Improved Power Analysis Attack Based on the Preprocessed Power Traces

Xueyang Han, Qiuliang Xu(✉), Fengbo Lin, and Minghao Zhao

School of Computer Science and Technology, Shandong University, Jinan, China
xueyanghan@hotmail.com, {xql,linfb}@sdu.edu.cn, zhaominghao@hrbeu.edu.cn

Abstract. In recent years, side-channel attacks have become a most powerful attack performed on cryptographic devices. And many side-channel attack methods have sprung up, such as time attacks, electromagnetic radiation attacks, power analysis attacks including simple power attack, differential power attack, correlation power attack, etc. And the correlation power attack has become the most common side-channel attack. In this paper, we introduce a method to improve the correlation power attack (CPA). Our method is mainly to preprocess the recorded power consumption of a cryptographic device. During the procedure, we introduce a four-dimension vector to express the basic unit which we deal with. And also we give the steps of performing our improved CPA (named as ICPA). Then the experiment shows that the ICPA method enhances the efficiency. Meanwhile, we briefly show that preprocessing power traces increases the signal-to-noise ratio (SNR) comparing with non-preprocessed power traces.

Keywords: Side-channel attack · Correlation power attack · ICPA · Power model · SNR

1 Introduction

In side-channel attacks, power analysis attack are the most widely used methods. Because of its easiness of implementation, high success rate and high effectiveness, since put forward, it attracts many researchers to study. Power analysis attacks mainly measure the power consumption when physical cryptographic devices encrypt or decrypt input data, then use statistical methods to crack the cryptographic algorithm and to get the secret key. According to the analysis principle of side-channel information, it can be divided into simple power analysis (SPA), differential power analysis (DPA) and correlation power analysis (CPA). In power analysis attacks, the leaked information is recorded as power traces. In order to perform CPA, it is necessary to collect enough power traces.

The success rate of such attacks largely depends on the measured data and the power model. In classical CPA, attackers first choose a proper intermediate result of executed algorithm, then measure the power consumption (that is

This work is supported by the National Natural Science Foundation of China under grant No. 61173139 and No. 61572294.

X. Huang et al. (Eds.): GPC 2016, LNCS 9663, pp. 278–289, 2016.
DOI: 10.1007/978-3-319-39077-2_18

recorded as power traces), thirdly calculate the hypothetical intermediate values and mapping them to power consumption values, finally perform correlation analysis on the hypothetical power consumption and measured power traces. As a result, the higher the correlation coefficient is, the better the hypothetical consumption and the measured power trace match. So the hypothetical key is more likely to be the real one.

But in the classical CPA steps, CPA directly deals with the recorded power traces. Inevitably, there are lots of noises in them which have an non-ignorable effect on the results. In this paper, we improve the correlation power attack by preprocessing recorded power traces. The main difference of our method is that the ICPA method is based on a new group of power traces (we call it 'the difference power traces') calculated by the recorded power traces, not on the original recorded power traces. In this way, we eliminate the const component and some electronic noise for every power trace. In order to acquire the difference power traces, we give two scenarios. In first scenario, we pick one power trace from recorded power traces as a base power trace. In second scenario, we collect another group of power traces of the cryptographic device fed with constant input, and calculate the base power trace by these traces. Based on the base power trace, new power traces can be obtained by calculating the difference of power traces and the base power trace. It's well known lots of power traces should be collected to perform CPA, which takes much time. As for the ICPA, we record fewer power traces and take less sampling time than CPA but get approximate or even higher correlation coefficient. In second scenario, the time of recording power traces of constant input is much less than the saved time of sampling time.

The article structure consists of the following components. First, we briefly introduce the Hamming weight model [1] and the signal-to-noise ratio concepts. Then we introduce a vector expression method to express the unit of power traces and to demonstrate how to preprocess the recorded power traces. Meanwhile we introduce the difference Hamming weight and briefly show that it simulates the difference power consumption well. After that we reveal the total steps of the ICPA and show it graphically. We use a simple experiment to exhibit the results of the ICPA. In the end, we give an outline of some future works.

