Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 209692, 11 pages
http://dx.doi.org/10.1155/2014/209692

Research Article

Two Improved Multiple-Differential Collision Attacks

An Wang,[1] Liji Wu,[1] Zongyue Wang,[2] Xuexin Zheng,[2] Man Chen,[3] and Jing Ma[4]

[1] *Institute of Microelectronics, Tsinghua University, Beijing 100084, China*
[2] *Key Lab of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China*
[3] *Science and Technology on Communication Security Laboratory, Chengdu 610041, China*
[4] *Science and Technology on Information Assurance Laboratory, Beijing 100072, China*

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

Received 4 November 2013; Revised 26 April 2014; Accepted 8 May 2014; Published 27 May 2014

Academic Editor: Ezzat G. Bakhoum

In CHES 2008, Bogdanov proposed multiple-differential collision attacks which could be applied to the power analysis attacks on practical cryptographic systems. However, due to the effect of countermeasures on FPGA, there are some difficulties during the collision detection, such as local high noise and the lack of sampling points. In this paper, keypoints voting test is proposed for solving these problems, which can increase the success ratio from 35% to 95% on the example of one implementation. Furthermore, we improve the ternary voting test of Bogdanov, which can improve the experiment efficiency markedly. Our experiments show that the number of power traces required in our attack is only a quarter of the requirement of traditional attack. Finally, some alternative countermeasures against our attacks are discussed.

1. Introduction

In practice, cryptographic algorithms are widely used in microprocessor, FPGA, and ASIC [1]. Over the years, the traditional cryptanalysis technologies [2] analyze the plaintexts and ciphertexts and recover the secret keys by method of mathematics. In Crypto 1999, Kocher et al. proposed power analysis attack [3] which recovered the secret key by analyzing the instantaneous power consumption of a running chip. In 2003, Schramm et al. gave collision attack [4] in which the equality of two intermediate values can be detected. Its primary step, collision detection, can usually be achieved by executing least square method or least absolute deviation [5] between two power traces. In 2007, Bogdanov presented a linear collision attack on AES [6]. In 2010, Moradi et al. gave a practical linear collision attack named correlation-enhanced collision attack [7]. In CHES 2012, Gérard and Standaert discussed the efficient postprocess on collisions among 16 S-boxes based on LDPC code [8].

In CHES 2008, Bogdanov showed some practical collision detection methods named multiple-differential collision attacks (MDCA) [9] whose idea of voting test seemed to be of much practical value. It consisted of two methods, binary voting test and ternary voting test. However, there exist the following problems in practice, which may lead to the failure of attack experiments.

(i) The variance of power traces with Gaussian noise is not constant. Some countermeasures especially bring high intensity noise in some local sampling points [10, 11].

(ii) The number of key measurement points is not enough because of the low sampling rate of oscilloscopes.

(iii) In some protected devices, the times of encrypting the same set of data repeatedly are limited. In other words, for a fixed plaintext, the number of power traces which can be acquired is limited. So, an efficient collision detection algorithm is required.

Our Contributions. In this paper, we try to overcome the problems above, improve the existing collision detection algorithms, and discuss their countermeasures.

(i) The idea of keypoints voting test which divides the keypoints into some groups of uniform weight for a voting test is proposed. So, all the problems above can

