*The Institution of Beijing Jinghang Computing Communication, Beijing* 100074*, China;*
[4]*College of Bioengineering, Beijing Polytechnic, Beijing* 100176*, China;*
[5]*Science and Technology on Information Assurance Laboratory, Beijing* 100072*, China*

Received June 9, 2016; accepted September 29, 2016; published online February 13, 2017

Citation Wang A, Hu W J, Tian W N, et al. Practical two-dimensional correlation power analysis and its backward fault-tolerance. Sci China Inf Sci, 2017, 60(6): 068101, doi: 10.1007/s11432-016-0398-y

Side-channel analysis was introduced by Kocher et al. [1, 2] which marked the outbreak of this new research field in the applied cryptography area. Subsequently, many side-channel analysis methods have been published, for example, correlation power analysis (CPA) [3], template attack [4], collision attack [5,6], mutual information analysis [7] and so on, among which CPA is most widely applied in practical attacks.

The traditional CPA method takes advantage of only one leakage point of each power trace which has multiple leakage points corresponding to several instructions and intermediate values. These intermediate values may all relate to a same key byte. Canonical correlation analysis (CCA) [8] calculates the correlation between multiple weighted points and the intermediate value. However inherently, it is a multivariate to univariate method, and just uses a single intermediate value which has lower efficiency, especially when many points of a power trace are calculated. Furthermore, some other multivariate power analyses are discussed [9,10] in recent years.

Recently, we find that when an encryption algorithm executes in a microprocessor, several different intermediate values corresponding to a same key byte are operated. And each intermediate value is involved in several instructions which can cause hundreds of leakage points on power traces. If all the leakage points and intermediate values can be jointly utilized, the success rate may be improved significantly. Therefore, we consider the choice and process of those leakage points both vertically and horizontally, and take full advantage of them. According to our practical experiment on AES (advanced encryption standard) software, a 128-bit key can be recovered by only 9 power traces. The success rate is 95.8% for each key byte, and for the whole key, the success rate is 50.3%.

Two-dimensional CPA. For convenience, we take MCS-51 assembly implementation of AES algorithm for example in this letter. When n plaintext $P_1, P_2, \ldots, P_n$ are encrypted, n power trace $T^{P_1}, T^{P_2}, \ldots, T^{P_n}$ are acquired. According to analysis on assembly code, there may be m sections $T_1^{P_1}, T_2^{P_1}, \ldots, T_m^{P_1}$ in the trace T^{P_1} corresponding to m instructions $I_1, I_2, \ldots, I_m$ respectively, which leak the first byte of AES secret key k_0. On a section $T_j^{P_i}$, there are n_j points $(x_{j,1}, t_{j,1}^{P_i}), (x_{j,2}, t_{j,2}^{P_i}), \ldots, (x_{j,n_j}, t_{j,n_j}^{P_i})$ each

* Corresponding author (email: liehuangz@bit.edu.cn)
The authors declare that they have no conflict of interest.

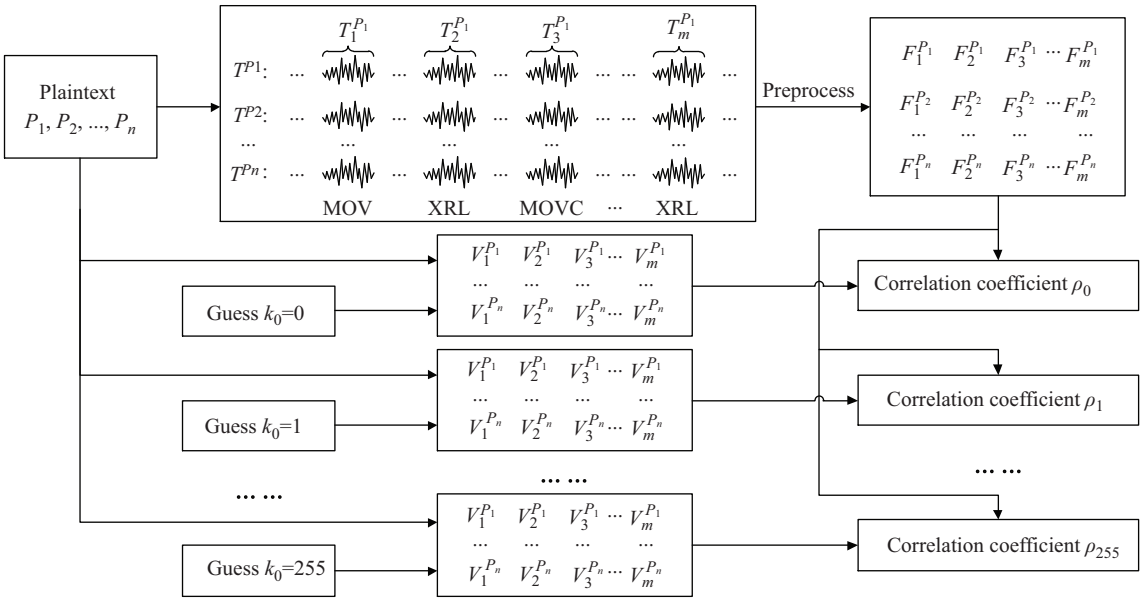

Figure 1 The flow chart of two-dimensional correlation power analysis.

of which follows the linear relation with the Hamming weight of the intermediate value. We show a CPA-like attack on the n plaintexts and $n \times m$ trace matrix, which is called two-dimensional correlation power analysis (2DCPA) and described in Figure 1.

