Bitwise Collision Attack Based on Second-Order Distance

Danhui Wang[1], An Wang[2, 3]
[1] China Academy of Electronics and Information Technology, Beijing 100041, China
[e-mail: wangdanhui2014@163.com]
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Institute of Microelectronics, Tsinghua University, Beijing 100084, China
[e-mail: wanganl@sina.com]
*Corresponding author: An Wang

Received March 11, 2014; accepted February 1, 2017; published March 31, 2017

Abstract

Correlation-enhanced collision attack has been proposed by Moradi et al. for several years. However, in practical operations, this method costs lots of time on trace acquisition, storage and averaging due to its bytewise collision detection. In this paper, we propose a bitwise collision attack based on second-order distance model. In this method, only 9 average traces are enough to finish a collision attack. Furthermore, two candidate models are given in this study to distinguish collisions, and the corresponding practical experiments are also performed. The experimental results indicate that the operation time of our attack is only 8% of that of correlation-enhanced collision attack, when the two success rates are both above 0.9.

Keywords: Collision Attack, Bitwise Collision, Second-Order Distance, Power Analysis Attack, Advanced Encryption Standard

This research was supported by National Natural Science Foundation of China (Nos. 61402252, 61402536), Beijing Natural Science Foundation (No. 4162053), Foundation of Science and Technology on Information Assurance Laboratory (No. KJ-15-005), and Beijing Institute of Technology Research Fund Program for Young Scholars.

https://doi.org/10.3837/tiis.2017.03.030 **ISSN : 1976-7277**

1. Introduction

With the development of information technology, information systems, which support data-intensive applications on software and hardware, gradually become the lifeblood of a functioning state and the pillar of social activities. The security of information systems is increasingly valued, because any destruction or malfunction may cause huge impacts on the whole community. Cryptanalysis is an important research area of information science and technology. It mainly relies on mathematical methods over the years. However, the mathematical methods are only concerned with plaintexts and ciphertexts, which has great difficulty in key recovery.

Since timing attack which observes variations on performing time was proposed by Kocher in 1996 [1], the field of side-channel attacks and countermeasures has gradually become an important branch of cryptography. Side-channel attacks, which pay close attention to the intermediate values, are based on the leakage of information from the physical implementation of a cryptosystem, rather than the traditional methods such as brute force or theoretical weaknesses. Although the information of timing, electromagnetism, or even sound can be exploited to attack a cryptosystem, power consumption analysis is the most effective means of cryptanalysis. In practice, these techniques are typically implemented on cryptographic chips, such as microprocessor, FPGA, and ASIC [2]. One of the most popular countermeasures against power analysis attacks is masking, which usually defeats first-order side-channel attacks [3-7], because all the intermediate values are randomly masked in different implementations.

In Crypto 1999, Kocher et al. presented differential power analysis which can recover secret keys by analyzing the information of instantaneous power consumption of cryptographic chips [8, 9]. Template attack was introduced by Chari et al. in 2002. The attacker matches the recorded power traces with the power consumption characteristics which are called templates for different key hypotheses in a template attack [10]. In 2004, Brier et al. proposed correlation power analysis which recovers secret keys with the correlation coefficient model [11]. On the basis of probability theory and information theory, mutual information analysis [12] was given by Gierlichs et al.

Collision attack is an effective method of cryptanalysis so far. In the pioneering work of Schramm et al., a concept of internal collision was proposed [13]. And then the practicability of collision attack against DES was presented [13, 14]. Soon after, the collision technique was successfully used on AES [15]. In 2007, linear collision attack was proposed by Bogdanov [16, 17]. The relationships among several key bytes are established by the collisions between different S-boxes. In order to measure the distance between two traces for collision detection, the mathematical models such as least absolute deviation and least square method [18] are generally employed in collision attacks. However, these models can only detect the collisions of the hamming weights of intermediate values.

In CHES 2010, Moradi et al. proposed correlationenhanced collision attack on hardware implementation of AES [19]. And their attack was improved by Clavier et al. with the collision-correlation method in CHES 2011 [20]. However, their inefficiency is a serious problem. In practice, while numbers of power traces are needed to acquire 256 average traces (due to its bytewise operations) in correlation-enhanced collision attack, the attacker need to average the traces on an oscilloscope and store them manually, or automatically store all the traces and then handle them in MATLAB. But the process of trace acquisition, storage and averaging is complex and time-consuming.

Our Contribution. In this work we present a collision attack on the basis of bitwise comparison and the distance model between power traces. Specifically,

- A flexible framework of collision attack is proposed, which can distinguish the collisions by bit instead of byte.
- Two distinguishing models are given to carry out bitwise collision attack. According to the experiments, the efficiency of our attack is much higher than the previous method of Moradi et al. (Especially on the operation time which is 8% of theirs).

Organization. The remainder of this article is organized as follows. In Section 2, we briefly describe linear collision attack, hamming weight model and distance mathematical model, and then review the correlation-enhanced collision attack. Section 3 introduces our basic idea on hamming weight model, distance distinguisher and the framework of our attack. Then the experiments and efficiency comparison are shown in Section 4. A conclusion is given in Section 5.

2. Related Work

2.1 Notations

This paper focuses on AES-128. The variables are represented by the following notations. $K = \{k_i \mid i = 1,2,...,16\}$ is the 16-byte user-supplied key. $P^j = \{p_i^j \mid i = 1,2,...,16\}$ denotes an 16-byte AES plaintext, and $j = 1,2,...,N$ is the number of an AES execution.

After an encryption, the recorded power trace can be separated into 16 sections $\{T_i \mid i = 1,2,...,16\}$ corresponding to 16 S-boxes. In one attack we need to encrypt a plaintext several times to obtain numbers of traces, and then compute an average trace which also consists of 16 sections $\{\overline{T}_i \mid i = 1,2,...,16\}$. There are l interesting points $\{t_{i,s} \mid s = 1,2,...,l\}$ on each section. We use $\{x_i \mid i = 1,2,...,16\}$ to denote the input bytes of 16 S-boxes in the first round.

