Hindawi
Security and Communication Networks
Volume 2018, Article ID 2483619, 13 pages
https://doi.org/10.1155/2018/2483619

Research Article

A Novel Multiple-Bits Collision Attack Based on Double Detection with Error-Tolerant Mechanism

Ye Yuan[1,2] **Liji Wu**[1,2] **Yijun Yang,**[1,2] **and Xiangmin Zhang**[1,2]

[1]*Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 10084, China*
[2]*Institute of Microelectronics, Tsinghua University, Beijing 10084, China*

Correspondence should be addressed to Liji Wu; lijiwu@tsinghua.edu.cn

Received 3 November 2017; Revised 6 April 2018; Accepted 3 May 2018; Published 5 June 2018

Academic Editor: Umar M. Khokhar

Copyright © 2018 Ye Yuan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Side-channel collision attacks are more powerful than traditional side-channel attack without knowing the leakage model or establishing the model. Most attack strategies proposed previously need quantities of power traces with high computational complexity and are sensitive to mistakes, which restricts the attack efficiency seriously. In this paper, we propose a multiple-bits side-channel collision attack based on double distance voting detection (DDVD) and also an improved version, involving the error-tolerant mechanism, which can find all 120 relations among 16 key bytes when applied to AES (Advanced Encryption Standard) algorithm. In addition, we compare our collision detection method called DDVD with the Euclidean distance and the correlation-enhanced collision method under different intensity of noise, which indicates that our detection technique performs better in the circumstances of noise. Furthermore, 4-bit model of our collision detection method is proven to be optimal in theory and in practice. Meanwhile the corresponding practical attack experiments are also performed on a hardware implementation of AES-128 on FPGA board successfully. Results show that our strategy needs less computation time but more traces than LDPC method and the online time for our strategy is about 90% less than CECA and 96% less than BCA with 90% success rate.

1. Introduction

Although modern cryptographic algorithms have been proven to be safe mathematically, this does not mean that the physical implementation is safe enough, where attacker can obtain some physical information from side channel. Side-channel attack (SCA) was proposed almost 20 years ago, which was first put forward in 1996 by Kocher [1] and became a powerful cryptanalysis technique. Power consumption analyses are widely used in SCA, which utilizes the relation between power consumption or electromagnetic signal of the executing device and processed data in order to recover the key value. Since Differential Power Analysis (DPA) was proposed in 1997 [2], whose distinguisher is the difference of the mean traces, various distinguishers have been designed and improved to enhance attack ability and efficiency, for example, Pearson correlation coefficient as a distinguisher for Correlation Power Analysis (CPA)[3], mutual information for Mutual Information Analysis (MIA)[4], and maximum likelihood for Template Attack [5, 6] (TA) and Template Based DPA [7]. However, the necessity of estimating and establishing the leakage model has been a serious restriction for SCA, which collision attack can ignore. Collision attack was first proposed to analyze Hash algorithm [8] and has become a branch of mathematical cryptanalysis, but it only reveals relation between input and output without exploiting internal information as SCA.

As a combination of SCA and collision attack, side-channel collision attack can exploit the information of internal leakage without a large number of power traces as well as the knowledge of the leakage model. Side-channel collision attack showed strong ability of attack, when first presented [9] against Data Encryption Standard (DES) by Schramm et al., which was applied to AES [10] successfully later. Then all kinds of improved versions [11–17] of side-channel collision attack sprang up, and most of these methods show high sensitivity to errors, where the recovered key is totally wrong even when error occurs only in 1 bit under the high noise level circumstance, leading to a low efficiency. Bogdanov presented some voting detection methods that seemed to be more practical [14], but they need too many traces in a profiling phase and encrypting the same plaintexts repeatedly

for decreasing the influence of noise may not be realistic. In 2010, Moradi proposed a correlation-enhanced method [15] that improves the probability of collision, but it may need lots of average power traces to process an attack and is sensitive to errors. In 2011, Bogdanov proposed an attack strategy [17] that uses the results of DPA to test chain separately. This method can improve the success probability in a sense that it cannot check the mistakes in collision detection which highly impact the attack results. Then Gérard et al. combined Low Density Parity Check (LDPC) decoding with correlation-enhanced and Euclidean Distance detection method in 2012 [16], which can be a globally efficient attack strategy in noisy settings. Two side-channel collision attack procedures based on bitwise collision detection were proposed, respectively, by Ren et al [18] in 2015 and by Wang et al [19] in 2017, which may have a poor performance on the detection success rate with high level noise. However, efficiency of collision detection and lack of error-tolerant and check mechanism are two main issues of existing side-channel collision attack.

Our Contribution. In this paper, we propose a novel multiple-bits collision attack framework. In particular, double distance voting detection (DDVD) and the error-tolerant and check mechanism are presented to ensure the high accuracy. In addition, we compare our collision detection method called DDVD with the Euclidean Distance and the correlation-enhanced collision methods under different intensity of noise, which indicates that our detection technique has a better performance in the circumstances of noise. Furthermore, 4-bit collision attack is proven to be optimal in theory and experiments. Practical attack experiments are performed successfully on a hardware implementation of AES in FPGA board.

The remainder of this paper is organized as follows. In Section 2, for a better understanding, we introduce some notations of our method as well as the basic linear collision attacks and then review the binary and ternary voting detection methods, correlation-enhanced collision attack, and LDPC decoding method in collision attack. In Section 3, a novel framework of multiple-bits collision attack is presented and we take the 4-bit model as an example to explain the attack procedure. In Section 4, we propose an improved version with an error-tolerant and check mechanism. In Section 5, we compare our collision detection method with other widely used detection techniques under different intensity of noise and analyze our model, and the experiments as well as the comparisons are also shown. Finally, we give the conclusion in Section 6.

2. Preliminaries

In order to understand the strategy easily, AES is chosen as the target block cipher to perform the attack method. As for the hardware implementation of this paper, it operates each of 16 S-boxes, which are used for the SubBytes operation, sequentially one by one. The following proposed statements and techniques can be successfully utilized in other cryptographic symmetric algorithms.

2.1. Notations. For a better description of the proposed method, we define some notations as follows. First we use letters k and p for 16-byte plaintext and first round subkey, with subscripts indicating a particular byte:

$$P \triangleq \{p_1, p_2, p_3, \cdots, p_{16}\},$$
$$K \triangleq \{k_1, k_2, k_3, \cdots, k_{16}\}. \tag{1}$$

Then we use the superscripts letters m and l for the 4 most significant bits and 4 least significant bits separately, meaning that $p_i^m \triangleq p_i[7{:}\,4]$, $k_i^m \triangleq k_i[7{:}\,4]$, $p_i^l \triangleq p_i[3{:}\,0]$, $k_i^l \triangleq k_i[3{:}\,0]$. Next, the attacker is able to choose the value of plaintext with key value all the same. The superscripts m_f and l_g state that the 4 most significant bits and 4 least significant bits are equal to values f and g in decimal format:

$$p_i^{m\_f} \triangleq \{p_i^m = f\},$$
$$p_i^{l\_g} \triangleq \{p_i^l = g\}. \tag{2}$$

Each trace acquired corresponding to first-round encryption contains 16 subtraces due to 16 sequential S-boxes, with subscripts indicating a particular S-box and each subtrace contains a number p of points, which are denoted by the subscripts:

$$T \triangleq \{T_1, T_2, T_3, \cdots, T_{16}\},$$
$$T_a \triangleq \{t_{a,1}, t_{a,2}, t_{a,3}, \cdots t_{a,p}\}. \tag{3}$$

Furthermore, we use T^{m_f} (T^{l_g}) to denote the power trace corresponding to the plaintext, where the value of 4 most (least) significant bits of all 16 bytes is $f(g)$ in decimal format; namely, $\{p_i^m = f\}_{i=1}^{16}$ ($\{p_i^l = g\}_{i=1}^{16}$).

However, if the superscript is a certain digit, it shows that the plaintext is this value or power trace is corresponding to the plaintext with this value. For example, p_1^{128} means that the first byte of plaintext equals 128 in decimal format and T_1^{128} is denoted as the power trace of the first S-box operation with the corresponding plaintext byte being 128. Meanwhile, we use $T(n)$ and $P(n)$ for the *nth* acquisition of power traces and plaintexts, respectively.

2.2. Linear Collision Attack. The internal collision was first presented for attacking DES [9]. It is based on the fact that if a collision on a key-dependent function can be detected, the attacker can acquire some relations between the different inputs.