2 Related Work

The side-channel attack refers to passing by the complicated analysis of the encryption algorithm and using the physically leaked information such as execution time, power consumption, electromagnetic radiation etc. of cryptographic algorithms' hardware implementation, combined with statistical theory to crack password system or exploit secret information of cryptographic devices. Nowadays, it tends to be more diverse.

Since proposed, side-channel attacks have received people's attention. In 1995, Paul Kocher first puts forward the concept of time attack [2]. Time attack gets the secret key of some encryption algorithms (such as DES, AES, RSA)

by accurately measuring the time consumed by a physical device. This is the first article on side channel attacks. In 1998, power analysis attack [2] is first proposed by Kocher. In the side-channel attack, power analysis attacks are the most effective and easiest ways to implement, and therefore favored by many cryptographic scholars. The basic working principle of power analysis attacks is to analyse the correlation between the instantaneous power consumption of a device and the device's operations and data it performs. In 1999, Kocher et al. first proposed SPA method in [2] and successfully attacked the DES algorithm implemented by hardware. SPA is a direct analysis technology to the power consumption collected during the execution of the cryptographic algorithm. At the same time, DPA is proposed by Kocher et al. in the literature [2]. Different from SPA attack, DPA attack does not need to get hold of the implementation details of cryptographic algorithm, and has certain immunity to noise. Currently DPA has developed a variety of forms, Mono-Bit DPA, Multi-Bit DPA, First-Order DPA, Higher-Order DPA and so on. Its standard form [3] shows that DPA is based on divide-conquer strategy, that is to say different parts of the key (usually referred to "sub-key") resume separately. Later, Messerges [4] apply this method to public key cryptosystem. Then Walter, Klima and others analyse RSA algorithm further. In 1999, Chari put forward the concept of correlation power analysis [5], they realized the AES algorithm on ST16 smart card, and successfully carried out the attack with CPA. Brier et al. [6] give a full insight on the data leakage and use the correlation power analysis (CPA) to identify the parameters of the leakage model. Then they show that efficient attacks can be performed against unprotected implementations of many algorithms such as DES or AES. In 2006, Le [7] and others summarize and expand the concept of the multi-bit DPA, proposed Partitioning Power Analysis (PPA). The energy is divided into multiple sets not only two and the difference is not the difference between two sets, but the algebra weighted sum of multi-means of power traces.

In early study, DPA mainly focused on attacking some specific algorithms (such as DES, RSA and ECC, etc.) and equipments (such as smart cards, DSP processors, etc.) and defenses are made according to the characteristics of DPA. In [8], Chari et al. put forward some general DPA countermeasures and formal methods to assess the effectiveness of these defensive countermeasures. Dakshi put forward the evaluating method [9] for electromagnetic leakage. Its core idea is to analyse the leaked electromagnetic quantitatively by the signal detection theory, combined with information theory. The literature [10] demonstrates a safety performance assessment method of power consumption leakage. This method analyses the security of the differential power attack mainly by the Hamming weight model.

3 Preliminaries

Before focusing on our target attacking method, we specify the definition of basic notations and concepts.

3.1 Hamming Weight Model

Classically, most power consumption analysis are based upon the Hamming weight model or the Hamming distance model which are commonly considered to be good models for power consumption. If we have a data word D, $D = d_0 d_1 \cdots d_j \cdots d_{m-1}$, with the bit values $d_j = 0$ or 1. Its Hamming weight is defined as

$$H(D) = \sum_{j=0}^{m-1} d_j \tag{1}$$

Namely, it is the number of bits set in D. If D is composed of m independent and uniformly distributed bits, its mean is $E(H(D)) = m/2$ and variance is $Var(H(D)) = \sigma^2_{H(D)} = m/4$. As for the Hamming distance, it always accompany with an initial state R. And the Hamming distance of D is depicted as $H(D \otimes R)$, so the Hamming weight is indeed a special case of the Hamming distance when R equals zero.