2. 2 Linear Collision Attack

AES-128 has 10 rounds of which each consists of four operations: AddRoundKey, SubBytes, ShiftRows and MixColumns. But the last MixColumns is replaced by AddRoundKey in the 10th round. AES-128 operates on a state which is a 4 × 4 matrix of bytes. A 128-bit plaintext is XORed with the 128-bit round subkey in AddRoundKey of an encryption. The 16 output bytes of AddRoundKey are the inputs of 16 parallel S-boxes in SubBytes.

Linear collision attack was proposed by Bogdanov et al. in 2007 [16]. If two S-boxes in one round accept the same inputs, an internal linear collision attack occurs. **Fig. 1** illustrates a detected collision in the first round. The S-box outputs

$$S(p_3 \oplus k_3) = S(p_{13} \oplus k_{13}),$$

then we obtain a linear relationship between key bytes k_3 and k_{13}:

$$k_3 \oplus k_{13} = p_3 \oplus p_{13} = \Delta_{3,13}.$$

Obviously $\Delta_{3,13}$ is a constant. If the attacker obtains some information about $\Delta_{3,13}$, one key byte may be recovered by searching another one.

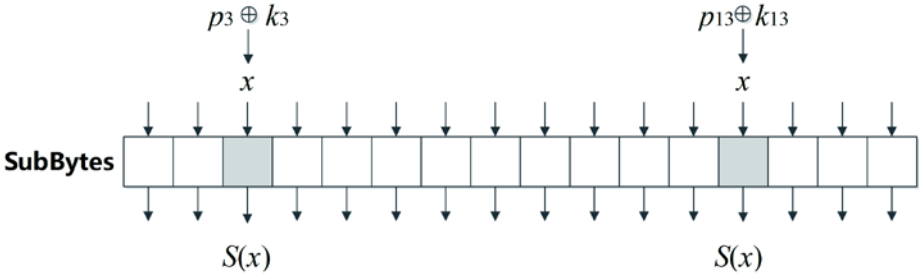

Fig. 1. A linear collision between two AES S-boxes

2.3 Hamming Weight Model and Collision Distinguisher

A linear relationship between the hamming weight and the power consumption per unit time can be derived from hamming weight model [21]. The expression of instantaneous power consumption with a reference point is

$$T = aHW(x) + b.$$

a and b are both constants, and $HW(x)$ denotes the hamming weight.

A plaintext including two bytes which correspond to S-box 1 and S-box 2 is chosen for example. We encrypt this plaintext several times to acquire power traces, and then average them into one trace. The two average sections $\overline{T}_1$ and $\overline{T}_2$ are extracted. The interesting points on these two sections are $\{t_{1,s} \mid s = 1,2,...,l\}$ and $\{t_{2,s} \mid s = 1,2,...,l\}$. Least

absolute deviation and least square method are generally used to measure the distance between two traces, and their expressions are as follows:

$$D_{LAD} = \overline{T}_2 - \overline{T}_1 = \sum_{s=1}^{l} | t_{2,s} - t_{1,s} |,$$

$$D_{LSM} = \sum_{s=1}^{l} (t_{2,s} - t_{1,s})^2 .$$

In addition, we use other distance models in our work, and the details are in Section 3. These methods can only be used to distinguish the collisions of $HW(x)$, rather than whether x_1 equals to x_2 for linear collisions.

2.4 Correlation-Enhanced Collision Attack

In 2010, Moradi et al. proposed correlation-enhanced collision attack [19] which can also be adopted as a method of collision detection. Their attack on the first two S-boxes is described in **Table 1**.

Table 1. Correlation-enhanced collision attack

Algorithm 1 Correlation-enhanced collision attack between S-box 1 and S-box 2

Online Stage:

1: $P = (P^1, P^2, ..., P^N) \leftarrow$ **RandomPlaintexts**()
2: $\{T_1^j \mid j = 1,2,...,N\} \leftarrow$ **AcquireTraces**(P)
3: $\{T_2^j \mid j = 1,2,...,N\} \leftarrow$ **AcquireTraces**(P)

Key-recovery Stage:

1: $\{\overline{T}_1^{\alpha} \mid \alpha = 0,1,...,255\} \leftarrow$ **AverageTraces**($\{T_1^j\}$)
2: $\{\overline{T}_2^{\alpha} \mid \alpha = 0,1,...,255\} \leftarrow$ **AverageTraces**($\{T_2^j\}$)
3: **for** each guess $\Delta \in \{0,1,...,255\}$ **do**
4: $\rho_{\Delta} \leftarrow$ **CorrelationCoefficient**($\{\overline{T}_1^{p_1}\}, \{\overline{T}_2^{p_1 \oplus \Delta}\}$)
5: **end for**

return $\arg\max_{\Delta} \rho_{\Delta}$

In the online stage, 128-bit plaintexts $P^1, P^2, ..., P^N$ are randomly chosen (**RandomPlaintexts**). Then two sections of power traces $\{T_1^j \mid j = 1,2,...,N\}$ and $\{T_2^j \mid j = 1,2,...,N\}$ for AES S-box 1 and S-box 2 are recorded (**AcquireTraces**).