First, because the differences among the power consumptions of $I_1, I_2, \ldots, I_m$ should be eliminated so that the correlation coefficient can be computed correctly, a preprocess function should be employed for the normalization of different $T_j^{P_i}$ for $j = 1, 2, \ldots, m$. As a result, the n_j keypoints $(x_{j,1}, t_{j,1}^{P_i}), (x_{j,2}, t_{j,2}^{P_i}), \ldots, (x_{j,n_j}, t_{j,n_j}^{P_i})$ of $T_j^{P_i}$ are averaged and scaled as a normal leakage $F_j^{P_i}$.

Second, according to a guess of k_0 and n plaintexts, all the intermediate value $V_j^{P_i}$ operated by $I_1, I_2, \ldots, I_m$ can be calculated. Therefore, each correlation coefficient ρ_k of the $n \times m$ pairs $(\{HW(V_j^{P_i})\}, \{F_j^{P_i}\})$ corresponding to the key guess $k_0 = k$ can be figured out. Finally, the value of k corresponding to the maximum ρ_k is correct key.

Normalization of trace sections. Firstly, in j-th trace section, n_j keypoints which leak information should be chosen from the x-axis of power trace. During this process, x-coordinates $\{x_{j,l} | l = 1, 2, \ldots, n_j\}$ of n_j keypoints from the j-th trace section is determined. After that, many plaintexts are chosen randomly and encrypted, and their corresponding power traces are acquired. Then their average trace $\{\overline{T}_j | j = 1, 2, \ldots, m\}$ of n_j, a standard reference trace, can be gotten. We express the point in this trace whose x-coordinate is x as $(x, \overline{T}_j(x))$.

On the one hand, we give a basic normalization. For the n_j keypoints $\{(x_{j,l}, t_{j,l}^{P_i}) | l = 1, 2, \ldots, n_j\}$ of each $T_j^{P_i}$, the mean value of their deviations (actually deviations represent signals of every point) $\frac{1}{n_j}\sum_{l=1}^{n_j}(t_{j,l}^{P_i} - \overline{T}_j(x_{j,l}))$ is computed. Then it is multiplied by a normalization factor c_j, i.e.,

$$F_j^{P_i} = c_j \times \frac{1}{n_j}\sum_{l=1}^{n_j}(t_{j,l}^{P_i} - \overline{T}_j(x_{j,l})).$$

The normalization factor c_j is defined as

$$c_j = \frac{c}{\frac{1}{n_j}\sum_{l=1}^{n_j}(\overline{y}_{j,l} - \overline{T}_j(x_{j,l}))}.$$

Here c is a constant, and $\overline{y}_{j,l}$ means the average y-coordinate corresponding to the lowest Hamming weights of the intermediate values in the point whose x-coordinate is $x_{j,l}$.

On the other hand, the basic normalization can be further improved if the adversary can compute the signal-to-noise ratio (SNR) in advance. For the n_j keypoints $\{(x_{j,l}, t_{j,l}^{P_i}) | l = 1, 2, \ldots, n_j\}$ of each $T_j^{P_i}$, the weighted mean value of their deviations, $\frac{1}{n_j}\sum_{l=1}^{n_j}\mathrm{SNR}_{j,l}(t_{j,l}^{P_i} - \overline{T}_j(x_{j,l}))$ is computed instead of $\frac{1}{n_j}\sum_{l=1}^{n_j}(t_{j,l}^{P_i} - \overline{T}_j(x_{j,l}))$. Simultaneously, the normalization factor c_j can also be defined as

$$c_j = \frac{c}{\frac{1}{n_j}\sum_{l=1}^{n_j}\mathrm{SNR}_{j,l}(\overline{y}_{j,l} - \overline{T}_j(x_{j,l}))}.$$

Therefore, the final normalization function can be

defined as

$$F_j^{P_i} = c_j \times \frac{1}{n_j} \sum_{l=1}^{n_j} \mathrm{SNR}_{j,l}(t_{j,l}^{P_i} - \overline{T}_j(x_{j,l})).$$

Implementation and efficiency. We implement AES algorithm as MCS-51 assembly codes on AT89S52 processor of MathMagic side-channel analyzer. With sampling rate 1 GSa/s, the power consumption can be acquired accurately during the encryption.