In power analysis attacks, there are many power models that can simulate power consumption, such as bit model, Hamming weight model and Hamming distance model, etc. Until now, it is widely believed that the consumed power depends on the energy required to flip the bits from one state to the next. So the Hamming weight model and distance model is a proper way to model the power consumption. The number of flipping bits from X to Y can be depicted as $H(X \otimes Y)$. In power analysis, the component X can be the chosen intermediate result, which is a function of a input data and a hypothetical key. The component Y is a reference state which is a constant machine word. So in general we substitute V's Hamming weight $H(V)$ for $X \otimes Y$'s Hamming distance $H(X \otimes Y)$. Therefore the power model for the data dependency can be written as:

$$W = \alpha H(V) + \beta \tag{2}$$

where α is a scalar gain between the Hamming weight $H(V)$ and the power consumed W, and β stands for the remaining power consumption except the data dependency part.

3.2 Signal-to-Noise Ratio (SNR)

Power analysis attacks exploit a fact that the measured power consumption of cryptographic devices depends on the processed data and the performed operations. Besides, there are two additional non-ignorable factors in practice, the noise component and a constant component. Therefore, for each point of a power trace, it is possible to model the point as the sum of the above four factors.

$$P_{total} = P_{op} + P_{data} + P_{el.noise} + P_{const} \tag{3}$$

The components P_{op}, P_{data}, $P_{el.noise}$, P_{const} stand for the operation-dependent component, data-dependent component, the noise component and a const component, respectively. In a digital environment, an SNR is the ratio between the

signal and the noise component of a measurement. When it comes to power consumption, the signal can be described as $P_{op} + P_{data}$ and the noise as $P_{el.noise}$ if we considered the input data bits is uniformly distributed. So the SNR can be calculated by the formula:

$$SNR = \frac{Var(P_{op} + P_{data})}{Var(P_{el.noise} + P_{const})}. \tag{4}$$

4 Getting Preprocessed Power Traces

In this section, we introduce another method to express the recorded power traces. Every point of a power trace is depicted as a vector. The preprocessed power traces is calculated based on this expression.

4.1 Vector Expression of a Power Trace Point

In the first step of power analysis attack, a large number of power traces should be recorded when the devices encrypt or decrypt different data blocks. Each point of a power trace can be modeled as formula (3). Because four components are independent with each other, they can be expressed as a four-dimension vector (we call it power trace vector):

$$\overrightarrow{P_{total}} = (P_{op}, P_{data}, P_{el.noise}, P_{const}) \tag{5}$$

Now, we give some definitions and properties here:

(1) Additivity

$$\begin{aligned}\overrightarrow{P_{total_1}} + \overrightarrow{P_{total_2}} &= (P_{op_1}, P_{data_1}, P_{el.noise_1}, P_{const_1}) \\ &\quad + (P_{op_2}, P_{data_2}, P_{el.noise_2}, P_{const_2}) \\ &= (P_{op_1} + P_{op_2}, P_{data_1} + P_{data_2}, P_{el.noise_1} + P_{el.noise_2}, P_{const_1} + P_{const_2})\end{aligned} \tag{6}$$

(2) Scalar-multiplication

$$\begin{aligned}\lambda\overrightarrow{P_{total}} &= \lambda(P_{op}, P_{data}, P_{el.noise}, P_{const}) \\ &= (\lambda P_{op}, \lambda P_{data}, \lambda P_{el.noise}, \lambda P_{const_1})\end{aligned} \tag{7}$$

where λ is a scalar number.

(3) The length of power trace vector

$$\|\overrightarrow{P_{total}}\| = \sqrt{P_{op}^2 + P_{data}^2 + P_{el.noise}^2 + P_{const}^2} \tag{8}$$

Note, for the same point of a power trace, the length of power consumption vector doesn't equal to P_{total}. But they have a similarity: both of them increase with the increase of each component. The vector $\overrightarrow{P_{total}}$ direction stands for the direction of the voltage or electronic current, which is similar to the sign of P_{total}.