In the key-recovery stage, $\{T_1^j \mid j = 1,2,...,N\}$ are separated into 256 groups of which each corresponds to a certain value of plaintext byte. Then the traces are averaged by group, and finally 256 average traces are acquired (**AverageTraces**). After performing the same

operations on $\{T_2^j \mid j = 1,2,...,N\}$, the attacker extracts two sets of average traces, i.e. $\{\overline{T}_1^\alpha \mid \alpha = 0,1,...,255\}$ and $\{\overline{T}_2^\alpha \mid \alpha = 0,1,...,255\}$ for two S-boxes. According to the equation

$$p_2 = p_1 \oplus \Delta,$$

the average traces $\{\overline{T}_2^\alpha \mid \alpha = 0,1,...,255\}$ are reordered for each of the 256 guessed Δ. The attacker computes the 256 correlation coefficients ρ_Δ of $\{\overline{T}_1^{p_1} \mid p_1 = 0,1,...,255\}$ and $\{\overline{T}_2^{p_1 \oplus \Delta} \mid p_1 = 0,1,...,255\}$, and at last chooses the maximum. The correct Δ is given by

$$\arg\max_\Delta \rho_\Delta.$$

Problem in practice. Plenty of power traces are acquired on the Agilent MSO-X 3054A oscilloscope to obtain 256 average traces. The data are stored from a low performance oscilloscope to a computer via cable, which costs much time. Thus the attacker may spend time on the following steps:

- Online automatic averaging. Original traces are averaged on an oscilloscope, which takes 20ms at a time. For example, averaging 10 traces costs 0.2s.
- Online manual storage. The traces are stored manually, and 5s is taken for one trace.
- Online automatic storage. The traces are automatically stored in computer without averaging. This automatic operation on one trace takes 0.6s.

Suppose that 10 original traces are averaged to acquire one average trace ($N = 10 \times 256 = 2560$), there are two implementations in practice:

- Traces are averaged on an oscilloscope. The average trace will be stored manually after averaging 10 traces. This process of one average trace costs 5.2 seconds. Then the total time for 256 average traces takes about 23 minutes.
- 2560 original traces are stored from an oscilloscope to a computer automatically. Then they are averaged in MATLAB. The storage process costs about 26 minutes.

An attack is expected to be fast and efficient as far as possible. In the next section, we propose an attack which only 9 average traces.

3. Second-Order Distance Collision Attack

3.1 Basic Idea

We choose 9 special plaintexts and operate on the bytes for S-box 1 and S-box 2 as an example. The bytes of one plaintext $P^\alpha (\alpha = 1,2,...,8)$ are equal to the same value p^α whose α th bit is 1 and others are 0. In other words, the bytes $\{p^\alpha \mid \alpha = 1,2,...,8\}$ are

0000 0001; 0000 0010; 0000 0100; 0000 1000;
0001 0000; 0010 0000; 0100 0000; 1000 0000.

Especially P^0 is an all-zero plaintext with $p^0 = 00000000$. We use $\{x_i^\alpha \mid \alpha = 0,1,...,8\}$ to denote the inputs of the i th S-box. Let the first two key bytes

$$k_1 = u_8 \| u_7 \| u_6 \| u_5 \| u_4 \| u_3 \| u_2 \| u_1,$$
$$k_2 = v_8 \| v_7 \| v_6 \| v_5 \| v_4 \| v_3 \| v_2 \| v_1.$$

Then we denote

$$\Delta = k_1 \oplus k_2.$$

When we choose two plaintexts with p^0 and p^1, the changes of $HW(x_1)$ and $HW(x_2)$ are relevant to the first bit of k_1 and k_2 (i.e. u_1 and v_1). This information can be used to detect whether $u_1 = v_1$. A distinguisher between collision and non-collision is given as follows:

1. When $p_1 = p_2 = p^0 = 00000000$,

$$x_1^0 = p^0 \oplus k_1 = u_8 \| u_7 \| u_6 \| u_5 \| u_4 \| u_3 \| u_2 \| u_1,$$
$$x_2^0 = p^0 \oplus k_2 = v_8 \| v_7 \| v_6 \| v_5 \| v_4 \| v_3 \| v_2 \| v_1.$$

We use $\Delta HW^\alpha (\alpha = 0,1,...,8)$ to express the difference between $HW(x_1^\alpha)$ and $HW(x_2^\alpha)$, then

$$\Delta HW^0 = HW(x_1^0) - HW(x_2^0). \tag{1}$$

We denote

$$h = HW(u_8 \| u_7 \| u_6 \| u_5 \| u_4 \| u_3 \| u_2) - HW(v_8 \| v_7 \| v_6 \| v_5 \| v_4 \| v_3 \| v_2).$$

Hence

$$\Delta HW^0 = h + (HW(u_1) - HW(v_1)).$$

2. When $p_1 = p_2 = p^1 = 00000001$,

$$x_1^1 = p^1 \oplus k_1 = u_8 \| u_7 \| u_6 \| u_5 \| u_4 \| u_3 \| u_2 \| (u_1 \oplus 1),$$
$$x_2^1 = p^1 \oplus k_2 = v_8 \| v_7 \| v_6 \| v_5 \| v_4 \| v_3 \| v_2 \| (v_1 \oplus 1).$$

Then

$$\Delta HW^1 = HW(x_1^1) - HW(x_2^1) = h + (HW(u_1 \oplus 1) - HW(v_1 \oplus 1)). \tag{2}$$

According to these two cases, we can draw the following conclusions:

- If and only if $u_1 = v_1$,

$$|\Delta HW^0 - \Delta HW^1| = 0. \tag{3}$$

- If and only if $u_1 \neq v_1$,

$$|\Delta HW^0 - \Delta HW^1| = 2. \tag{4}$$

With this method, the collisions on the other bit positions can be detected (choose p^0 and p^α for the α th bit).

In order to construct the templates for collision distinguisher in practice, we hope that Formulae (3) and (4) can be approximated by traces (see **Fig. 2**).