Linear collision is based on the internal collision. When it is applied to AES, if a collision between two S-boxes operations of the first round is detected (e.g., the collision between the *ith* and the *jth S-boxes* in Figure 1), it is obvious that (4) is tenable:

$$S\_box(p_i \oplus k_i) = S\_box(p_j \oplus k_j) \tag{4}$$

Then one can obtain a linear equation about the relation between plaintexts and first round subkey:

$$p_i \oplus p_j = k_i \oplus k_j = \Delta k_{i,j} \tag{5}$$

Online Stage:
(1) $\{P^n \mid n = 0, 1, 2, 3, \cdots, N\} \leftarrow$ **Plaintexts**
(2) $\{T^n \mid n = 0, 1, 2, 3, \cdots, N\} \leftarrow$ **AcquireTrace**$(\{P^n\}_{n=0}^{N})$
Offline Stage:
(3) $\{T_1(n) \mid n = 0, 1, 2, 3, \cdots, N\} \leftarrow$ **CutTrace**$(\{T^n\}_{n=0}^{N})$
(4) $\{T_2(n) \mid n = 0, 1, 2, 3, \cdots, N\} \leftarrow$ **CutTrace**$(\{T^n\}_{n=0}^{N})$
(5) $\{\overline{T_1^b} \mid b = 0, 1, 2, 3, \cdots, 255\} \leftarrow$ **AverageTraces**$(\{T_1(n)\}_{n=0}^{N})$
(6) $\{\overline{T_2^b} \mid b = 0, 1, 2, 3, \cdots, 255\} \leftarrow$ **AverageTraces**$(\{T_2(n)\}_{n=0}^{N})$
(7) **for** $\Delta \in \{0, 1, 2, 3, \cdots, 255\}$
(8) $\quad \rho(\Delta) \leftarrow$ **Correlation**$(\{\overline{T_1^b}\}_{b=0}^{255}, \{\overline{T_2^{b\oplus\Delta}}\}_{b=0}^{255})$
(9) **end for**
(10) **return** $\arg\max_{\Delta}\rho(\Delta)$

Algorithm 1: Correlation enhanced detection of S-box 1 and S-box 2.

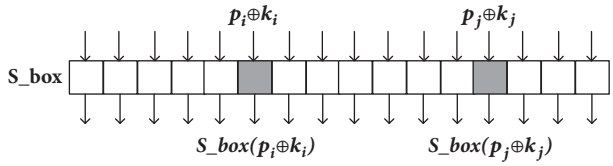


Figure 1: Collisions between two S_boxes.

If the attacker can find all possible relations among 16 key bytes by detecting the collision of *S-boxes*, then he will obtain an equation set about the key bytes of the first round containing 15 linear equations:

$$\begin{aligned} k_1 \oplus k_2 &= \Delta k_{1,2} \\ k_2 \oplus k_3 &= \Delta k_{2,3} \\ &\vdots \\ k_{15} \oplus k_{16} &= \Delta k_{15,16} \end{aligned} \tag{6}$$

Note that all the equations in the set are relevant, and there is only one free variable. Thus, this equation set only has 2^8 possible solutions, which means that we just need to enumerate all 256 possible candidates of 1 key byte to recover the whole key value.

However, under the noisy experiment setting, the collision detection method may detect wrong collisions, which lead to some incorrect equations in (6). For all equations in (6) are relevant, even if only one bit error occurs in any equation, the equation system will have no solution. Thus, in this paper, we propose a detection method called DDVD which is to ensure a high detection success rate.

2.3. Voting Detection Methods. In [14], Bogdanov proposed the voting detection containing binary voting test and ternary voting test. Both binary and ternary voting tests are based on Euclidean Distance. For binary voting test, if an attacker can acquire a lot of power traces of the same plaintexts, he will calculate the Euclidean Distance between two trace pairs of different plaintexts. Then the attacker should calculate the total number of the trace pairs whose distance is less than a predetermined threshold. When the total number is more than the predetermined voting value, it can be confirmed that one collision is detected. However, the basic strategy of ternary voting test is the same as binary voting test, but instead of calculating the Euclidean Distance directly, this method requires calculating the distance between each of the obtained power traces with certain plaintexts and the reference power traces that are obtained during a profiling phase preparing a set of reference traces without knowing related encrypting values.

2.4. Correlation-Enhanced Collision Attack. Correlation-enhanced collision attack was one of the last major advanced detection techniques proposed by Moradi et al in 2010 [15]. This method compares the correlation coefficient between two sets of power traces corresponding to two different S-boxes rather than detecting the collision between two single power traces.

As can be seen in Algorithm 1, we take the detection between S-box 1 and S-box 2 as an example. In the online stage, an attacker should obtain N power traces corresponding to N plaintexts. When in offline stage, the attacker cuts the power trace into 16 sections based on the operation of 16 S-boxes and takes the section for S-box 1 and S-box 2. Then $T_1(n)$ is divided into 256 groups according to the plaintext byte value and the attacker can get the averaged power traces of each group $(\{\overline{T_1^b}\}_{b=0}^{255})$, which is the same for S-box 2. Next, for each value of $\Delta \in GF(2^8)$, the attacker rearranges $\{\overline{T_2}\}$ based on the value of $b \oplus \Delta$ and calculates correlation coefficients $\rho(\Delta)$ between $\{\overline{T_1^b}\}_{b=0}^{255}$ and $\{\overline{T_2^{b\oplus\Delta}}\}_{b=0}^{255}$. If $\Delta = k_1 \oplus k_2$, the correlation $\rho(\Delta)$ shall reach a maximum value; otherwise, it should have a pretty low value.

2.5. LDPC Decoding Problem in Collision Attack. In [16], Gérard et al. proposed a unified and optimized collision attack method. The proposed method rewrote the linear collision attack as a LDPC decoding problem, according to the linear relationship:

$$\Delta k_{i_1,i_2} \oplus \Delta k_{i_2,i_3} = \Delta k_{i_1,i_3} \quad (\forall 1 \le i_1 < i_2 < i_3 \le 16) \tag{7}$$

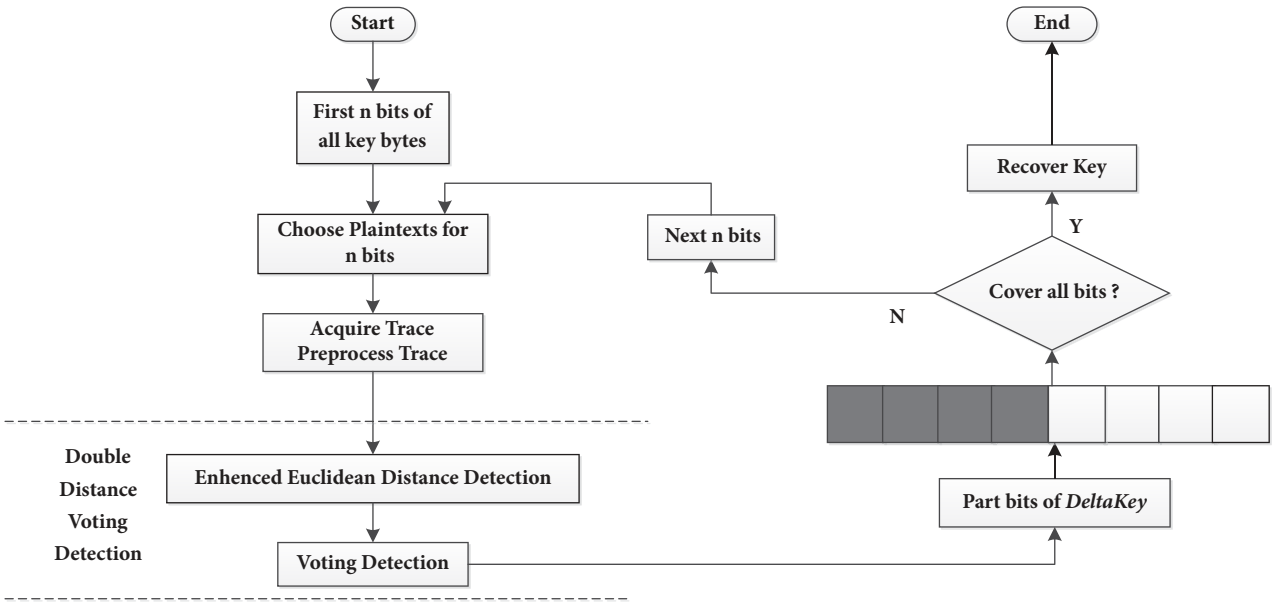


Figure 2: Framework of multiple-bits (n bits) collision attack.

(1) $\{P(n) \mid n = 1, 2, 3, \cdots, 16N\} \leftarrow$ **ChoosePlaintexts**()
(2) $\{T^{m\_j}(n) \mid n = 1, 2, 3, \cdots, N\}_{j=0}^{15} \leftarrow$ **AcquireTrace**$(\{\mathbf{P(n)}\}_{\mathbf{n=1}}^{\mathbf{16N}})$
(3) $\{\bar{t}_i^{m\_j} \mid 1 \le i \le 16\}_{j=0}^{15} \leftarrow$ **PreProTrace**$(\{\mathbf{T^{m\_j}(n)} \mid \mathbf{0} \le \mathbf{n} \le \mathbf{N}\}_{\mathbf{j=0}}^{\mathbf{15}})$
(4) **for** each $(\{(i_1, i_2) \mid 1 \le i_1 < i_2 \le 16\})$
(5) $\quad \Delta k_{i_1,i_2} \leftarrow$ ***DDVD***$(\{\bar{t}_{i_1}^{m\_j}\}_{j=0}^{15}, \{\bar{t}_{i_2}^{m\_j}\}_{j=0}^{15})$
(6) **end for**
(7) **Recoverkey**$(\Delta_{i_1,i_2}(1 \le i_1 < i_2 \le 16))$

Algorithm 2: Multiple-bits side-channel collision attack.