4.2 Preprocessing Recorded Power Traces

In general, we choose second scenario to get the base power trace. In this scenario, we collect two groups of power traces G^A and G^B. G^A is fed with const input while G^B with the random input. The base power trace is based on the group G^A. For each point of the base power trace, it's represented as:

$$\overrightarrow{P_{total}^{base}} = (P_{op}^{base}, P_{data}^{base}, P_{el.noise}^{base}, P_{const}^{base}) \tag{9}$$

where P_{total}^{base} can be the mean of group G^A, that is

$$\overrightarrow{P_{total}^{base}} = \frac{1}{N} \sum_{i=1}^{N} \overrightarrow{P_{total_i}} \tag{10}$$

in which N is the number of power traces collected in group G^A. The vector $\overrightarrow{P_{total}^{base}}$ can be calculated according to the properties (additivity and scalar-multiplication), given in the formulas (6) and (7). The method to calculate each point vector of new power traces is as follows:

$$\overrightarrow{T_j} = \overrightarrow{P_{total_j}} - \overrightarrow{P_{total}^{base}} = (P_{op_j} - P_{op}^{base}, P_{data_j} - P_{data}^{base}, P_{el.noise_1} - P_{el.noise}^{base}, P_{const_j} - P_{const}^{base}) \tag{11}$$

Because of fixed operations executed in device, The P_{op} component is identical in every $\overrightarrow{P_{total}}$. And it is the same to P_{const} component if we keep all traces collected in the same external environment. Obviously, there are two components in $\overrightarrow{T_j}$ that equal to 0. Therefore, the formula (11) can be simplified as follows:

$$\overrightarrow{T_j} = (0, P_{data_j} - P_{data}^{base}, P_{el.noise_1} - P_{el.noise}^{base}, 0) \tag{12}$$

In the new power traces (we also call it difference power traces), the signal corresponds to $data_j - P_{data}^{base}$ and the noise corresponds to $P_{el.noise_1} - P_{el.noise}^{base}$. According to the SNR definition (4) and the formula (12), the signal-to-noise ratio SNR of the difference power traces is apparently bigger than SNR in (4), noticing that $Var(P_{op})$ equals 0 because of the same performed operations when it is applied to specific power consumption attacks.

4.3 Corresponding Difference Hamming Weight Model

It is commonly known that Hamming weight model can simulate the power consumption well. According to the difference power traces, we use a difference Hamming weight model as follows and show that it is a good simulation for difference power consumption.

$$W^* = W^r - W^c = \alpha(H(r) - H(c)) + (\beta_r - \beta_c) \tag{13}$$

Correlation analysis refers to analysing the relative degree of two or more relevant variables. Traditionally, it is usual to use the correlation factor ρ_{XY} to evaluate

the linear fitting rate of variables X and Y. As for the difference of Hamming weight and the difference power consumption, we introduce $\rho_{W^*H^*}$ to describe the relationships.

$$\rho_{W^*H^*} = \frac{cov(W^*, H^*)}{\sigma_{W^*}\sigma_{H^*}} = \frac{\alpha\sigma_{H^*}}{\sigma_{W^*}} = \frac{\alpha\sigma_{H^*}}{\sqrt{\alpha^2\sigma^2_{H^*} + \beta^{*2}}}$$
$$= \frac{\alpha\sqrt{m}}{\sqrt{m\alpha^2 + 4\sigma^2_{\beta^*}}} = \frac{1}{\sqrt{1 + \frac{4\sigma^2_{\beta^*}}{\alpha^2 m}}} \tag{14}$$

with $H^* = H(r) - H(c)$ and $\beta^* = \beta_r - \beta_c$. Here, the input of Hamming weight $H(r)$ is a random number and the input of $H(c)$ is constant. The two groups of input comes from the input of G^B and G^A, respectively. β_r and β_c are the corresponding noise, respectively. The equation has the property: $|\rho_{W^*H^*}| \leq 1$. When the variance of noise tends to 0, we get a perfect Hamming weight model $|\rho_{W^*H^*}| = 1$.