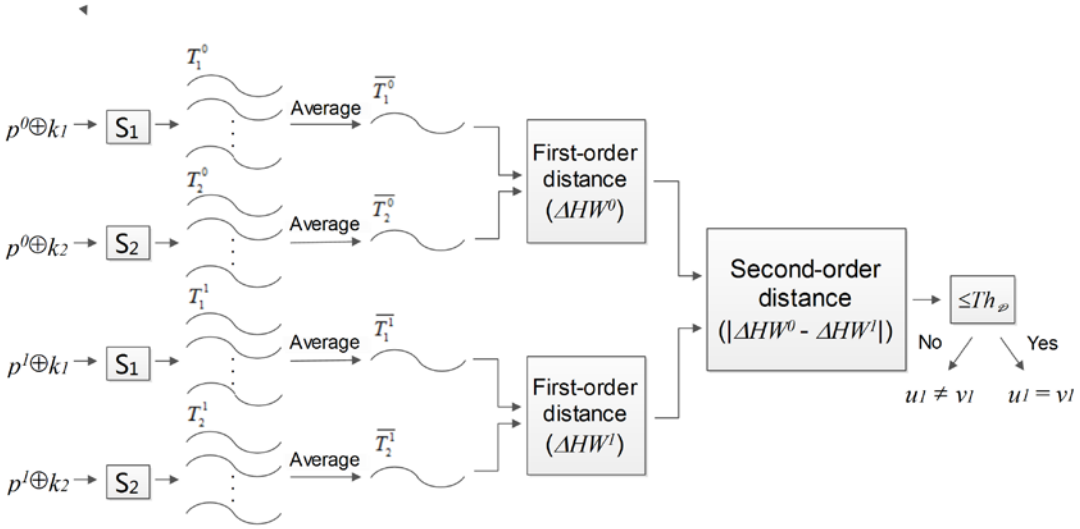

Fig. 2. Flow chart of bitwise collision attack on u_1 and v_1

3.2 Second-Order Distance Distinguisher

3.2.1 First-order distance

In Section 3.1, the hamming weights reflect the traces with none noise. Otherwise, the attacker can average original traces for noise reduction. Then the average traces approximates to the hamming weights.

The section of average trace with plaintext byte p^0 for the i th S-box is denoted by $\overline{T}_i^0$, and p^1 corresponds to $\overline{T}_i^1$. Each section contains l interesting points. According to the formulae of ΔHW^0 and ΔHW^1 (Formulae (1) and (2)), the first-order distances are given by the following expressions:

$$D^0 = \overline{T}_2^0 - \overline{T}_1^0 = \sum_{s=1}^{l} (t_{2,s}^0 - t_{1,s}^0),$$

$$D^1 = \overline{T}_2^1 - \overline{T}_1^1 = \sum_{s=1}^{l} (t_{2,s}^1 - t_{1,s}^1).$$

D^α represents the distance relation between $\overline{T}_1^\alpha$ and $\overline{T}_2^\alpha$ when the value of plaintext byte is p^α, and it reflects the difference between $HW(x_1^\alpha)$ and $HW(x_2^\alpha)$.

3.2.2 Second-order distance

We define a second-order distance between two firstorder distances to judge the change of ΔHW^0 and ΔHW^1. The second-order distance is expressed as follow:

$$\mathcal{D} = | D^0 - D^1 | . \tag{5}$$

A threshold value $Th_{\mathcal{D}}$ should be precomputed to measure this difference. On cases of $u_1 = v_1$ and $u_1 \neq v_1$, we give two average values of $| D^0 - D^1 |$ respectively on statistics. $Th_{\mathcal{D}}$ is a half of the difference between the two average values. Then we compare $\mathcal{D}$ with $Th_{\mathcal{D}}$ to judge which value that $\mathcal{D}$ is close to.

The following conclusions are given:

- If $\mathcal{D} \leq Th_{\mathcal{D}}$, we conclude that

$$u_1 = v_1 .$$

- If $\mathcal{D} > Th_{\mathcal{D}}$, we conclude that

$$u_1 \neq v_1 .$$

According to the hamming weight model distinguisher, the difference of two first-order distances reflects the change of ΔHW^0 and ΔHW^1. Then Formulae (3) and (4) can be interpreted as follows:

- If and only if $u_1 = v_1$, the difference between D^0 and D^1 is approaching to 0.
- If and only if $u_1 \neq v_1$, the difference between D^0 and D^1 is a distinct value.

This method is used on the other bits with the same $Th_{\mathcal{D}}$. Finally the Δ between k_1 and k_2 can be derived, so do other key bytes.

3.3 Framework of Second-Order Distance Collision Attack

The process of our attack on u1 and v1 is illustrated by the flow chart (**Fig. 2**), and this attack on all the bit positions for the first two S-boxes is described in **Table 2**.

In the online stage, we choose an all-zero plaintext and 8 special plaintexts of which only one bit is 1 in different bit positions as mentioned in Section 3.1 (**ChoosePlaintexts**). Then we encrypt these plaintexts n times respectively, and record the power traces. The total number of recorded traces is $N = n \times 9$. The two sections of traces corresponding to Sbox 1 and S-box 2 are extracted (**AcquireTraces**). Subsequently, the n traces for a certain plaintext are averaged (**AverageTraces**), at last we obtain average traces $\{\overline{T}_1^{\alpha} \mid \alpha = 0,1,...,8\}$ and $\{\overline{T}_2^{\alpha} \mid \alpha = 0,1,...,8\}$.

In the key-recovery stage, we firstly compute the first-order distance D^0 between two average traces with all-zero plaintext. Then with the other plaintexts, we compute the distance $D^{\alpha}(\alpha = 0,1,...,8)$ between two corresponding average traces (**First-OrderDistance**). The difference between D^{α} and D^0 (**Second-OrderDistance**) is

compared with a threshold value $Th_{\wp}$. The result determines whether $u^{\alpha} = v^{\alpha} (\alpha = 0,1,...,8)$. After the comparisons on 8 bit positions, $\Delta = k_1 \oplus k_2$ will be recovered.