The vector $\Delta K = (\Delta k_{1,2}, \Delta k_{2,3}, \cdots, \Delta k_{15,16})$ can be seen as an LDPC codeword whose dimension is 15 and length is 120. Finding the right key value is just to decode the LDPC code. Furthermore, in order to make the attack method have a better performance in the noisy settings, actual posteriori probability value of each code was used for LDPC decoding, which is called soft decision decoding. Unlike soft decision decoding, hard decision decoding uses bit value for decoding. Compared to soft decision decoding, the effect of hard decision decoding is worse in a noise setting but the computation complexity is lower.

In this paper, for the error-tolerant and check mechanism, we choose top three possible values for each $\Delta k_{i_1,i_3}$ as candidates and find the likeliest value based on (7). It may be seen as a kind of hard decision decoding procedure. However, due to the DDVD, the detection success rate may remain in a high level in some noisy settings.

3. A Novel Framework of Multiple-Bits Collision Attack

In this section, a framework of multiple-bits side-channel collision attack is presented. As can be seen in Figure 2, plaintexts need to be chosen based on multiple-bits (n-bits) model and then we prepare the power trace. Double distance voting detection is the important part of the framework ensuring the high success probability along with high efficiency. However, the principal part of the framework is based on a circulation. Each iteration stands for an attack, where we obtain only n-bits of a byte relation between all key bytes. After several iterations, the whole byte value of Δ between all key bytes can be acquired, which will be utilized to recover the key value.

Due to the fact that the 4-bit collision attack leads to the highest efficiency, which will be proven and verified in Section 5, we take the 4-bit model as an example to explain our attack method. According to our attack framework, for 4-bit model, 2 iterations are enough to recover the key value, whereof one is for the four most significant bits and the other is for the four least significant bits. In the rest of this paper, we only describe the strategy for the four most significant bits of one byte. The remaining four least significant bits can be found using the same technique.

3.1. The Idea in a Nutshell. For a better understanding, we describe the main flow of our attack strategy in Algorithm 2. As our description is based on 4-bit model, all the following

Input: the total number of plaintexts 16N
Output: 16N plaintexts: $\{\mathbf{P(n)}\}$ $(\mathbf{1 \le n \le 16N})$
(1) **for** $j = 0$: 15
(2) **for** $h = 1$: N
(3) $P(16j + h) = \{p_i \mid p_i^m = j, p_i^l = random(16)\}_{i=1}^{16}$
(random(n) is to generate an integer ranging from 0 to n-1)
(4) **end for**
(5) **end for**
(6) **return** $\{P(n)\}$ $(1 \le n \le 16N)$

Algorithm 3: *ChoosePlaintexts*.

Input: 16 sets of power traces:$\{\mathbf{T^{m\_j}(n) \mid n = 1, 2, 3, \cdots, N}\}_{\mathbf{j=0}}^{\mathbf{15}}$
Output: 16 averaged traces:$\{\overline{\mathbf{T}}^{\mathbf{m\_j}}\}_{\mathbf{j=0}}^{\mathbf{15}} = \{\bar{\mathbf{t}}_{\mathbf{i}}^{\mathbf{m\_j}} \mid \mathbf{i = 1, 2, 3, \cdots, 16}\}_{\mathbf{j=0}}^{\mathbf{15}}$
(1) **for** $j = 0$: 15
(2) $\overline{T}^{m\_j} = (1/N) \sum_{n=1}^{N} T^{m\_j}(n)$
(3) Cut $\overline{T}^{m\_j}$ into 16 sub-traces: $\overline{T}^{m\_j} = \{\bar{t}_i^{m\_j} \mid i = 1, 2, 3, \cdots, 16\}$
(4) **end for**
(5) **return** $\{\overline{T}^{m\_j}\}_{j=0}^{15} = \{\bar{t}_i^{m\_j} \mid i = 1, 2, 3, \cdots, 16\}_{j=0}^{15}$

Algorithm 4: *PreprocessTraces*.

statements can be applied to our multiple-bits models. For example, some parameters of 4-bit model are 15 or 16, which can be interpreted as $2^4 - 1$ or 2^4, and thus for other n-bits model, the parameters should be $2^n - 1$ or 2^n.

Like most of other attack strategies, our method also first gets some traces and proceeds with some preprocessing, seen from Steps1, 2, and 3. Double distance voting detection is the core of our method combining enhanced Euclidean Distance detection and voting detection, which ensures our success rate. Finally, based on the main idea in Section 2.2, when we find all possible relations among 16 key bytes, the brute force way is able to find the right key value quickly, for we only need to enumerate 256 key values. Some details will be presented in the following sections.

3.2. Choose Plaintexts. According to our attack strategy, we assume that the attacker is able to choose the plaintext. The 4-bit side-channel collision attack model aims to detect the collision of 4 bits between 2 different S-boxes, and in this situation other 4 bits are the noise for the detection. It is important for improving the efficiency of our method to determine how to choose the value of $\{p_i^m\}_{i=1}^{16}$. Algorithm 3 presents the strategy of plaintexts choice, which can generate $16N$ plaintexts. For $\{p_i^m\}_{i=1}^{16}$ equal to each of the values belonging to $GF(2^4)$, N plaintexts can be obtained with the other 4 bits ($\{p_i^l\}_{i=1}^{16}$) being random. Due to the fact that the value belonging to $GF(2^4)$ ranges from $(0000)_2$ to $(1111)_2$, $16N$ plaintexts should be obtained.

3.3. Acquire Traces and Preprocess Traces. We can obtain $16N$ power traces of the first round operation corresponding to $16N$ plaintexts. The obtained power traces can be divided into 16 sets according to the values of $\{p_i^m\}_{i=1}^{16}$. For the values of $\{p_i^m\}_{i=1}^{16}$ are the same all the time ranging from 0 to 15 referred to Section 3.2 and each value corresponds to N random value for $\{p_i^l\}_{i=1}^{16}$, the number of the sets is 16 and each set contains N power traces.

This can be easily expanded to the attack for the four least significant bits with $\{p_i^l\}_{i=1}^{16}$ ranging from 0 to 15 and $\{p_i^m\}_{i=1}^{16}$ being random.

As for preprocessing the power traces, the detailed procedures are stated in Algorithm 4. For each of the trace sets, we can average all N power traces in this set to a single averaged trace. Each of the averaged power traces is composed of 16 subtraces corresponding to 16 sequential S-boxes operations and can be cut into 16 subtraces. Thus, we can obtain 16 averaged power traces containing 16 subtraces.

3.4. Double Distance Voting Detection. Double distance voting detection is the core of our attack technique ensuring the high success rate and stability. As is seen in Algorithm 5, DDVD is composed of enhanced Euclidean Distance detection and voting detection. For a better understanding of our DDVD technique, a diagrammatic sketch is shown in Figure 3. Taking S-boxes i_1 and i_2 as an example, there shall be 16 subtraces $\{\bar{\mathbf{t}}_{\mathbf{i_1}}^{\mathbf{m\_j_1}}\}_{\mathbf{j_1=0}}^{\mathbf{15}}$ and $\{\bar{\mathbf{t}}_{\mathbf{i_2}}^{\mathbf{m\_j_2}}\}_{\mathbf{j_2=0}}^{\mathbf{15}}$ for S-boxes i_1 and i_2, respectively, after the former operation. Each single trace $\bar{\mathbf{t}}_{\mathbf{i_1}}^{\mathbf{m\_j_1}}$ of $\{\bar{\mathbf{t}}_{\mathbf{i_1}}^{\mathbf{m\_j_1}}\}_{\mathbf{j_1=0}}^{\mathbf{15}}$ should operate the enhanced Euclidean Distance detection with the trace set $\{\bar{\mathbf{t}}_{\mathbf{i_1}}^{\mathbf{m\_j_1}}\}_{\mathbf{j_1=0}}^{\mathbf{15}}$ with 16 traces, which is seen as a decision making unit.

For example, we compute the Euclidean distance between $\bar{\mathbf{t}}_{\mathbf{i_1}}^{\mathbf{m\_6}}$ and each trace of $\{\bar{\mathbf{t}}_{\mathbf{i_2}}^{\mathbf{m\_j_2}}\}_{\mathbf{j_2=0}}^{\mathbf{15}}$, and if the minimum distance is between $\bar{\mathbf{t}}_{\mathbf{i_1}}^{\mathbf{m\_6}}$ and $\bar{\mathbf{t}}_{\mathbf{i_2}}^{\mathbf{m\_j_2}}$, this decision making unit generates one of the possible values for $\Delta k_{i_1,i_2}^m$, namely, $\Delta_6 = (0110)_2 \oplus$