5 The Improved Correlation Power Attack

In classic CPA attacks, there exists a general attack strategy which contains several basic steps. Compared to this, the ICPA has some different operations in each step. Based on our method introduced above, our method is mainly composed of the four following steps.

Step 1: Select an Attacking Point. In the first step, we should select an intermediate result in the encryption algorithm as the attacked point. Be aware that the intermediate result should be a result of a function $f(r_i, k_j)$ where r_i and k_j are known to us. In fact, r_i may be the input data, and k_j is a part of the hypothetical key.

Step 2: Measuring the Power Consumption and Calculate the Difference Power Traces. There are two parts in this step. Firstly, we recorded the power consumption by some special equipments when the cryptographic device performs encryption operations. Here we use two input modes. The first input mode is applied in G^A and the other input mode is applied in G^B. In the first input mode, we write every input as $r_1, r_2 \cdots r_i \cdots r_d$. Note that r_i is generated by a random generator, so each input r_i is most likely to be different from each other. However in the second input mode, all inputs $c_1, c_2 \cdots c_i \cdots c_d$ is the same, that is to say, $c_1 = c_2 = \cdots = c_i = \cdots = c_d$. Respectively, the recorded trace of each input is denoted by T_i^r or T_i^c, where $1 \leq i \leq d$. Note that d is the number of our input data or recorded traces.

Secondly, we calculate the difference power traces. Every difference power trace can be calculated by the operation $T_i^* = T_i^r - T^{base}$. This operation is based on the difference of power trace vector at the same time point. The base trace T^{base} is the mean of T_i^c. Figure 1 shows the process of getting the difference power traces. In the figure, the subscript s stands for the length of a power trace.

Step 3: Calculating the Hypothetical Difference Power Consumption. The aim of this step is to obtain the hypothetical difference power consumption. Firstly we calculate the hypothetical intermediate value $f(r_i, k_j)$ with $1 \leq i \leq d$ and $1 \leq j \leq K$. The k_j is a hypothetical key and K is the number of possible choices which usually equals to 256. So the key space of hypothetical key can be denoted as $(k_1, k_2, \cdots, k_K)$. Then under the technique of the difference Hamming weight model proposed before, we use the hypothetical intermediate values to simulate the difference power consumption $H^*_{i,j}$. Figure 2 shows the overall process of how to get the difference Hamming weight.