We first aim at the output value of S-box and mount a traditional CPA with the keypoints integrated by raw integration. In this attack, 50 keypoints corresponding to only one MOV instruction leaking the output of S-box are analyzed. According to our experiments, we can recover the correct key byte with about 120 power traces at least.

When attacking by CCA method, we count 10 leakage sections from the trace whose operations all relate to the output of S-box. Under this environment, 9 power traces are enough to recover the target key byte. The success rate by CCA attack is 55.8% for a key byte and $55.8\%^{16} \approx 0.0088\%$ for a 128-bit key.

With 2DCPA approach, we can recover a key byte by 9 power traces with the success rate 95.8% which is more efficient than CCA. Therefore, to recover all the 128-bit key by 2DCPA, the success rate is $95.8\%^{16} \approx 50.3\%$.

Discussion on backward fault-tolerance. With a few traces, the recovered key by 2DCPA may be wrong with certain probability. If we can divide the 16 key bytes into 4 groups and detect them respectively, the wrong group may be detected. According to the experiment result above, the success rate of each recovered 4-byte key is $95.8\%^4 \approx 84.2\%$ with 9 power traces, and the number of wrong recovered key groups follows binomial distribution. In other words, for the number of wrong recovered key groups, 0, 1, 2, 3, and 4, its probability is 0.503, 0.377, 0.106, 0.013, and 0.001, respectively.

If the key is wrong, the most likely case (with probability $\frac{0.377}{1-0.503} \approx 0.759$) is that there is just one wrong group. In this situation, we can save this experiment to find the correct 128-bit key by traversing this 4-byte key with the other 12-byte key being certain. By contrast, if there are more than two wrong groups we consider the experiment is unsuccessful. Therefore, the success rate may be improved from 50.3% to 0.503+0.377=88.0% with exhaustive search. In practice, this fault-tolerance may be implemented according to the leakage from the backward operations.

Conclusion. In this article, we present a closed form to perform 2DCPA on just 9 power traces. In contrast to traditional approaches, we choose and process all the leakage keypoints both vertically and horizontally, and employ basic or SNR-based normalization to preprocess the power traces. Then we demonstrate that this technique performs better than traditional methods such as CPA and CCA.

We think that 2DCPA seem like a simplified version of the soft analytical side-channel attacks [10] which is an almost perfect theoretical side-channel analysis for cryptographic software with simulation experiments. Our method is a practical and concrete scheme, which does not require much pre-treatment and pre-analysis. As a result, we show the practical experiment results based on the AT89S52 single-chip.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61402252, 61402536), Beijing Natural Science Foundation (Grant No. 4162053), Foundation of Science and Technology on Information Assurance Laboratory (Grant No. KJ-14-006), and Beijing Institute of Technology Research Fund Program for Young Scholars.

References

1 Kocher P. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: CRYPTO 1996, Lecture Notes in Computer Science, vol. 1109. New York: Springer, 1996. 104–113
2 Kocher P, Jaffe J, Jun B. Differential power analysis. In: CRYPTO 1999, Lecture Notes in Computer Science, vol. 1666. New York: Springer, 1999. 388–397
3 Brier E, Clavier C, Olivier F. Correlation power analysis with a leakage model. In: CHES 2004, Lecture Notes in Computer Science, vol. 3156. New York: Springer, 2004. 16–29
4 Chari S, Rao J R, Rohatgi P. Template attacks. In: CHES 2002, Lecture Notes in Computer Science, vol. 2523. New York: Springer, 2003. 13–28
5 Schramm K, Wollinger T, Paar C. A new class of collision attacks and its application to DES. In: FSE 2003, Lecture Notes in Computer Science, vol. 2887. New York: Springer, 2003. 206–222
6 Wang A, Wang Z Y, Zheng X X, et al. Efficient collision attacks on smart card implementations of masked AES. Sci China Inf Sci, 2015, 58: 052107
7 Gierlichs B, Batina L, Tuyls P, et al. Mutual information analysis. In: CHES 2008, Lecture Notes in Computer Science, vol. 5154. New York: Springer, 2008. 426–442
8 Oswald D, Paar C. Improving side-channel analysis with optimal linear transforms. In: Smart Card Research and Advanced Applications 2013, Lecture Notes in Computer Science, vol. 7771. New York: Springer, 2013. 219–233
9 Bogdanov A, Kizhvatov I. Beyond the limits of DPA: combined side-channel collision attacks. IEEE Trans Comput, 2012, 61: 1153–1164
10 Veyrat-Charvillon N, Gerard B, Standaert F X. Soft analytical side-channel attacks. In: ASIACRYPT 2014, Lecture Notes in Computer Science, vol. 8873. New York: Springer, 2014. 282–296