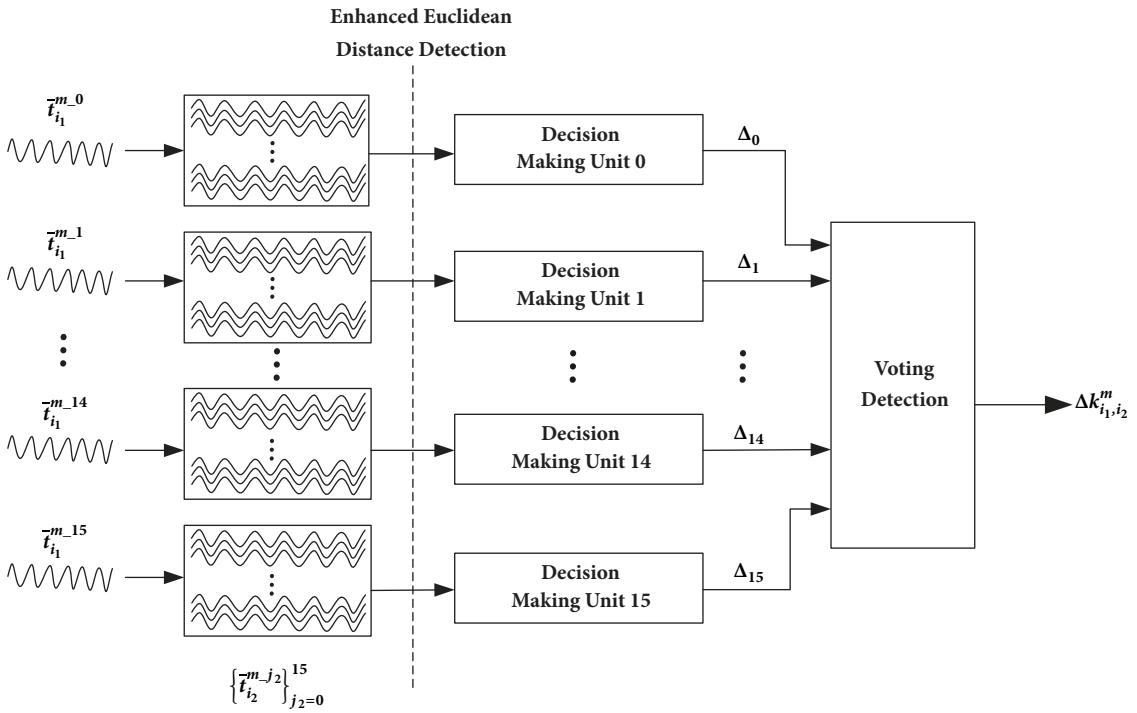


Figure 3: Flow of double distance voting detection.

Input: 2 sets of sub-traces: $\{\bar{\mathbf{t}}_{\mathbf{i}_1}^{\mathbf{mj_1}}\}_{\mathbf{j_1}=0}^{\mathbf{15}}, \{\bar{\mathbf{t}}_{\mathbf{i}_2}^{\mathbf{mj_2}}\}_{\mathbf{j_2}=0}^{\mathbf{15}}$
Output: the 4 most significant bits of $\Delta\mathbf{k}_{\mathbf{i_1},\mathbf{i_2}}$: $\Delta\mathbf{k}_{\mathbf{i_1},\mathbf{i_2}}^{\mathbf{m}}$
Enhanced Euclidean Distance Detection:
(1) **for** $(0 \le j_1 \le 15)$
(2) **for** $(0 \le j_2 \le 15)$
(3) $Distance(j_1 \oplus j_2) = \sum_{l=1}^{L} (\bar{t}_{i_1,l}^{mj_1} - \bar{t}_{i_2,l}^{mj_2})^2$
(4) **end for**
(5) $\Delta_{j_1} = \arg\min_{j_1 \oplus j_2} Distance(j_1 \oplus j_2)$
(6) **end for**
Voting Detection:
(7) $num_n = 0 (0 \le n \le 15)$
(8) **for** $(0 \le j \le 15)$
(9) **for** $(0 \le n \le 15)$
(10) **if** $(\Delta_j = n)$
(11) $num_n = num_n + 1$
(12) **else**
(13) $num_n = num_n$
(14) **end if**
(15) **end for**
(16) **end for**
(17) $\Delta k_{i_1,i_2}^m = \arg\max_n num_n$
(18) **return** $\Delta k_{i_1,i_2}^m$

Algorithm 5: Double distance voting detection.

j_2. This must be done for all 16 single traces of $\{\bar{\mathbf{t}}_{\mathbf{i}_1}^{\mathbf{m\_j_1}}\}_{\mathbf{j_1}=0}^{\mathbf{15}}$; therefore there shall be 16 decision making units generating 16 possible values $\{\Delta_{j_1}\}_{j_1=0}^{15}$ for the candidates of $\Delta k_{i_1,i_2}^m$. During voting detection stage, the value that occurs the maximum times among $\{\Delta_{j_1}\}_{j_1=0}^{15}$ will be voted as the final value of $\Delta k_{i_1,i_2}^m$.

4. Improved Framework

In this section, we propose an improved framework of multiple-bits side-channel collision attack, where we modify our double distance voting detection and insert the error-tolerant and check mechanism. As is shown in Figure 4, in this new framework, the modified double distance detection works with the error-tolerant and check mechanism, which leads to a remarkable promotion in the success rate as well as the attack efficiency.

We still take the 4-bit model as an example to describe the improved attack framework of our method. The procedure is shown in Algorithm 6. Like Section 3, we only care about the four most significant bits, with the four least significant bits being almost the same. Algorithms of *ChoosePlaintexts*, *AcquireTrace*, and *PreprocessTrace* are all the same. In the rest of this section, we only explain the modified double distance voting detection and the fresh error-tolerant and check mechanism.

4.1. Modified Double Distance Voting Detection. The modified detection method is shown in Algorithm 7. Just like the original one, the input of the DDVD is still 2 sets of subtraces corresponding to 2 different S-boxes, but the output changes from a single value to a 1×3 matrix including 3 candidate values of $\Delta k_{i_1,i_2}^m$. Euclidean Distance between each subtrace of a certain S-box and a set of subtraces of another S-box also should be calculated first. Then, instead of choosing n with the maximum number as the result, we prefer three values whose number is in the top three $\Delta k_{i_1,i_2}^m$ where i_1 and i_2 range from 1 to 15.

(1)$\{P(n)|n = 1, 2, 3, \cdots, 16N \leftarrow$ **ChoosePlaintexts**$()\}$
(2)$\{T^{mj}(n) \mid n = 0, 1, 2, 3, \cdots, N\}_{j=0}^{15} \leftarrow$ **AcquireTrace**$(\{P(n)\}_{n=1}^{16N})$
(3)$\{\bar{t}_i^{mj} \mid 1 \le i \le 16\}_{j=0}^{15} \leftarrow$ **PreProcessTrace**$(\{T^{mj}(n) \mid 0 \le n \le N\}_{j=0}^{15})$
(4) **for** each $(\{(i_1, i_2) \mid 1 \le i_1 < i_2 \le 16\})$
(5) $\Delta k_{i_1,i_2}^{m}[1: 3] \leftarrow$ **DDVD**$(\{\bar{t}_{i_1}^{mj}\}_{j=0}^{15}, \{\bar{t}_{i_2}^{mj}\}_{j=0}^{15})$
(6) **end for**
(7)$\{\Delta k^{m}[1: 15], pass\} \leftarrow$ **Error_tolerant**$(\Delta k_{i_1,i_2}^{m}[1: 3] \mid 1 \le i_1 < i_2 \le 16\})$
(8) **if** $(pass=1)$
(9) **Recoverkey** $(\Delta k^{m}[1: 15])$
(10) **else**
(11) **Back to (1)**
(12) **end if**

Algorithm 6: Improved framework.

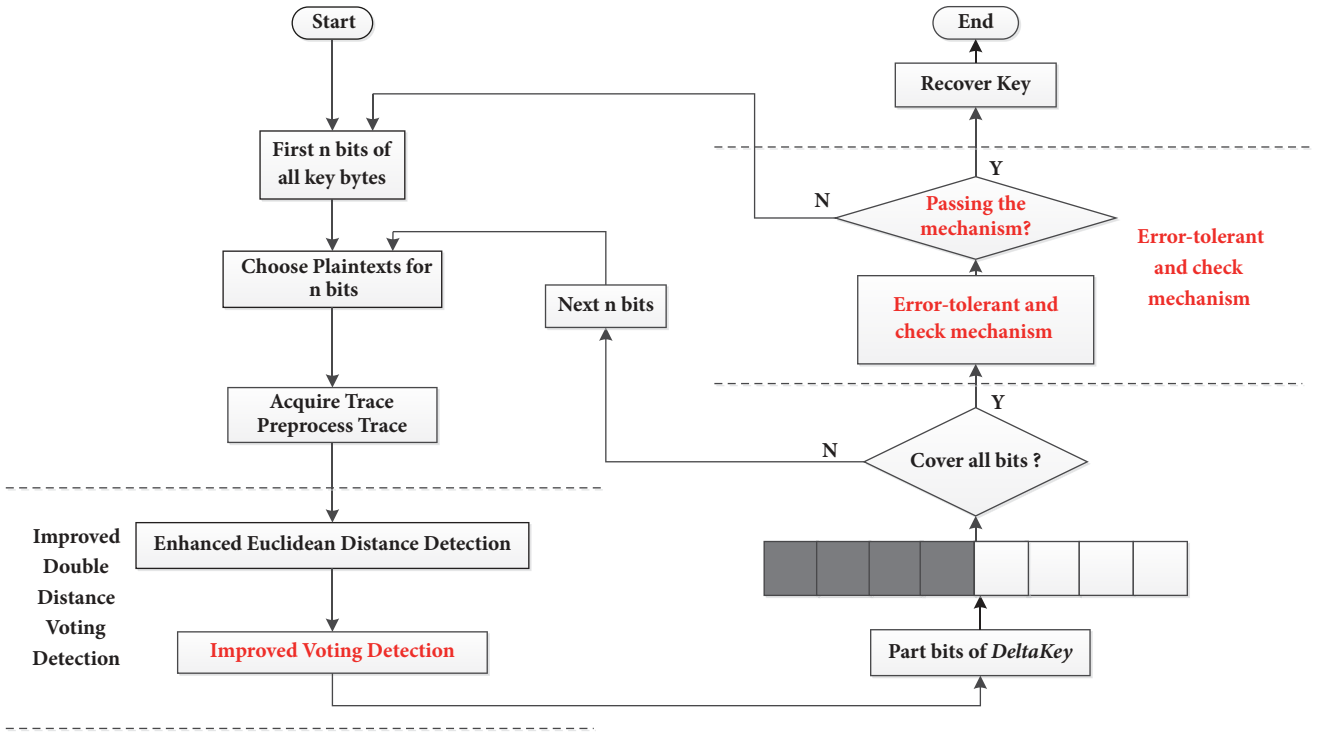


Figure 4: Improved framework with error-tolerant mechanism.