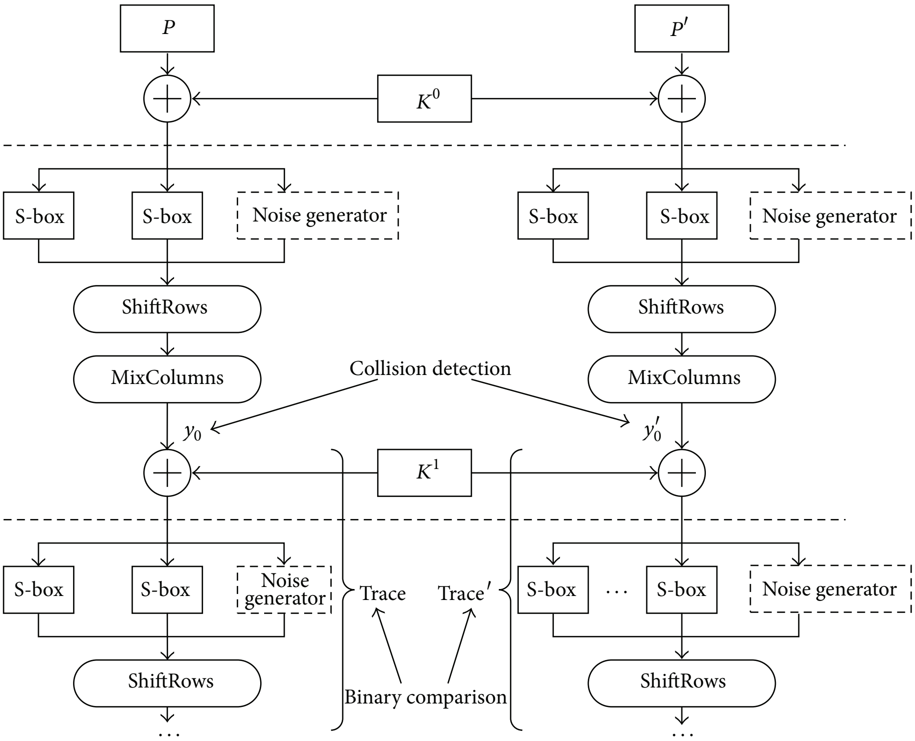


FIGURE 1: Round function of AES algorithm and collision attack on it.

be solved. Subsequently, an experiment environment is built, in which we have verified that the new method can increase the success ratio from 35% to 95%.

(ii) We improved the ternary voting test of Bogdanov by establishing the standard templates during preprocess, which reduces the complexity of collision detection and increases the success ratio markedly. Our experimental investigation shows that the number of power traces required in our attack is only 1/4 of the requirement of traditional attack.

Organization. This paper is organized as follows. In Section 2, we review the traditional collision attacks and their collision detection methods, binary comparison and binary and ternary voting test. In Section 3, keypoints voting test is proposed, and the corresponding experiment results are shown. In Section 4, we improve Bogdanov's ternary voting test and show its theoretical and practical superiority, respectively. Subsequently, we discuss the alternative countermeasures against our attack and show our experiment results in Section 5. Finally, we conclude this paper in Section 6.

2. Related Works

2.1. Collision Attack and Countermeasures. The cryptographic device usually includes a cryptographic chip at least, microprocessor or digital logic circuit, in which one or more cryptographic algorithms are running. The attackers are interested in the secret keys stored in the chip [1]. In the process of power analysis attack, an oscilloscope can be employed for acquiring the instantaneous power consumption of the chip because different operations or operands may consume different powers in practice. Therefore, the power analysis attacks represented by collision attack [4] and correlation power analysis [12] can be mounted effectively. Take collision attack and AES algorithm [13], for example; the attacker executes the following steps.

The first round of AES includes S-boxes, ShiftRows, MixColumns, and AddRoundKey, which is described in Figure 1. Firstly, the attacker chooses two 128-bit plaintexts P and P', encrypts them for m times, respectively, acquires $2m$ power traces, and averages them, respectively. During collision attack, collision detection is the most important step. In order to decide whether two intermediate bytes $y_0 = y_0'$ (see Figure 1), the attack considers the similarity between the two averaged traces which follow y_0 and y_0', respectively. In this step, a collision detection algorithm is needed, which we describe in Sections 2.2 and 2.3.

Usually, the plaintext P is fixed. The collision must happen because the plaintext P' can be changed arbitrarily and the encryption can be repeated over and over again. Once a collision is detected, an equation can be built for reducing key information since one key byte may be expressed by another one [4, 6].

In the past few years, some countermeasures are designed against these attacks, which can be classified in reducing the signal-to-noise ratio (SNR) [11], timing disarrangement [14], masking [15], and hiding [16]. Generating Gaussian noise especially is widely studied, such as the techniques of shift register lookup tables, RAM write collisions, and short

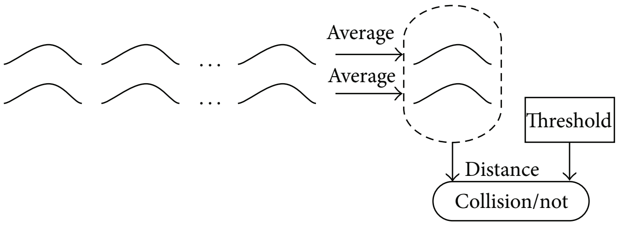


FIGURE 2: Flow chart of binary comparison proposed by Bogdanov.

circuits in switch boxes [11]. Furthermore, dummy rounds/S-boxes [17, 18] can also reduce the SNR markedly.