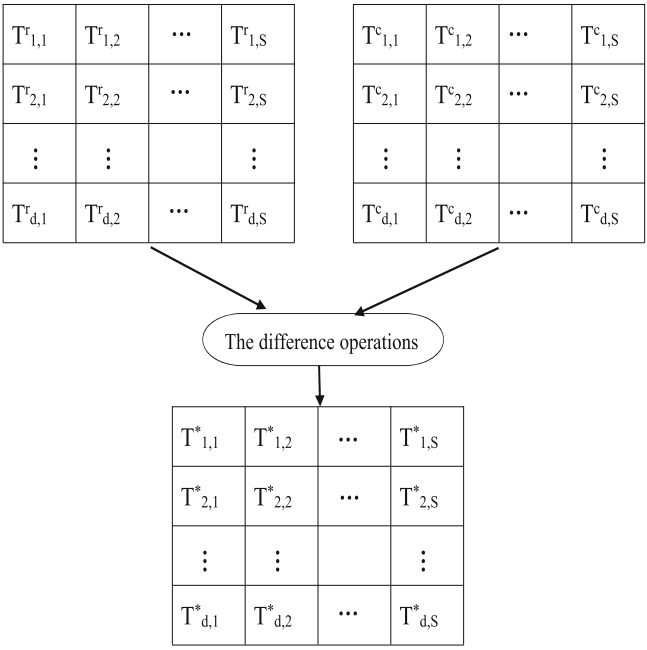


Fig. 1. Calculating the difference power traces

Step 4: Correlating the Hypothetical Difference Power Consumption with the Difference Power Traces. In this step, we calculate the correlation coefficient to response the relationships between the hypothetical difference power consumption and the difference power traces. As we all know the correlation coefficient $\rho_{(X,Y)}$ is a commonly used way to measure a linear relationship between two variables. And we have proved that the difference Hamming weight model is a good simulation for the difference power consumption in the last section. Plenty of power trace samples is necessary to estimate the correlation coefficient.