4.2. Error-Tolerant and Check Mechanism. Error-tolerant and check mechanism is presented in Algorithm 8. Three candidate values ensure the error-tolerant mechanism. The main thought of the error detection and tolerance is based on (6), which provides a way to find errors occurring in collision detections.

In order to recover the key value correctly, 15 delta values $(\Delta k_{1,2}, \Delta k_{2,3}, \ldots, \Delta k_{15,16})$ should be right. Thus, for each candidate of $\Delta k_{n,n+1}$, any exiting relations in (7) should be checked. If there exists a candidate of $\Delta k_{n,n+1}$ that can pass the check, it will be the final result of $\Delta k_{n,n+1}$; otherwise this attack is considered to have failed and should start from the beginning again.

5. Model Analysis and Experiments Results

5.1. Comparison of Detection Success Rate under Noise. Collision detection technique has played an important role in side-channel collision attack. In this section, we compare double distance voting detection (DDVD) proposed in this paper with other two widely used detection techniques, which are correlation-enhanced detection[15] and traditional Euclidean Distance detection with dimension reduction[17], respectively.

The detailed procedure of correlation-enhanced detection is already proposed in Section 2.4. In [17], Bogdanov presented a collision attack method based on Euclidean Distance combining DPA. However, according to the method, if

Input : 2 sets of sub-traces:$\{\bar{\mathbf{t}}_{\mathbf{i_1}}^{\mathbf{m,j_1}}\}_{\mathbf{j_1=0}}^{\mathbf{15}}, \{\bar{\mathbf{t}}_{\mathbf{i_2}}^{\mathbf{m,j_2}}\}_{\mathbf{j_2=0}}^{\mathbf{15}}$
Output: the 4 most significant bits of $\Delta k_{i_1,i_2}^{m}$: $\mathbf{\Delta k_{i_1,i_2}^{m}}$**(1 × 3 matrix)**
Enhanced Euclidean Distance Detection:
(1) **for** $(0 \le j_1 \le 15)$
(2) **for** $(0 \le j_2 \le 15)$
(3) $\text{Distance}(j_1 \oplus j_2) = \sum_{l=1}^{L} (\bar{t}_{i_1,l}^{m,j_1} - \bar{t}_{i_2,l}^{m,j_2})^2$
(4) **end for**
(5) $\Delta_{j_1} = \arg\min_{j_1 \oplus j_2} \text{Distance}(j_1 \oplus j_2)$
(6) **end for**
Improved Voting Detection:
(7) $\text{num}_n = 0 (0 \le n \le 15)$
(8) **for** $(0 \le j \le 15)$
(9) **for** $(0 \le n \le 15)$
(10) **if** $(\Delta_j = n)$
(11) $num_n = num_n + 1$
(12) **else**
(13) $num_n = num_n$
(14) **end if**
(15) **end for**
(16) **end for**
(17) $\Delta k_{i_1,i_2}^{m}[1] = \arg\max_n num_n$
(18) $\Delta k_{i_1,i_2}^{m}[2] = \arg\max_n num_n \ (n \ne \Delta k_{i_1,i_2}^{m}[1])$
(19) $\Delta k_{i_1,i_2}^{m}[3] = \arg\max_n num_n \ (n \ne \Delta k_{i_1,i_2}^{m}[1], \Delta k_{i_1,i_2}^{m}[2])$
(20) **return** $\Delta k_{i_1,i_2}^{m}$

Algorithm 7: Double distance voting detection (modified).

Input : 3 candidates for each $\Delta k_{i_1,i_2}^{m}$: $\{\mathbf{\Delta k_{i_1,i_2}^{m}[1:3]} \mid \mathbf{1 \le i_1 < i_2 \le 16}\}$
Output: $\mathbf{\Delta k^{m}[1:15]}$, **pass**
(1) **for** $(1 \le i_1 \le 14)$
(2) $i_2 = i_1 + 1$
(3) **for** $(1 \le t \le 3)$
(4) **if** (existing $x \in \Delta k_{i_2,h}^{m}[1:3]$ *and* $y \in \Delta k_{i_1,h}^{m}[1:3] (i_2 < h \le 16)$
satisfying $\mathbf{\Delta k_{i_1,i_2}^{m}[t] = x \oplus y}$)
(5) $\Delta k^{m}[i_1] = \Delta k_{i_1,i_2}^{m}[t]$ $pass = 1$
(6) **else**
(7) $pass = 0$ exiting all loops
(8) **end if**
(9) **end for**
(10)**end for**

Algorithm 8: Error-tolerant.

collision detection generates incorrect results, DPA makes no sense for recovering the right key value. Therefore, the success rate of Euclidean Distance detection for that method is the key part.

The power traces are obtained from an AES hardware design implemented on a SAKURA-G board. Each trace shall be averaged by four power traces with the same input. The noise of the traces usually comes from both electronic noise mainly containing power supply noise, clock generator noise, conducted emissions, and radiated emissions and algorithm noise which are the power assumption of other uncorrelated operations. For SAKURA-G is a dedicated board that may be far from being noisy, we can add the Gaussian noise of different intensity into the averaged traces to model the noise, which can be used for an initial analysis of efficiency of different detection techniques [14]. SNR (signal-to-noise ratio) is used for indicating the intensity of the noise, which is defined as follows:

$$SNR = 10 \log_{10} \frac{P_{signal}}{P_{noise}} \tag{8}$$

The comparison result is shown in Figure 5. Detection technique proposed in this paper is marked with DDVD, and correlation-enhanced method in [15] and Euclidean Distance in [17] are denoted as CE and ED, respectively. The value of SNR ranges from 0 dB to 30 dB. Each technique is done

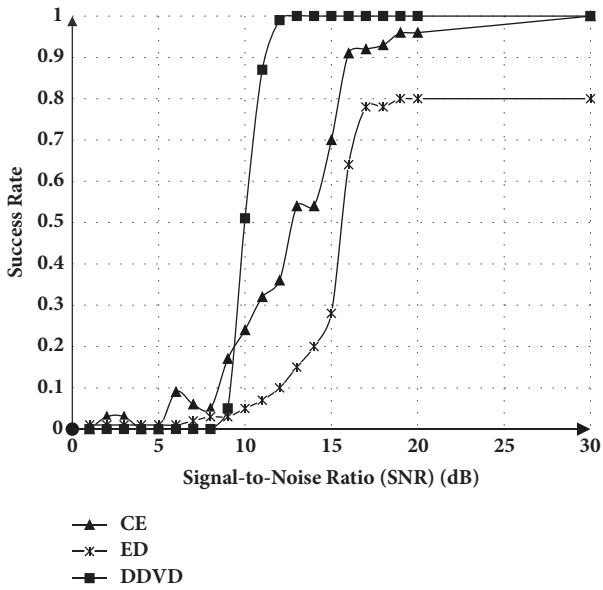


Figure 5: Comparison of detection success rate.

for 1000 times to calculate the success rate. As is shown in Figure 5, our detection technique performs better under the noise.

5.2. How Many Bits Are Best for Multiple-Bits Model. In this paper, we propose a multiple-bits collision attack model, and all the statements are based on the 4-bit model, which can be expanded to other n-bits (n ranging from 1 to 8) models. However, one question we should figure out is how many bits are best for the multiple-bits model of our attack method.

What matters in our attack strategy is the necessary number of power traces to reach a given success rate, which reflects the attack efficiency. Therefore, we will analyze the necessary number of power traces for different model both in theory and in experiment.