In the countermeasures above, amplifying local noise usually brings errors to the traditional collision detection, and some collisions would be misjudged as noncollisions. We discuss solution of this problem in Section 3.

2.2. Binary Comparison. Binary comparison (BC) [4] adopts averaging method for reducing the noise. Then the "distance" between the two traces is figured out by least square method. Comparing the distance with a predetermined threshold, collision or noncollision can be decided. Figure 2 shows this process.

Specifically, assuming that operation 1 is executed for m times, let $\tau_1 = (\tau_{1,1}, \tau_{1,2}, \ldots, \tau_{1,l}) \in \mathbb{R}^l$ (including l keypoints and some other points we do not care about) denote the average trace of the m traces. Likewise, $\tau_2 = (\tau_{2,1}, \tau_{2,2}, \ldots, \tau_{2,l}) \in \mathbb{R}^l$ denotes the average trace of operation 2. Collision can be decided:

$$\Psi^{\mathrm{BC}}(\tau_1, \tau_2) = \begin{cases} 0 \ \text{(noncollision)}, & \text{if } \Theta^{\mathrm{BC}}(\tau_1, \tau_2) > Y^{\mathrm{BC}}, \\ 1 \ \text{(collision)}, & \text{if } \Theta^{\mathrm{BC}}(\tau_1, \tau_2) \le Y^{\mathrm{BC}}. \end{cases} \tag{1}$$

Here Y^{BC} is a predetermined threshold, and $\Theta^{\mathrm{BC}}(\tau_1, \tau_2)$ denotes the Euclidean distance between the two traces. Consider

$$\Theta^{\mathrm{BC}}(\tau_1, \tau_2) = \sum_{r=1}^{l} (\tau_{1,r} - \tau_{2,r})^2. \tag{2}$$

2.3. Binary Voting Test of Multiple-Differential Collision Attacks. The binary voting test proposed by Bogdanov [9] constructs m pairs by the $2m$ traces corresponding to operations 1 and 2. Instead of being average, the two traces of each pair are compared, whose result is regarded as a vote (zero or one standing for noncollision or collision, resp.). Finally, collision or noncollision of the two operations can be decided by the sum of vote and a predetermined vote threshold. Figure 3 describes this process, which shows the idea of "multiple-differential."

Let $\tilde{\tau}_1 = \{\tau_1^1, \tau_1^2, \ldots, \tau_1^m\}$ and $\tilde{\tau}_2 = \{\tau_2^1, \tau_2^2, \ldots, \tau_2^m\}$, respectively, denote the m trace corresponding to executing operations 1 and 2 for m times. In collision detection stage, the total vote can be summed based on the binary comparison function Ψ^{BC}:

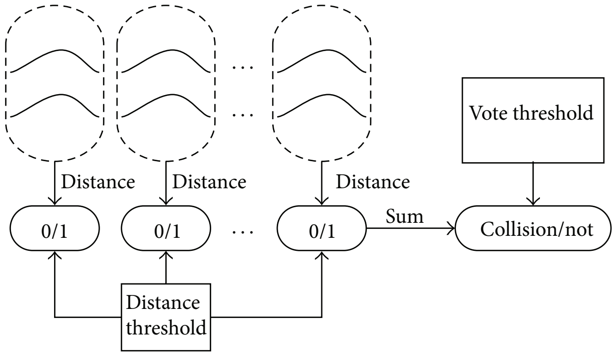


FIGURE 3: Flow chart of binary voting test proposed by Bogdanov.

$$\Theta^{\mathrm{BV}}(\tilde{\tau}_1, \tilde{\tau}_2) = \sum_{i=1}^{m} \Psi^{\mathrm{BC}}(\tau_1^i, \tau_2^i). \tag{3}$$

Then the vote can be compared with a predetermined threshold for the decision of collision. Consider

$$\Psi^{\mathrm{BV}}(\tilde{\tau}_1, \tilde{\tau}_2) = \begin{cases} 0 \ \text{(noncollision)}, & \text{if } \Theta^{\mathrm{BV}}(\tilde{\tau}_1, \tilde{\tau}_2) < Y^{\mathrm{BV}}, \\ 1 \ \text{(collision)}, & \text{if } \Theta^{\mathrm{BV}}(\tilde{\tau}_1, \tilde{\tau}_2) \ge Y^{\mathrm{BV}}. \end{cases} \tag{4}$$