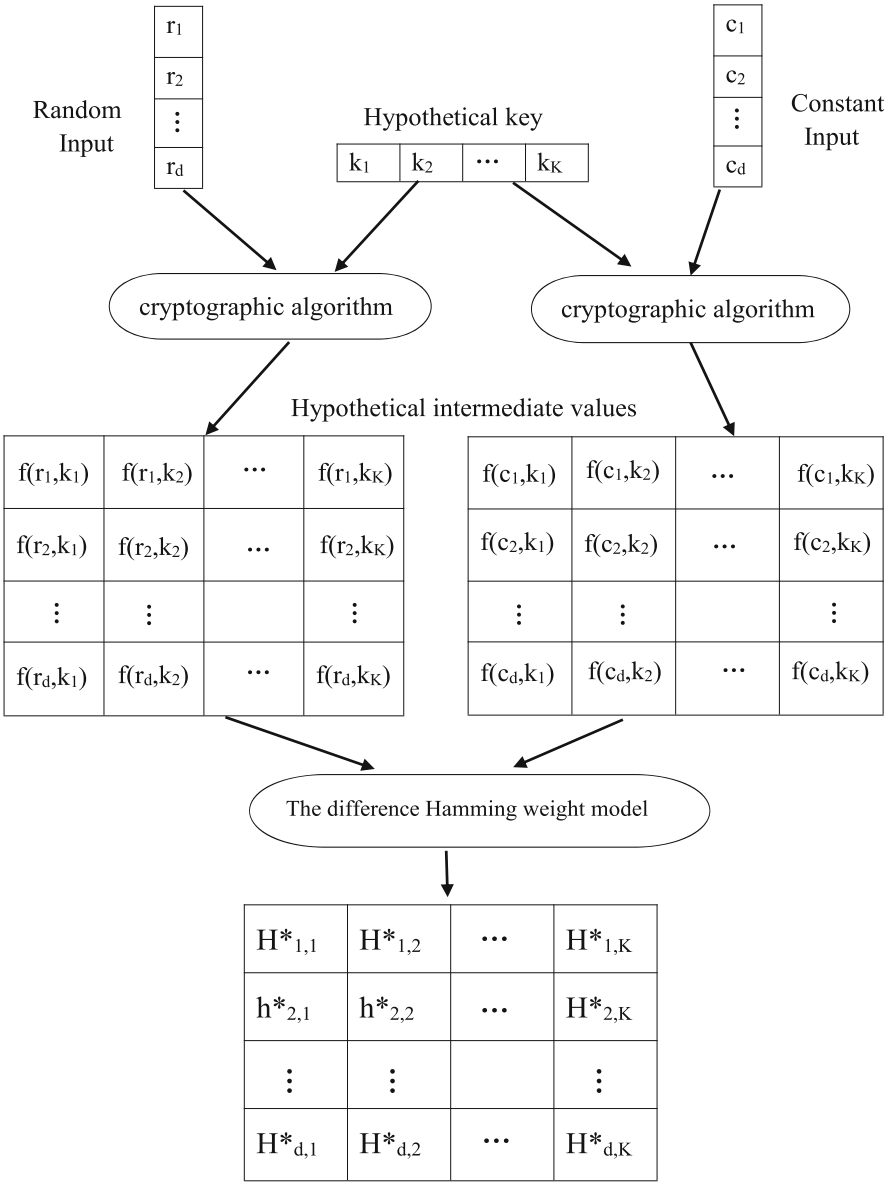


Fig. 2. The process of calculating the hypothetical power consumption

$$\rho_{(T^*,H_j^*)} = \frac{N\sum_{i=1}^{N}\|\overrightarrow{T_i^*(t)}\|H_j^* - \sum_{i=1}^{N}\|\overrightarrow{T_i^*(t)}\|\sum_{i=1}^{N}H_j^*}{\sqrt{N\sum_{i=1}^{N}\|\overrightarrow{T_i^*(t)}\|^2 - (\sum_{i=1}^{N}\|\overrightarrow{T_i^*(t)}\|)^2}\sqrt{N\sum_{i=1}^{N}H_j^{*2} - (\sum_{i=1}^{N}H_j^*)^2}} \tag{15}$$

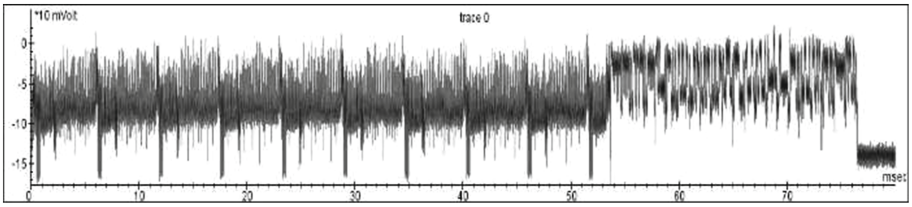
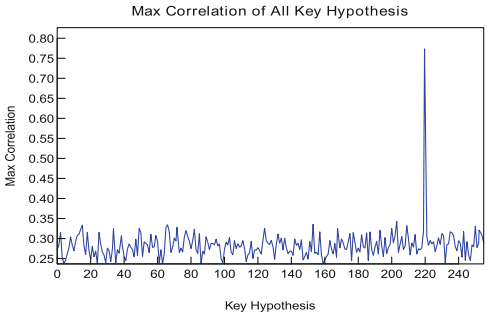


Fig. 3. Record power trace

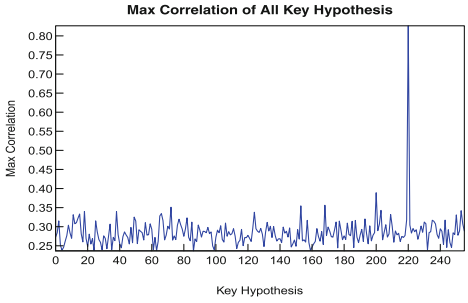
In this formula, the $\overrightarrow{T_i^*(t)}$ stands for the value in $\overrightarrow{T_i^*}$ at time t. The largest correlation coefficient $\rho_{(T^*,H_j^*)}$ shows that the hypothetical key k_j is most likely to be the real key used in the cryptographic device.

6 Results of the ICPA

In this section, we compare the performance between CPA and the ICPA. The experimental is performed on an IC chip which integrates AES algorithm without countermeasures. The environment is composed of a power tracer, an