In Section 5.1, the performance of our detection method in noise environment is presented; thus, for a simple analysis in this section, we assume that the noise for an n-bits model only comes from the operation of other *8-n* bits and all 2^n decision making units in Figure 3 are independent. So, for the n-bits model, when preparing the power traces (Sections 3.2 and 3.3), we need h power traces of each byte to obtain the averaged traces. When we compute the Euclidean Distance between two single averaged traces whose n-bits can cause the collision (e.g., $p_{i_1}[n-1:0] \oplus p_{i_2}[n-1:0] = \Delta k_{i_1 i_2}[n-1:0]$), the probability that these two averaged traces have the least Euclidean Distance is

$$Pr = 1 - \frac{C_{2^{8-n}}^{h} \times C_{2^{8-n}-h}^{h}}{C_{2^{8-n}}^{h} \times C_{2^{8-n}}^{h}} \tag{9}$$

where letter C is denoted as combinatorial number, n equals the number of bits in the model, and h equals the number of traces for calculating the average. Due to the fact that 8 n-bits of one byte are random, there are a total of $C_{2^{8-n}}^{h} \times C_{2^{8-n}}^{h}$ kinds of choices to determine these two averaged power traces.

Table 1: The necessary number of original attacks to reach 90% success rate.

n-bits	1	2	3	**4**	5	6	7	8
Number	288	160	168	**128**	192	256	512	256

Table 2: The necessary number of improved attacks to reach 90% success rate.

n-bits	1	2	3	**4**	5	6	7	8
Number	No	128	120	**96**	140	256	256	256

Since there shall be only one corresponding plaintext of one byte which can cause a collision with each plaintext of another byte, there are a total of $C_{2^{8-n}}^{h} \times C_{2^{8-n}-h}^{h}$ kinds of choices that include no collision plaintext pair.

The probability of successful detection for the method proposed in Section 3 and the improved method presented in Section 4 is calculated separately as follows:

$$Pr\_det = 1 - \sum_{i=2^{n-1}}^{2^n} C_{2^n}^{i} \times (1 - Pr)^i \times Pr^{2^n - i} \tag{10}$$

$$Pr\_impro = 1 - \sum_{i=2^n - 2^{n-2}}^{2^n} C_{2^n}^{i} \times (1 - Pr)^i \times Pr^{2^n - i} \tag{11}$$

where Pr is equal to (9) and n is the number of bits in the model. From the illustration of Figure 3, there are 2^n decision making units for the n-bits model. According to the rules of voting detection that the value that occurs the maximum times is chosen as the final result, if more than half of the decision making units generate the wrong answer, the voting detection shall fail. Result of all 2^n decision making units can be seen as the binomial distribution, so the probability that more than half of the units generate wrong result is $\sum_{i=2^{n-1}}^{2^n} C_{2^n}^{i} \times (1 - Pr)^i \times Pr^{2^n - i}$. Analysis for the improved method in Section 4 is similar, and if more than three-quarters of the decision making units generate the wrong answer, the voting detection shall fail.

As for the necessary number of the power traces, it can be calculated as follows:

$$Trace\_Number = 2^n \times h \times \left\lceil \frac{8}{n} \right\rceil \tag{12}$$

where $\lceil n \rceil$ stands for the minimum integer that is larger than n. According to our attack strategy, for each n-bits model, we should get 2^n averaged power traces and each trace is averaged by h original power traces. Obviously, the method should be operated for $\lceil 8/n \rceil$ times.

According to (9), (10), (11), and (12), we estimate the necessary number of the power traces to reach a 90% success rate for the basic attack method in Section 3 and an improved version in Section 4, shown in Tables 1 and 2, respectively. The 1-bit model only has two possible results, so improved method makes no sense for it. It is obvious that, in theory, 4-bit model collision attack needs the least number of traces with high efficiency, which will be verified later in experiments.

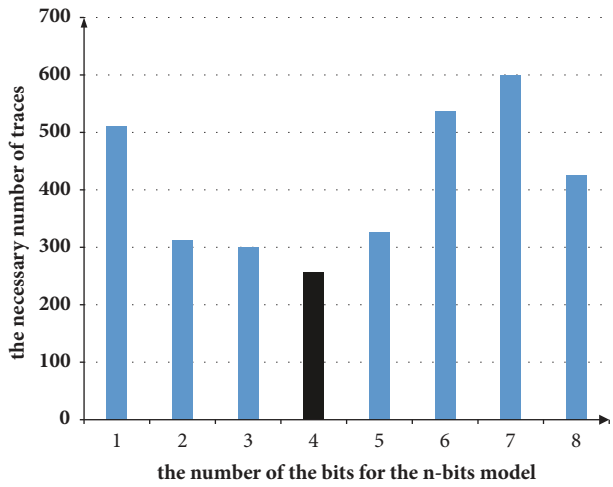


Figure 6: Necessary number of traces for MBDD to reach 90% success rate.

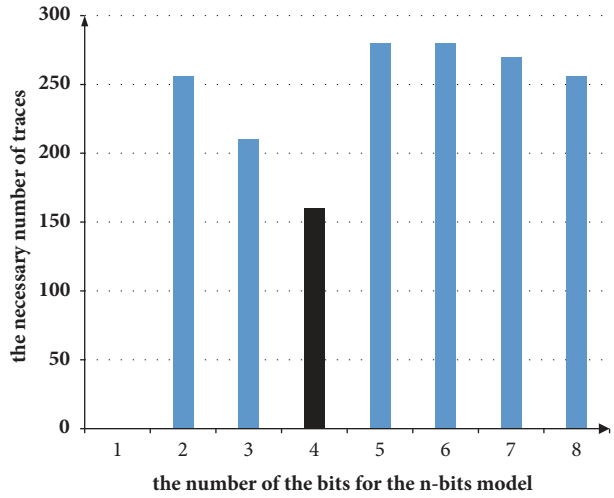


Figure 7: Necessary number of traces for MBDD-ET to reach 90% success rate.

Furthermore, the practical experiments have been carried out to find how many bits are best for the proposed multiple-bits model. Figure 6 shows the necessary number of power traces to reach a 90% success rate for the original approach presented in Section 3 (denoted as MBDD), while Figure 7 shows the he necessary number of power traces for the improved version proposed in Section 4 (denoted as MBDD-ET). As can be seen in Figures 6 and 7, it is verified that 4-bit model (in black) is the best choice when operating the proposed attack strategy.

5.3. Experiments and Results. The attack method and the improved method with error-tolerant mechanism have been performed successfully in practice against a hardware design with an 8-bit data path of AES, where 16 S-boxes are sequentially operated in every round operation. The target AES is implemented on a Xilinx SPARTAN-6 FPGA of a SAKURA-G circuit board. An Agilent MSO-X 9104A oscilloscope is employed to collect original power traces. In our case, each power trace obtained contains about 32365 points.

For a better understanding, operation on k_1^m and k_2^m corresponding to S-box 1 and S-box 2 is taken as an example to present the process of double distance voting detection. Without loss of generality, k_1^m and k_2^m are fixed as

$$k_1^m = (1101)_2\,,$$
$$k_2^m = (0110)_2\,, \tag{13}$$
$$\Delta k_{1,2}^m = (1011)_2\,.$$

Two corresponding sets of subtraces are denoted as $\{\bar{t}_1^{m\_j_1}\}_{j_1=0}^{15}$ and $\{\bar{t}_2^{m\_j_2}\}_{j_2=0}^{15}$. Figure 4 shows square of difference between each subtrace $\bar{t}_1^{m\_j_1}$ of $\{\bar{t}_1^{m\_j_1}\}_{j_1=0}^{15}$ and the trace set $\{\bar{t}_2^{m\_j_1}\}_{j_1=0}^{15}$ with 16 traces. Figures 8(a)–8(p) are like 16 decision making units in Figure 3 corresponding to j_1 ranging from 0 to 15. The accumulation of all points' square of difference between two subtraces is the value of Euclidean Distance.

We take Figure 8(a) as an example to describe its meaning. Figure 8(a) shows the result of $(\bar{t}_{1,l}^{m\_0} - \bar{t}_{2,l}^{m\_j_2})^2$ for each point l ranging from 1 to 32365 and each value of j_2 ranging from 0 to 15. The black curve represents the square of difference between each point of $\bar{t}_{1,l}^{m\_0}$ and $\bar{t}_{2,l}^{m\_0\oplus\Delta k_{1,2}^m}$, which are two traces corresponding to a collision in theory. However, square of differences between $\bar{t}_{1,l}^{m\_0}$ and other traces in the trace set $\{\bar{t}_2^{m\_j_1}\}_{j_1=0}^{15}$ is marked by grey curves. If the black curve is lower than any other grey curves, the decision making unit will generate the right candidate. An initial and rough conclusion can be drawn that when in situations like Figure 8(a), whose black curve is close to zero, two traces corresponding to a collision may have the lowest distance, meaning that the corresponding decision making unit generates the right candidate, but in some exceptional situations such as Figure 8(p), whose black curve is higher than some grey curves, collision cannot be assured by minimum Euclidean Distance and the unit generates the wrong candidate. Therefore, voting detection works to determine the final value of $\Delta k_{1,2}^m$. As is shown in Figure 9, $(1011)_2$ occurs the maximum times, and voting detection chooses it to be the final result.

5.4. Comparison. In this section, we compare our improved attack version denoted as MBDD with correlation-enhanced collision attack [15], bitwise collision attack [19], and LDPC method with Euclidean Distance detection [16] denoted as CECA, BCA, and LDPC, respectively. Comparisons are done from three aspects, which are relation between success rate and necessary number of traces, relation between success rate and online time, and relation between offline time and online time. Each compared method was performed 1000 times for calculating an actual success rate.