2.4. Ternary Voting Test of Multiple-Differential Collision Attacks. During the preprocess of ternary voting test with profiling [9], a set of reference traces is built firstly. Then, the two traces to be detected are, respectively, compared with every reference trace (binary comparison algorithm can be employed for this comparison). So, every reference trace corresponds to two results whose value may be (0, 0), (0, 1), (1, 0), or (1, 1) assuming 0 and 1 denote noncollision and collision, respectively. At last, collision or noncollision of the two operations can be decided by the number of (1, 1) and a predetermined threshold. Figure 4 describes this process.

In reference traces generation stage, N^{TV} plaintexts are chosen and encrypted once. So, N^{TV} traces denoted as $\overline{\tau}_1, \overline{\tau}_2, \ldots, \overline{\tau}_{N^{\mathrm{TV}}}$ are acquired, which are taken as reference traces. l keypoints are selected from each trace.

Let τ_1 and τ_2, respectively, denote the average trace of m traces corresponding to executing operations 1 and 2 for m times. In collision detection stage, for every reference trace $\overline{\tau}_i$, two binary comparisons are executed, and the two results are multiplied together, which is regarded as one vote:

$$F(\tau_1, \tau_2, \overline{\tau}_i) = \Psi^{\mathrm{BC}}(\tau_1, \overline{\tau}_i) \cdot \Psi^{\mathrm{BC}}(\tau_2, \overline{\tau}_i). \tag{5}$$

When i traverses from 1 to N^{TV}, the total vote can be summed:

$$\Theta^{\mathrm{TV}}(\tau_1, \tau_2) = \sum_{i=1}^{N^{\mathrm{TV}}} F(\tau_1, \tau_2, \tau_i). \tag{6}$$

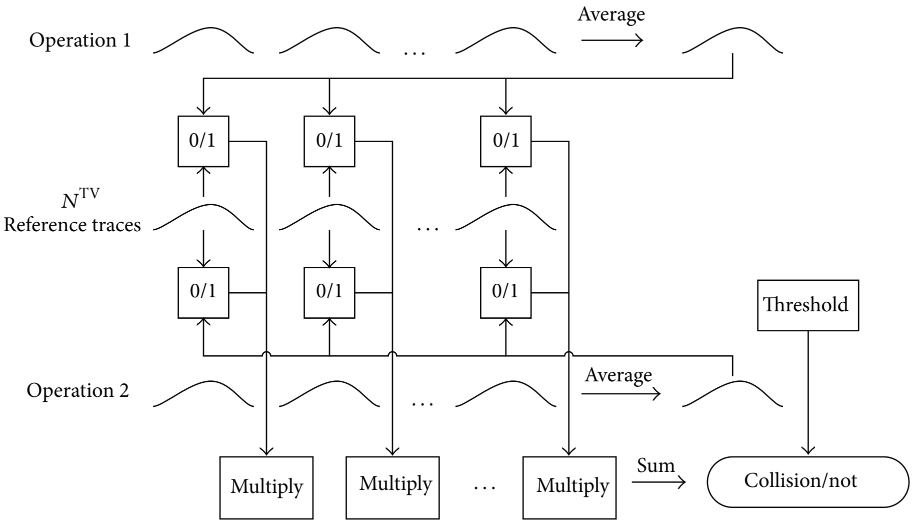


FIGURE 4: Flow chart of ternary voting test, in which 0/1 means noncollision/collision.

Finally, the vote can be compared with a threshold for a determination of collision:

$$\Psi^{TV}(\tau_1,\tau_2)=\begin{cases}0\ \text{(noncollision)}, & \text{if } \Theta^{TV}(\tau_1,\tau_2)<Y^{TV},\\ 1\ \text{(collision)}, & \text{if } \Theta^{TV}(\tau_1,\tau_2)\geq Y^{TV}.\end{cases} \tag{7}$$

Ternary voting test can also be executed without profiling. In other words, each trace of AES encryption acquired in online stage can be divided to 160 reference traces corresponding to 16 S-boxes in 10 rounds. So, the reference traces generation stage can be omitted.

3. Keypoints Voting Test

The countermeasure of amplifying local noise in Figure 1 brings errors to the traditional collision detection. Using for reference the idea of multiple-differential, keypoints voting test proposed in this section can solve this problem well because the local noise can only have influence on a small number of votes even if the noise is high enough.