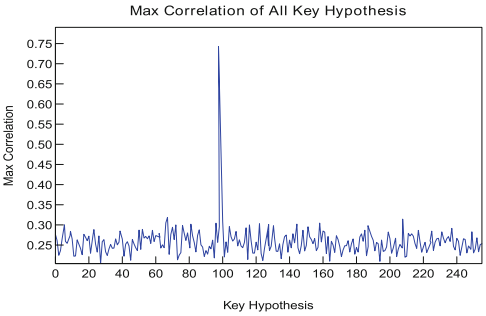
(a) Result of CPA attack



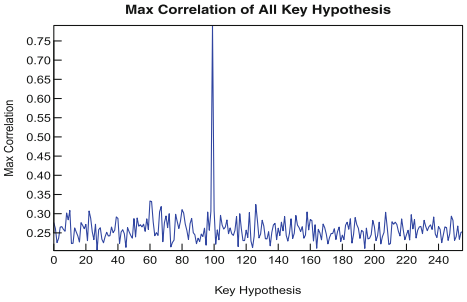
(b) Result of ICPA attack

Fig. 4. The first byte of key hypothesis

oscilloscope and a set of software such as Inspector. In the experiment, CPA deals with recorded power traces shown in Fig. 3. In the ICPA, we just need about 80 % recorded power traces to calculate the difference power traces. The AES algorithm uses the round key byte by byte in the AddRoundKey step, it is feasible to perform exhaustive attack in 256 key space. In experiment we write programs to calculate the new power traces and the maximum correlation factor of each key byte. For every byte of AES key, we get 256 correlation factors. Each hypothetical key byte corresponds to one correlation factor. Then we create line chart. Each group contains two figures below.



(a) Result of CPA attack



(b) Result of ICPA attack

Fig. 5. The second byte of key hypothesis

The figure (a) represents the correlation factor of CPA and the figure (b) is about the results of ICPA attack. Here are line charts of the first two bytes of AES round key. From the line chart, we learn that the correlation factor of wrong hypothetical keys converge to zero while the correlation coefficient of the correct key guess is close to 1. In Fig. 4, the biggest "Max Correlation" appears on the x-coordinate 220, which means $220(0xDC)$ is most likely to be the real key byte. Both figure (a) and (b) have the biggest "Max Correlation" on the x-coordinate 220, but apparently figure (b) have the bigger "Max Correlation" here. Hence we increase the "Max Correlation" of the correct key by about 6.4 %. As for Fig. 5,

the value is increased by about 6.6 %. Because we use fewer power traces and acquire equivalent or even better results, the ICPA method is more efficient.

7 Conclusion and Future Work

In this paper, we introduce a method to improve the power analysis attack. The main idea is to reduce some noise in recorded power traces. In order to preprocess power traces, we introduce a data structure to express the basic unit of power traces. Every power unit is modeled as a vector. The difference power traces are based on the vector expression and the base power trace. Then we demonstrate full steps of the ICPA. The experiment above shows that ICPA uses fewer power traces and get better results.

In the next future, we'll work further on the following aspects. Firstly, more experiments should be carried out on the cryptographic devices with countermeasures or other cryptographic algorithm. Secondly, we'll research on other side-channel analysis and compound attack based on this method.

References

1. Akkar, M.-L., Bévan, R., Dischamp, P., Moyart, D.: Power analysis, what is now possible. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 489–502. Springer, Heidelberg (2000)
2. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
3. Mangard, S., Oswald, E., Standaert, F.-X.: One for all–all for one: unifying standard differential power analysis attacks. IET Inf. Secur. **5**(2), 100–110 (2011)
4. Messerges, T.S.: Using second-order power analysis to attack DPA resistant software. In: Paar, C., Koç, Ç.K. (eds.) CHES 2000. LNCS, vol. 1965, pp. 238–251. Springer, Heidelberg (2000)
5. Chari, S., Jutla, C., Rao, J., Rohatgi, P.: A cautionary note regarding evaluation of AES candidates on smart-cards. In: Proceedings of the 2nd Advanced Encryption Standard Candidate Conference, Rome, Italy, 22–23 March 1999
6. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)
7. Le, T.-H., Clédière, J., Canovas, C., Robisson, B., Servière, C., Lacoume, J.-L.: A proposition for correlation power analysis enhancement. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 174–186. Springer, Heidelberg (2006)
8. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 398–412. Springer, Heidelberg (1999)
9. Agrawal, D., Archabeault, B., Rao, J.R.: The EM side-channel: attacks and assessment methodologies. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 29–45. Springer, Heidelberg (2003)
10. Mangard, S.: Hardware countermeasures against DPA – a statistical analysis of their effectiveness. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 222–235. Springer, Heidelberg (2004)