Table 2. Bitwise Collision Attack

Algorithm 2 Bitwise Collision Attack on S-box 1 and S-box 2

Online Stage:

1: **for** $\alpha = 0,1,...,8$ **do**
2: $\{T_1^j \mid j = 1,2,...,n\} \leftarrow$ **AcquireTraces**(P)
3: $\{T_2^j \mid j = 1,2,...,n\} \leftarrow$ **AcquireTraces**(P)
4: $\overline{T}_1^{\alpha} \leftarrow$ **AverageTraces**($\{T_1^j\}$)
5: $\overline{T}_2^{\alpha} \leftarrow$ **AverageTraces**($\{T_2^j\}$)
6: **end for**

Key-recovery Stage:

1: $D^0 = \overline{T}_2^0 - \overline{T}_1^0$
2: **for** $\alpha = 1,2,...,8$ **do**
3: $D^{\alpha} = \overline{T}_2^{\alpha} - \overline{T}_1^{\alpha}$
4: $\wp = \mid D^0 - D^{\alpha} \mid$
5: **If** $\wp \le Th_{\wp}$
6: $\delta_{\alpha} = 0 \quad (u_{\alpha} = v_{\alpha})$
7: **else**
8: $\delta_{\alpha} = 1 \quad (u_{\alpha} \neq v_{\alpha})$
9: **end if**
10: **end for**
11: **return** $\Delta = \delta_8 \| \delta_7 \| \delta_6 \| \delta_5 \| \delta_4 \| \delta_3 \| \delta_2 \| \delta_1$

3.4 The Choice of First-Order Distance Function

In this section we introduce another workable choice of the first-order distance, and then discuss the feasibilities of other candidates such as the least absolute deviation and least square method. The experimental data are given in Section 4.2.

3.4.1 Another distance model

The hamming weight model of first-order distance (see Formulae (1) and (2)) can be replaced by

$$\Delta HW^0 = HW(x_1^0) + HW(x_2^0),$$
$$\Delta HW^1 = HW(x_1^1) - HW(x_2^1).$$

Suppose that

$$h = HW(u_8 \| u_7 \| u_6 \| u_5 \| u_4 \| u_3 \| u_2) + HW(v_8 \| v_7 \| v_6 \| v_5 \| v_4 \| v_3 \| v_2),$$

then

$$\Delta HW^0 = h + (HW(u_1) + HW(v_1)),$$

$$\Delta HW^1 = h + (HW(u_1 \oplus 1) + HW(v_1 \oplus 1)).$$

It would be easy to deduce the following conclusions:

- If and only if $u_1 = v_1$,

$$|\Delta HW^0 - \Delta HW^1| = 2.$$

- If and only if $u_1 \neq v_1$,

$$|\Delta HW^0 - \Delta HW^1| = 0.$$

These conclusions are exactly contrary to Formulae (3) and (4) in Section 3.1.

Thus the first-order distances of traces are actually computed as follows:

$$D^0 = \overline{T}_2^0 + \overline{T}_1^0 = \sum_{s=1}^{l} (t_{2,s}^0 + t_{1,s}^0),$$

$$D^1 = \overline{T}_2^1 + \overline{T}_1^1 = \sum_{s=1}^{l} (t_{2,s}^1 + t_{1,s}^1).$$

The second-order distance is the same to Formula (5) but measured by the following conclusions:

- If $\mathscr{D} > Th_{\mathscr{D}}$, we conclude that

$$u_1 = v_1.$$

- If $\mathscr{D} \leq Th_{\mathscr{D}}$, we conclude that

$$u_1 \neq v_1.$$

3.4.2 Other candidates of distance function

Generally, the least absolute deviation and least square method are used to measure the distance between two traces, but not in our attack.

If the least absolute deviation is used for the first-order distance, the hamming weight expressions (Formulae (1) and (2)) described in Section 3.1 turn into

$$\Delta HW^0 = |HW(x_1^0) - HW(x_2^0)| = |h + (HW(u_1) - HW(v_1))|,$$

$$\Delta HW^1 = |HW(x_1^1) - HW(x_2^1)| = |h + (HW(u_1 \oplus 1) - HW(v_1 \oplus 1))|.$$

We take notice of the case when $h = 0$. Even though $u_1 \neq v_1$, ΔHW^0 always equals to ΔHW^1. Then $|\Delta HW^0 - \Delta HW^1| = 0$ will cause a misjudgment. This kind of errors occurs with non-negligible probability.

The hamming weight expressions with the least square method are

$$\Delta HW^0 = (HW(x_1^0) - HW(x_2^0))^2,$$
$$\Delta HW^1 = (HW(x_1^1) - HW(x_2^1))^2.$$

Obviously if $h = 0$, The same kind of errors will occur.

As a result, these two methods are not effective for our attack.

4. Experiments and Efficiency

4.1 Experiment in Practice

Our attack has been implemented on an AT89S52 singlechip for practical experiment, and the original traces are acquired from the Agilent MSO-X 3054A oscilloscope. We average 1000 traces for one average trace, and each trace has 1000 sampling points. Without loss of generality, two key bytes for S-box 1 and S-box 2 are fixed as

$$k_1 = 01100110,$$
$$k_2 = 10101010.$$

In this experiment, $\overline{T}^0$ and $\overline{T}^\alpha$ are generated to approximate ΔHW^0 and $\Delta HW^\alpha (\alpha = 0,1,...,8)$. The distance between two traces is related to the αth bits of k_1 and k_2.

After implementing on 8 bit positions, the differences of corresponding average traces are shown in **Fig. 3** from (a) to (h). In each figure the light trace express $\Delta\overline{T}^0$, and the dark trace is $\Delta\overline{T}^\alpha$ on the αth bit.