3.1. Basic Idea. After the averages for reducing the noise, the l keypoint pairs from the two traces vote on the collision, which is described in Figure 5. Let $\tau_1=(\tau_{1,1},\tau_{1,2},\ldots,\tau_{1,l})\in\mathbb{R}^l$ and $\tau_2=(\tau_{2,1},\tau_{2,2},\ldots,\tau_{2,l})\in\mathbb{R}^l$ denote the averaged traces consisting of l keypoints, respectively. For a keypoint pair $(\tau_{1,r},\tau_{2,r})$ $(r=1,2,\ldots,l)$, the vote is defined as

$$\Psi^{PC}(\tau_{1,r},\tau_{2,r})=\begin{cases}0\ \text{(vote 0)}, & \text{if } \left|\tau_{1,r}-\tau_{2,r}\right|>Y_r^{PC},\\ 1\ \text{(vote 1)}, & \text{if } \left|\tau_{1,r}-\tau_{2,r}\right|\leq Y_r^{PC}.\end{cases} \tag{8}$$

Subsequently, the total votes $\Theta^{PV}(\tau_1,\tau_2)$ can be summed:

$$\Theta^{PV}(\tau_1,\tau_2)=\sum_{i=1}^{l}\Psi^{PC}(\tau_{1,r},\tau_{2,r}). \tag{9}$$

Finally, a threshold Y^{PV} is adopted for the collision decision:

$$\Psi^{PV}(\tau_1,\tau_2)=\begin{cases}0\ \text{(noncollision)}, & \text{if } \Theta^{PV}(\tau_1,\tau_2)<Y^{PV},\\ 1\ \text{(collision)}, & \text{if } \Theta^{PV}(\tau_1,\tau_2)\geq Y^{PV}.\end{cases} \tag{10}$$

Remark. There is a compromise between keypoints voting test and binary comparison. Assuming that l is divisible by n, then the l pairs from the two traces are divided into n groups which correspond to n votes. In each group, the l/n pairs can be input into binary comparison algorithm, which output a vote. If the total votes are more than a threshold, collision can be decided.

3.2. Experiment and Efficiency. We adopt EP3C25Q240C6 FPGA of Altera [19] for building the experiment environment, which is described in Figure 6. A resistor of 1 ohm is connected between the power supply and FPGA in series. So, a differential probe connected to an oscilloscope can be employed for acquiring the voltage across the resistance, which is related to the power consumption of FPGA.

We implemented AES in Verilog HDL based on FPGA. The power consumption trace of the 10-round encryption can be gotten, which is shown in Figure 7. In the digital logic circuit of AES, we designed a countermeasure according to the idea of Gaussian noise generator [11]. Random dummy S-boxes join the computation of round function, which amplifies the noise of power consumed by S-boxes locally. Figure 8 which zooms in the part of the first round in Figure 7 shows the local noise. The variance of amplified noise is five times greater than that of the noise from nonprotected implementation.

In the case of the same operation and operands, we acquired two averaged traces for an experiment. 3000 keypoints were selected from each trace. We employed binary comparison and keypoints voting test (every 300 points were regarded as a vote and 10 votes in all) for collision detection. To decide which algorithm was better, we compared the

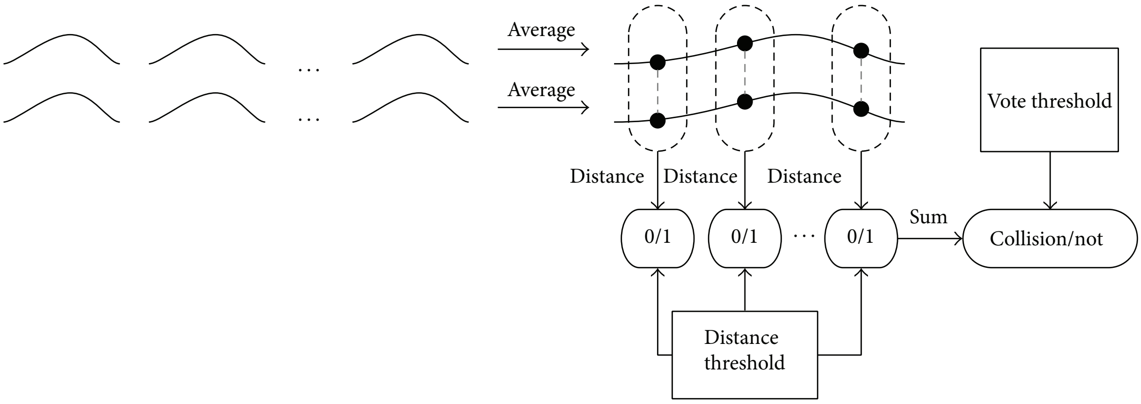


Figure 5: Flow chart of keypoints voting test.