In this section, t_{ave} is used for indicating the total time that the oscilloscope spends on capturing and averaging one power trace in real time, and t_s is for indicating the time that the oscilloscope spends on acquiring and saving one trace. Taking Agilent MSO-X 9104A oscilloscope that we use for

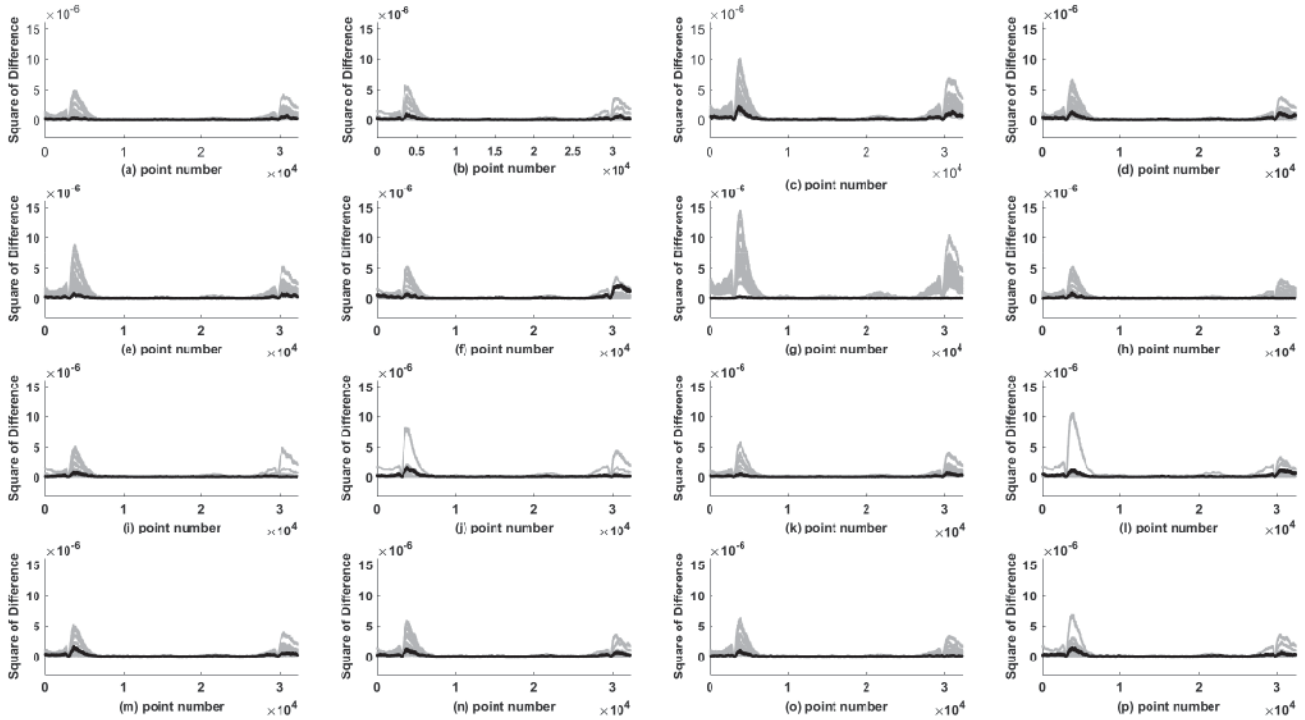


FIGURE 8: Square of difference between each subtrace in set $\{\bar{t}_1^{mj_1}\}_{j_1=0}^{15}$ and all subtraces in set $\{\bar{t}_2^{mj_1}\}_{j_1=0}^{15}$.

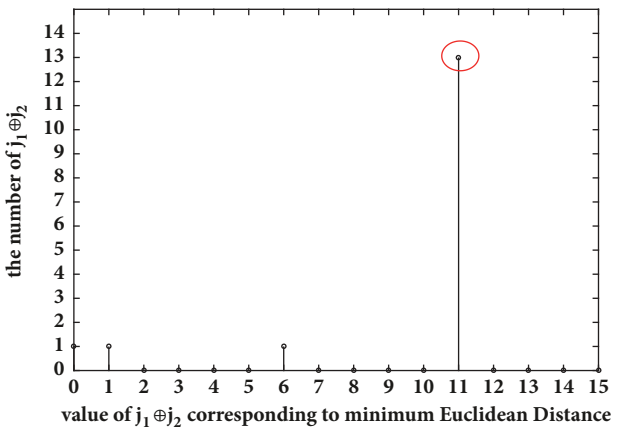


FIGURE 9: Result of the voting stage.

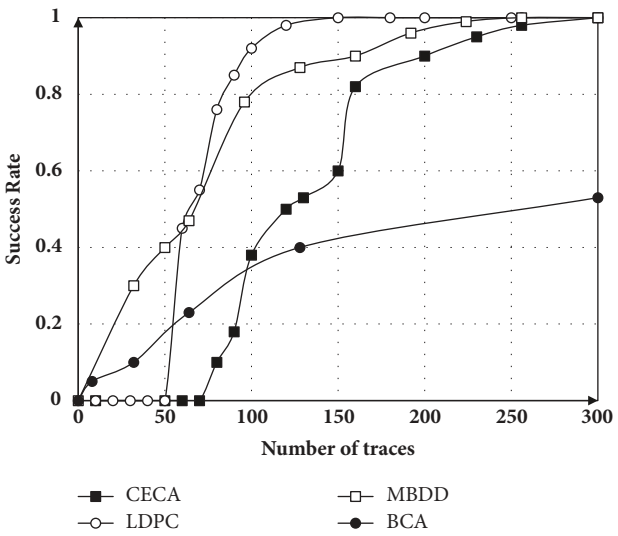


FIGURE 10: Relations between success rate and number of traces.

acquiring power traces as an example, t_s is about 50 times of t_{ave}. The number of power traces used to obtain one averaged power trace in oscilloscope is denoted as q, and the number of saved averaged power traces is n. Therefore, the online time denoted as t_{ol} can be written as

$$t_{ol} = n\left(qt_{ave} + t_s\right) = n\left(0.02q + 1\right)t_s. \tag{14}$$

And we fix $q = 6$ for this experiment, so

$$t_{ol} = 1.12nt_s. \tag{15}$$

Figure 10 presents the relations between success rate and number of traces. As can be seen from Figure 10, LDPC has a better performance. To get a high given success rate, LDPC needs less number of traces. However, in Figure 11, the success rate is as a function of the total online time rather than the number of original power traces. As is mentioned above, we can decrease the online time due to the fact that the time an oscilloscope spends on averaging one trace is much less than saving one trace. It is obvious in Figure 11 that the performance of MBDD with error-tolerant mechanism got a promotion under this setting. Due to the fact that the

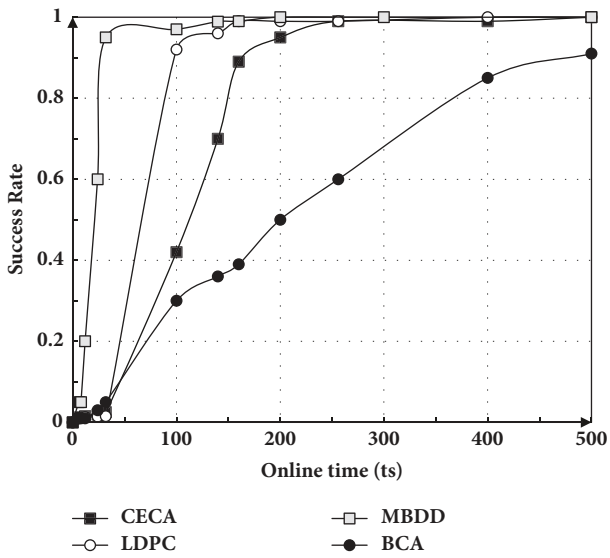


Figure 11: Relations between success rate and online time.

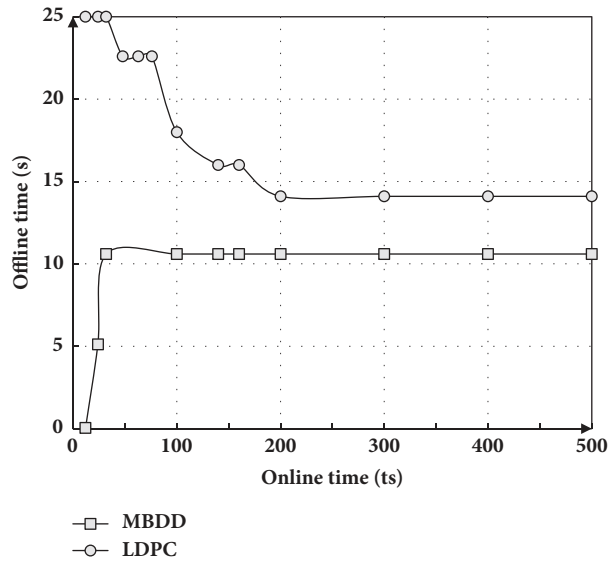


Figure 12: Relations between success rate and online time.