For example, the case $u_1 = v_1$ is shown in **Fig. 3(a)**. It illustrates the two traces are overlapping throughout. And in **Fig. 3(c)**, another example shows that the two traces have significant gap from point 330 to 900 when $u_3 \neq v_3$.

At last, **Fig. 4** is plotted to show the secondorder distances on 8 bit positions which are described by the points. The threshold value $Th_\varnothing$ is expressed by the horizontal line. It is obvious that the second-order distances of the four equal bit positions are under the threshold line. And the unequal points are much higher than the threshold line if the traces are averaged enough times.

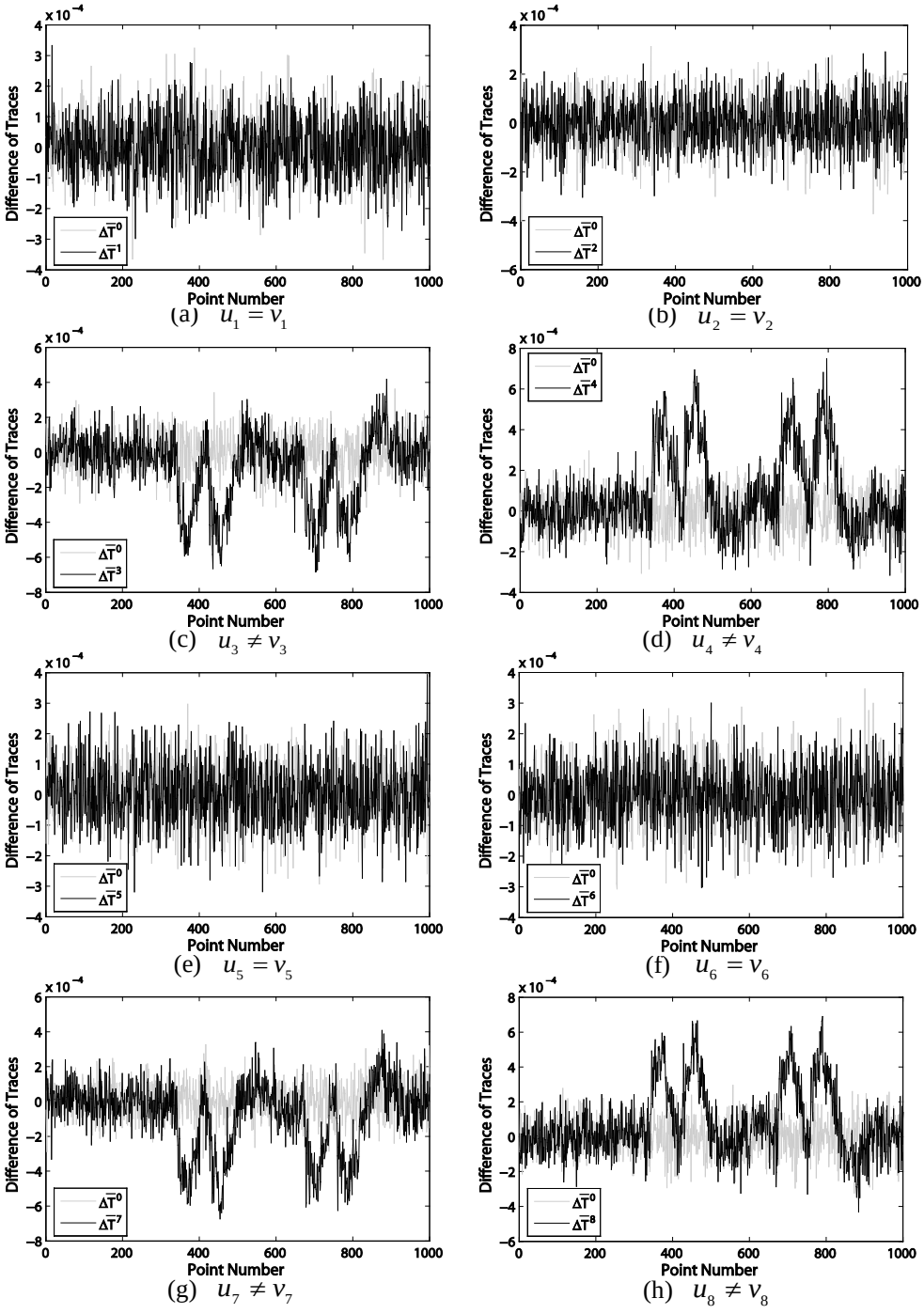

Fig. 3. The trace approximations of the fist-order distances on 8 bit positions

4.2 Efficiency Comparison

In common with the result of our attack, the XORed values of key bytes can also be recovered by correlationenhanced collision attack (CECA). We made some simulations in MATLAB to evaluate the efficiency of our attack. In addition, except the distance model used in **Table 1** (named Model 1) and the second model (named Model 2) described in Section 3.4.1, we also verify the feasibilities of least absolute deviation (LAD) and least square method (LSM) for the firstorder distance function.

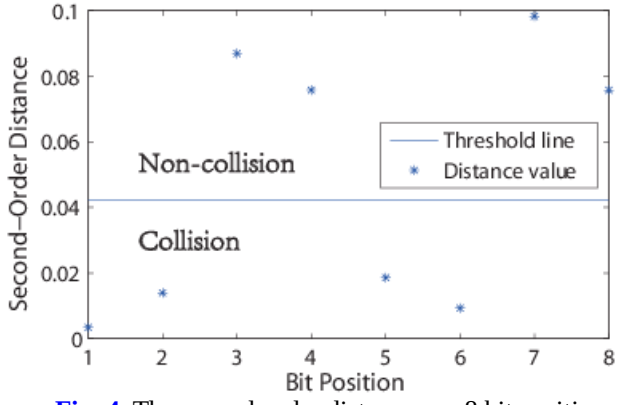

Fig. 4. The second-order distances on 8 bit positions

Success rates on trace number. In this experiment, the trace number is chosen from 256 to 2048. The standard deviation of noise $\sigma = 5$, and the number of interesting points is fixed 5. Each attack was repeated 1000 times to compute success rates with a certain number of traces. Finally the relations between success rates and trace number are depicted in **Fig. 5**.