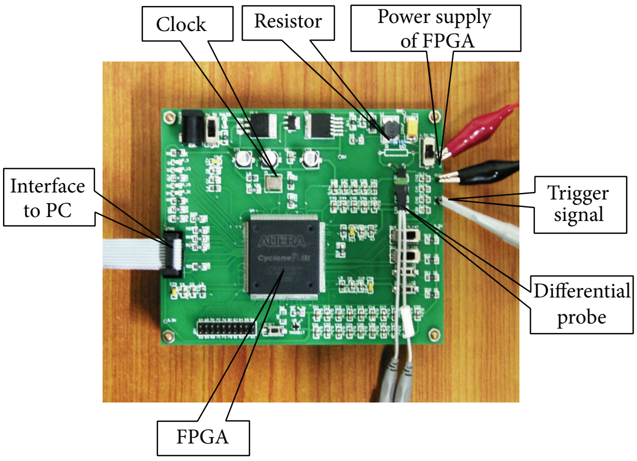


Figure 6: Our experiment environment includes FPGA board, computer, power supply, oscilloscope, and its probes.

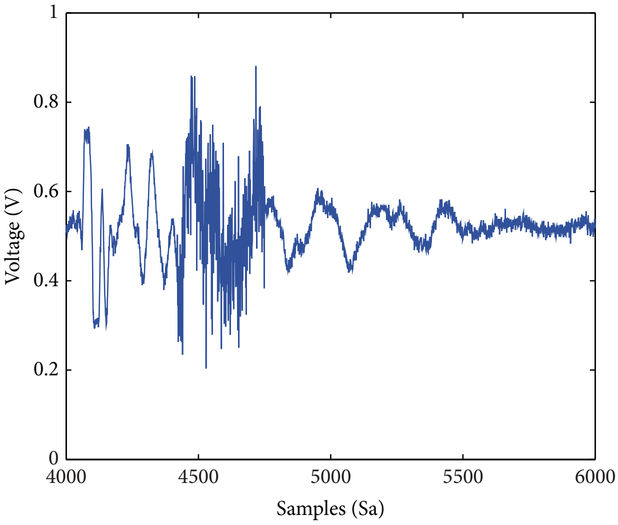


Figure 8: Zooming in the power trace of the first round.

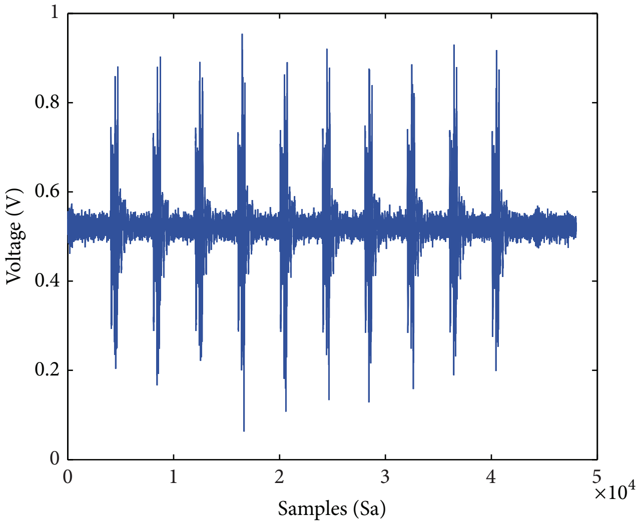


Figure 7: The pretreated power trace of 10 rounds of AES encryption.

success ratio of them, where success meant the result of detection was collision, the same as the fact. After repeating the experiments for many times, it is shown that the success ratios of binary comparison and keypoints voting test are about 35% and 95%, respectively. Figure 9 shows the relation between number of experiments and success ratio. Obviously, the keypoints voting test can overcome the high intensity noise in some local sampling points better than the binary comparison.