4-bit model of MBDD can find all 120 relations among 16 key bytes with 32 averaged power traces, the fact that the oscilloscope spends less time on averaging traces does a favor for MBDD to have a higher success rate with the same online time. Meanwhile, it seems that LDPC method does not have a remarkable promotion as MBDD with the help of averaging traces. The reason may be that the collision detection method of LDPC will need more averaged traces to detect all collisions occurring among 16 key bytes, even if traces are far from being noisy. However, the results of Figures 10 and 11 can reflect that LDPC is more tolerant to noise because the performance of LDPC in a noisy setting is almost the same as that in a less noisy setting.

Finally, we show the relation between offline time and online time for LDPC and MBDD. The offline time, which reflects the computational complexity, was estimated by MATLAB. As is shown in Figure 12, LDPC is more costly in terms of computation time than MMBD. However, the increased time overhead is slight. For LDPC, the offline time decreases as the online time increases, which indicates that the number of iterations for LDPC decoding decreases. For MBDD, the offline time increases as the online time increases, and it quickly converges to a certain value.

From these comparisons, it can be confirmed that LDPC with soft decision decoding has less trace overhead but more computation time overhead than MBDD, which can be seen as a kind of hard decision decoding procedure. In addition, the necessary number of traces for our method is 90% less than CECA and 96% less than BCA.

6. Conclusion

In this paper, we proposed a basic multiple-bits side-channel collision attack framework based on double distance voting detection. Then an improved version with modified double detection as well as error-tolerant and mechanism is presented. The 4-bit model is proven to be the optimal choice for the novel attack strategy in both theory and practice. Practical attack experiments are performed successfully on a hardware implementation of AES on SAKURA-G circuit board with Xilinx SPARTAN-6. Results show that our detection method performs steadily in noisy environment. We compare our methods with other attacking methods; our method needs less computation time but more traces than LDPC method, and to reach 90% success rate, the necessary number of traces for our method is 90% less than CECA and 96% less than BCA. The novel framework proposed in this paper can be utilized in other cryptographic symmetric algorithms.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Major Program "Core of Electronic Devices, High-End General Chips, and Basis of Software Products" of the Ministry of Industry and Information Technology of China (no. 2014ZX01032205).

References

[1] P. C. Kocher, "Cryptanalysis of Diffie-Hellman, RSA, DSS, and Other Systems Using Timing Attacks," in *Advances in Cryptology—CRYPTO '96*, Lecture Notes in Computer Science, pp. 104–113, Springer, Berlin, Germany, 1996.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 1666, pp. 388–397, 1999.

[3] E. Brier, C. Clavier, and F. Olivier, *Correlation Power Analysis with a Leakage Model International Workshop on Cryptographic*

Hardware and Embedded Systems, Springer, Berlin, Heidelberg, Germany, 2004.

[4] B. Gierlichs et al., *Mutual information analysis: A generic side-channel distinguisher*, Springer-Verlag, 2008.

[5] S. Chari, J. R. Rao, and P. Rohatgi, *Template attacks International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, Berlin, Heidelberg, Germany, 2003.

[6] S. Jin, T. Kim, H. Kim, and S. Hong, "Power Trace Selection Method in Template Profiling Phase for Improvements of Template Attack," *Journal of the Korea Institute of Information Security and Cryptology*, vol. 27, no. 1, pp. 15–23, 2017.

[7] O. Markowitch, L. Lerman, and G. Bontempi, "Side channel attack: an approach based on machine learning," in *Proceedings of the International Workshop on Constructive Side-Channel Analysis and Security Design*, vol. 2011.

[8] H. Dobbertin, "Cryptanalysis of MD4," *Journal of Cryptology*, vol. 11, no. 4, pp. 253–271, 1998.

[9] K. Schramm, T. Wollinger, and C. Paar, "A New Class of Collision Attacks and Its Application to DES," in *Fast Software Encryption*, vol. 2887 of *Lecture Notes in Computer Science*, pp. 206–222, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[10] K. Schramm, G. Leander, P. Felke, and C. Paar, "A Collision-Attack on AES," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, vol. 3156 of *Lecture Notes in Computer Science*, pp. 163–175, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[11] H. Ledig, F. Muller, and F. Valette, "Enhancing Collision Attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, vol. 3156 of *Lecture Notes in Computer Science*, pp. 176–190, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[12] A. Bogdanov, "Improved side-channel collision attacks on AES," in *Proceedings of the 14th International Workshop on Selected Areas in Cryptography*, vol. 4876 of *LNCS*, pp. 84–95, 2007.

[13] C. Clavier, B. Feix, G. Gagnerot, M. Roussellet, and V. Verneuil, "Improved collision-correlation power analysis on first order protected AES," in *Proceedings of the 13th International Workshop on Cryptographic Hardware and Embedded Systems, CHES 2011*, vol. 6917, pp. 49–62, Springer, October 2011.

[14] A. Bogdanov, "Multiple-differential side channel collision attacks on AES," in *Proceedings of the Cryptographic Hardware and Embedded Systems*, vol. 5154 of *LNCS*, pp. 30–40, 2008.

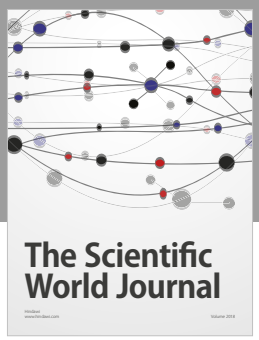
[15] A. Moradi, O. Mischke, and T. Eisenbarth, "Correlation-enhanced power analysis collision attack," in *Cryptographic Hardware and Embedded Systems, CHES 2010*, S. Mangard and F.-X. Standaert, Eds., vol. 6225 of *Lecture Notes in Computer Science*, pp. 125–139, Springer, Berlin, Germany, 2010.

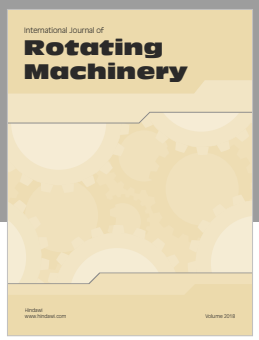
[16] B. Gérard and F. Standaert, "Unified and Optimized Linear Collision Attacks and Their Application in a Non-profiled Setting," in *Cryptographic Hardware and Embedded Systems – CHES 2012*, vol. 7428 of *Lecture Notes in Computer Science*, pp. 175–192, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

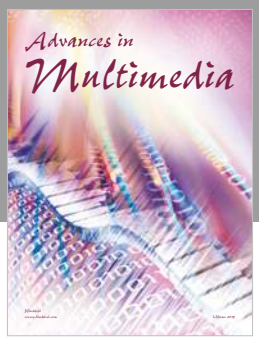
[17] A. Bogdanov and I. Kizhvatov, "Beyond the limits of DPA: combined side-channel collision attacks," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 61, no. 8, pp. 1153–1164, 2012.

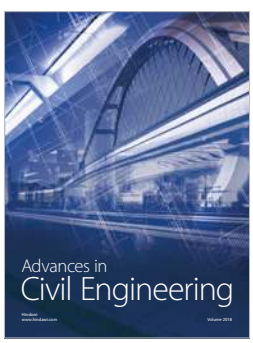
[18] Y. Ren, L. Wu, and A. Wang, "Double sieve collision attack based on bitwise detection," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 1, pp. 296–308, 2015.

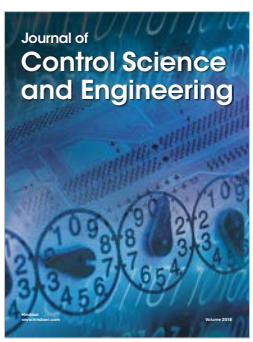
[19] D. Wang and A. Wang, "Bitwise collision attack based on second-order distance," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 3, pp. 1802–1819, 2017.

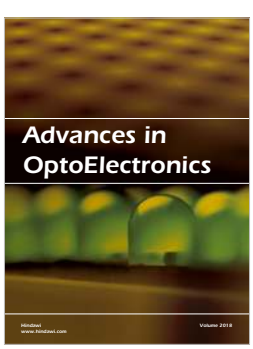


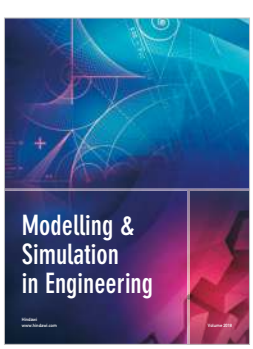


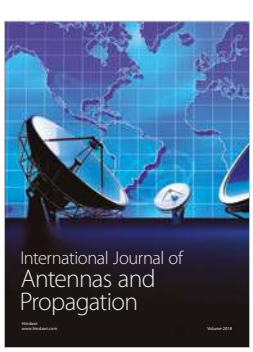


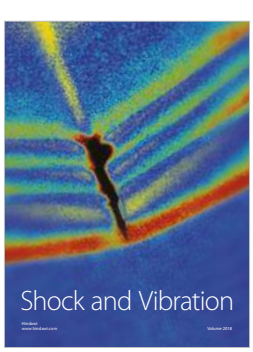


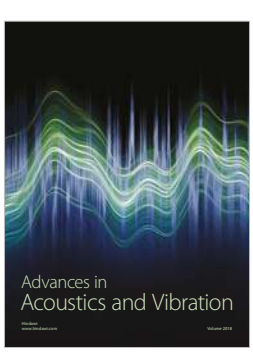












Hindawi

Submit your manuscripts at
www.hindawi.com