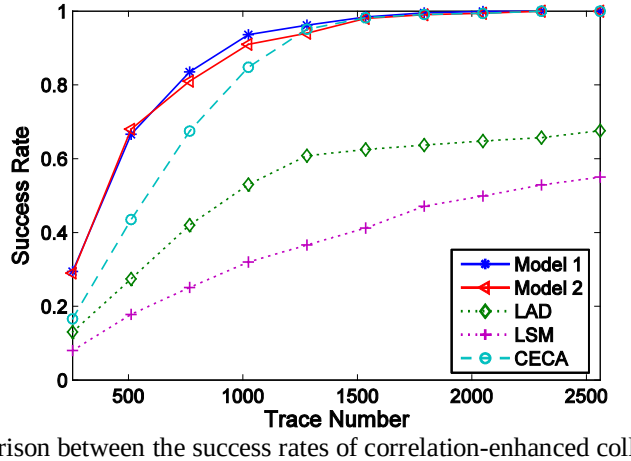

Fig. 5. The comparison between the success rates of correlation-enhanced collision attack and our attack with different distance functions

The two success rates of our attack with Model 1 and Model 2 are almost the same. It is obvious that the success rate of our attack is higher than that of correlation-enhanced collision attack, but the curves almost overlap to each other when the trace number is more than 1280. This experiment demonstrates that our attack has significant advantages while few traces are recorded. For example, when the two success rates are both above 0.9, our attack needs around 900 traces, which is only 70% of the trace number of correlationenhanced collision attack.

Success rates on operation time. Our attack has more advantage on operation time as illustrated in **Fig. 6**. Model 1 is used to make comparison with correlation-enhanced collision attack. Suppose that the time calculation is the same with that described in Section 2.4, there are two methods for carrying out the attacks:

1. The traces are averaged on an oscilloscope. Subsequently, the average traces are manually stored. In this case, handling 9 average traces of our attack costs 45s (Our attack 1), but the correlation-enhanced collision attack needs 1280s at least (CECA 1). Therefore, our operation time is about 3.5% of that of CECA 1.
2. All the original traces are automatically stored, and then averaged in MATLAB. The shapes of two curves (Our attack 2 and CECA 2) are the same with those in Figure 4. Thus our operation time is 70% of CECA 2 in this case, when the two success rates are above 0.9.

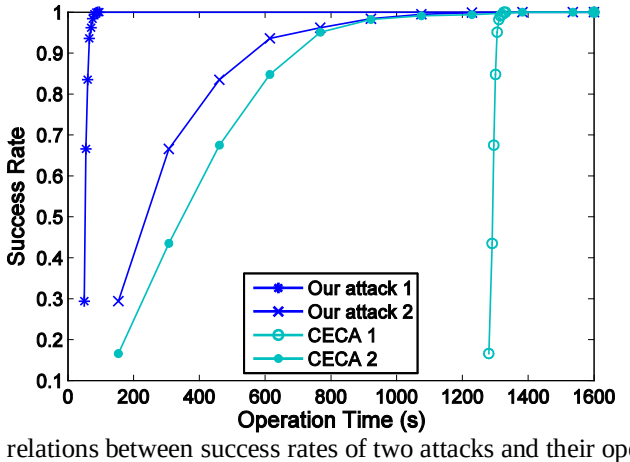

Fig. 6. The relations between success rates of two attacks and their operation time

It is easy to plot the relations between success rates and operation time in accordance with the previous experimental data. If both the two attacks are carried out with the more efficient method (Our attack 1 and CECA 2), and the success rates are expected above 0.9, the operation time of our attack is about 65s and the other is 768s. Therefore, our operation time is about 8% of that of CECA 2. As a result, no matter which method is adopted, our

attack takes far less time than correlation-enhanced collision attack at the same success rates.

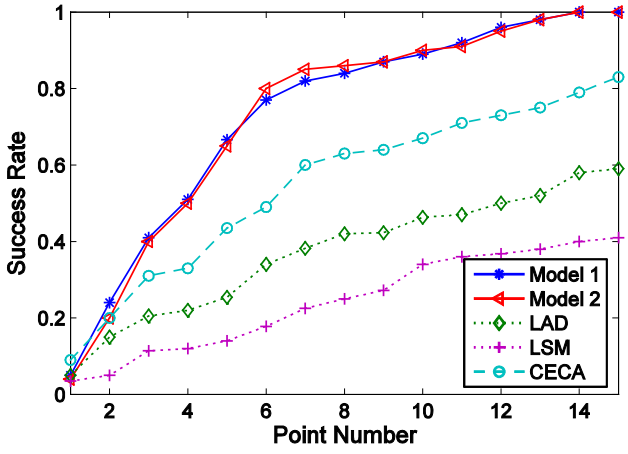

Fig. 7. The relations between success rates and point number

Success rates on point number. Then we change the number of interesting points to make another experiment (see **Fig. 7**). If $\sigma = 5$ and the trace number is fixed 512, we observe that the point number affects the success rate greatly. **Fig. 7** shows the relative curves when point number is increasing from 1 to 15. With the same point numbers, our attack is always more efficient than correlation-enhanced collision attack.

Fig. 5 and **Fig. 7** also illustrate the success rates of least absolute deviation and least square method for the first-order distance. Because of the non-negligible errors described in Section 3.4.2, these two curves are lower than the others.