We made ten keypoints voting test for determining the number of ballots. Regarding 3000 points as 1 vote, 2 votes,. . ., and 10 votes, respectively, the 10 counts can show the influence on success ratio. If we chose 75% of the total votes as threshold, then the relation between number of ballots and success ratio can be gotten, which is the red line of Figure 10. The blue line means the success ratio of binary comparison which is unrelated to the number of ballots. Obviously, in this environment, dividing 3000 points into more than six votes is scientifically reasonable.

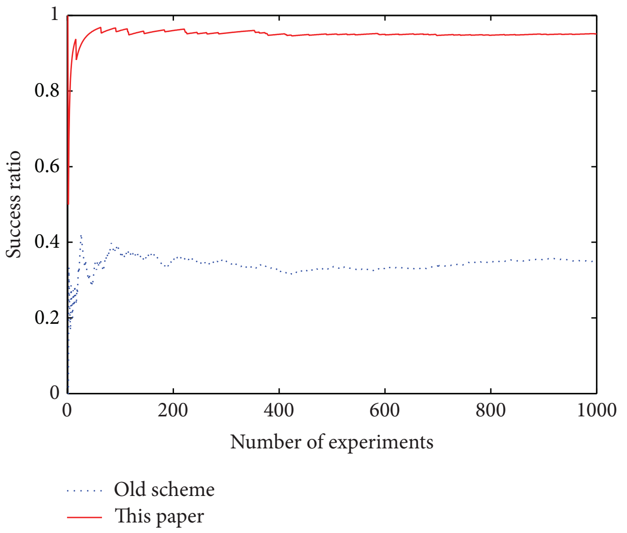


FIGURE 9: The relation between number of experiments and success ratio.

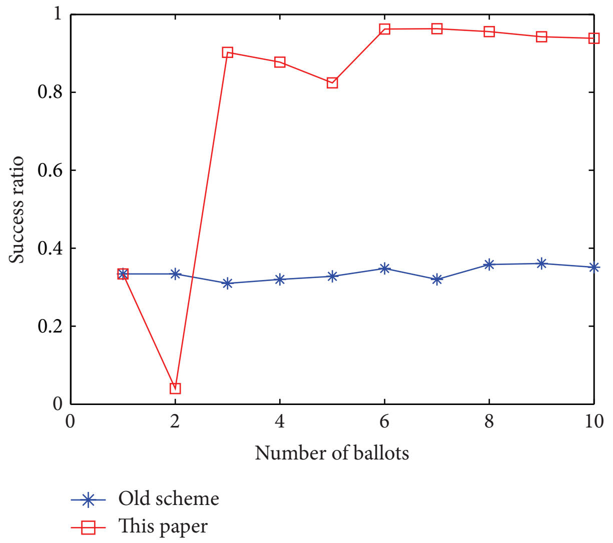


FIGURE 10: The relation between number of ballots and success ratio.

3.2.1. Theoretical Analysis. Under local noise, the keypoints voting test shows higher efficiency than binary comparison because the vote corresponding to a keypoint limits its influence on the collision distinguisher effectively. Intuitively, let $I_1, I_2, \ldots, I_{10}$ denote the information (with noise) of ten keypoints. Assume that I_i follows the normal distribution $N(I, \sigma^2)$ for $i = 1, 2, \ldots, 9$ and $I_{10} \sim N(I, \sigma'^2)$, where $\sigma' \gg \sigma$. In binary comparison, the information is accumulated as a collision/noncollision distinguisher which follows the normal distribution $N(10I, \sqrt{9\sigma^2 + \sigma'^2})$. In the case of $\sigma' \gg \sigma$, the distinguisher may show great errors. However in keypoints voting test, no matter how great the σ' is, the keypoint with great noise can cast only one vote. Therefore, the error of distinguisher will be decreased significantly.

3.3. Combined with Other Methods. As shown in the previous section, the keypoints voting test owns higher efficiency than binary comparison. However in fact, the two methods cost more traces than binary voting test because their averaging process wastes too much information. Fortunately, our keypoints voting test is multivariate, differential, and chosen plaintexts. So, it can improve some other collision attacks by being combined with them.