5. Conclusions

In this work we have presented a collision attack which uses second-order distance to detect collisions by bit to recover key, and two appropriate distance models are given. Since the target of this attack is the same with correlation-enhanced collision attack, we make a comparison on the efficiencies. The experiments are not only implemented in simulation but also on an AT89S52 singlechip in practice. We also discuss the possibilities of the choice of first-order distance function, and give the experimental data on the hypotheses. These models in our work can be extended to the cryptographic symmetric algorithms which are vulnerable to the collision attacks.

References

[1] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *Proc. of CRYPTO'96*, LNCS 1109, pp. 104-113, August 18-22, 1996. Article (CrossRef Link).

[2] N. Ferguson, B. Schneier and T. Kohno, *Cryptography engineering: design principles and practical applications*, Wiley, Hoboken, 2010. Article (CrossRef Link).

[3] B. Bilgin, S. Nikova, V. Nikov, V. Rijmen and G. Stutz, "Threshold Implementations of All 33 and 44 SBoxes," in *Proc. of CHES 2012*, LNCS 7428, pp. 76-91, September 9-12, 2012. Article (CrossRef Link).

[4] D. Canright and L. Batina, "A Very Compact "Perfectly Masked" S-Box for AES," in *Proc. of ACNS 2008*, LNCS 5037, pp. 446-459, June 3-6, 2008. Article (CrossRef Link).

[5] C. Carlet, L. Goubin, E. Prouff, M. Quisquater and M. Rivain, "Higher-Order Masking Schemes for SBoxes," in *Proc. of FSE 2012*, LNCS 7549, pp. 366-384, March 19-21, 2012. Article (CrossRef Link).

[6] L. Genelle, E. Prouff and M. Quisquater, "Thwarting Higher-Order Side Channel Analysis with Additive and Multiplicative Maskings," in *Proc. of CHES 2011*, LNCS 6917, pp. 240-255, September 28-October 1, 2011. Article (CrossRef Link).

[7] Roy and S. Vivek, "Analysis and Improvement of the Generic Higher-Order Masking Scheme of FSE 2012," in *Proc. of CHES 2013*, LNCS 8086, pp. 417-434, August 20-23, 2013. Article (CrossRef Link).

[8] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis," in *Proc. of Crypto 1999*, LNCS 1666, pp. 388-397, August 15-19, 1999. Article (CrossRef Link).

[9] S. Mangard, E. Oswald and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer, Heidelberg, 2007. Article (CrossRef Link).

[10] S. Chari, J. R. Rao and P. Rohatgi, "Template Attacks," in *Proc. of CHES 2002*, LNCS 2523, pp. 13-28, August 13-15, 2003. Article (CrossRef Link).

[11] E. Brier, C. Clavier and F. Olivier, "Correlation Power Analysis with a Leakage Model," in *Proc. of CHES 2004*, LNCS 3156, pp. 16-29, August 11-13, 2004. Article (CrossRef Link).

[12] B. Gierlichs, L. Batina, P. Tuyls and B. Preneel, "Mutual Information Analysis," in *Proc. of CHES 2008*, LNCS 5154, pp. 426-442, August 10-13, 2008. Article (CrossRef Link).

[13] K. Schramm, T. Wollinger and C. Paar, "A New Class of Collision Attacks and Its Application to DES," in *Proc. of FSE 2003*, LNCS 2887, pp. 206-222, February 24-26, 2003. Article (CrossRef Link).

[14] H. Ledig, F. Muller and F. Valette, "Enhancing Collision Attacks," in *Proc. of CHES 2004*, LNCS 3156, pp. 176-190, August 11-13, 2004. Article (CrossRef Link).

[15] K. Schramm, G. Leander, P. Felke and C. Parr, "A Collision-Attack on AES Combining Side Channeland Differential- Attack," in *Proc. of CHES 2004*, LNCS 3156, pp. 163-175, August 11-13, 2004. Article (CrossRef Link).

[16] Bogdanov, "Improved side-channel collision attacks on AES," in *Proc. of SAC07*, LNCS 4876, pp. 84-95, August 16-17, 2007. Article (CrossRef Link).

[17] Bogdanov, "Multiple-Differential Side-Channel Collision Attacks on AES," in *Proc. of CHES 2008*, LNCS 5154, pp. 30-44, August 10-13, 2008. Article (CrossRef Link).

[18] A. Sveshnikov and R. A. Silverman, *Problems in probability theory, mathematical statistics and theory of random functions*, Dover Publications, New York, 1979.
Article (CrossRef Link).

[19] Moradi, O. Mischke and T. Eisenbarth, "Correlation-enhanced power analysis collision attack," in *Proc. of CHES 2010*, LNCS 6225, pp. 125-139, August 17-20, 2010.
Article (CrossRef Link).

[20] C. Clavier, B. Feix, G. Gagnerot, M. Roussellet and V. Verneuil, "Improved Collision-Correlation Power Analysis on First Order Protected AES," in *Proc. of CHES 2011*, LNCS 6917, pp. 49-62, September 28-October 1, 2011. Article (CrossRef Link).

[21] M.-L. Akkar, R. Bevan and P. Dischamp, "Power analysis, what is now possible..." in *Proc. of ASIACRYPT 2000*, LNCS 1976, pp. 489-502, December 3-7, 2000. Article (CrossRef Link).

Danhui Wang was born in 1984. She received her Ph.D. degree in Shangdong University in 2014. She currently works in China Academy of Electronics and Information Technology. Her main research interests include side-channel analysis, internet of things, and cloud computing security.

An Wang was born in 1983. He received his Ph.D. degree in Shangdong University in 2011. He currently works in Beijing Institute of Technology. His main research interests include side-channel analysis, embedded system, and cryptographic implementation.