3.3.1. Improved Binary Voting Test. The keypoints voting can join the binary voting test [9] inherently because the former regards each point as a vote, and the latter only considers each pair of trace. So, the combined test may be called two-dimensional voting test. Figure 11 describes the flow chart of combined scheme. Intuitively, keypoint voting just substitutes $\Theta^{\mathrm{PV}}(\tau_1^i, \tau_2^i)$ for the function $\Psi^{\mathrm{BC}}(\tau_1^i, \tau_2^i)$ in the step

$$\Theta^{\mathrm{BV}}\left(\tilde{\tau}_1, \tilde{\tau}_2\right) = \sum_{i=1}^{m} \Psi^{\mathrm{BC}}\left(\tau_1^i, \tau_2^i\right) \quad (11)$$

of binary voting test.

3.3.2. Improved Correlation-Enhanced Collision Attack. The correlation-enhanced collision attack [7] compares the similarity between two sets of traces corresponding to two operations. The most similar case will result in the maximal correlation coefficient so that the most likely key guess can be gotten. According to the keypoints voting, multiple votes can be employed for multiple references of correlation coefficient, which is described in Figure 12. But the original correlation-enhanced attack only chooses the key corresponding to the maximal correlation coefficient for all the keypoints.

3.3.3. Efficiency Comparisons. To compare different methods further, we made some simulations in MATLAB for the binary voting test and correlation-enhanced collision attack with/without keypoints voting test. First, we generated 50000 traces, respectively, for two intermediate values x_1 and x_2. Each trace consisted of 30 keypoints, which followed the normal distribution $N(HW(x_i), \mathrm{sigma})$. Thus, after repeating the attacks for dozens of times, we could get their success rates. We show the relation between number of traces and success rate for binary voting test with/without keypoints voting test in Figure 13 and for correlation-enhanced collision attack with/without keypoints voting test in Figure 14.

4. Improved Ternary Voting Test

In Bogdanov's ternary voting test, each reference trace seems to be a judge who executes a decision algorithm by the standard of itself. However, this standard contains noise, which is unqualified. What is more, there are so many judges that the algorithm is inefficient. In this section, we discuss this problem.

4.1. Basic Idea. Our improved attack first reduces all the reference traces to a small number of "standard" ones with

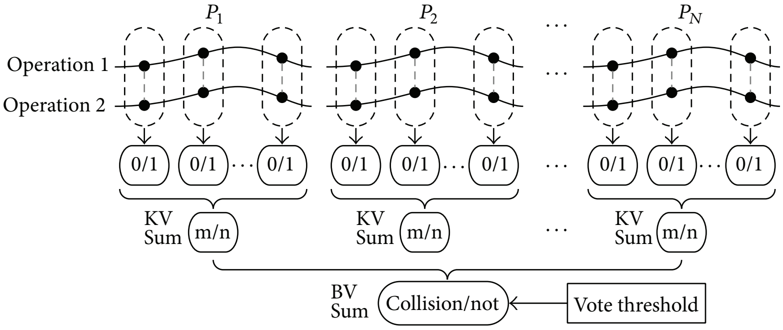


FIGURE 11: Flow chart of combination between keypoints voting test and binary voting test.

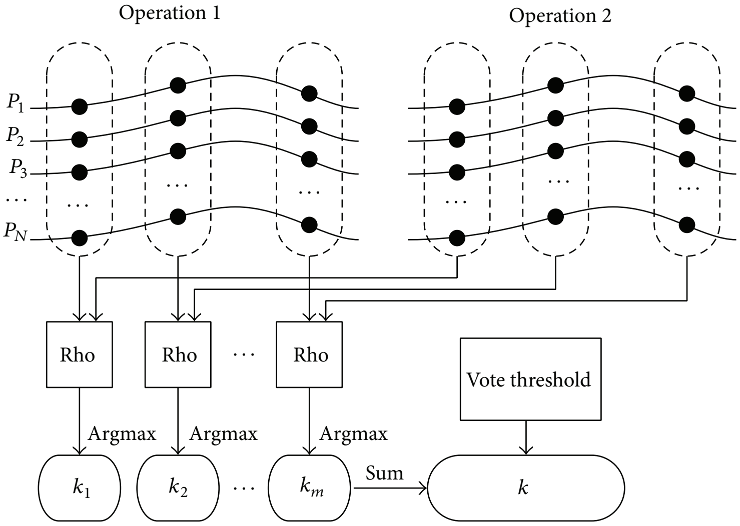


FIGURE 12: Flow chart of combination of keypoints voting test and correlation-enhanced collision attack.

very low noise. Then they are employed for estimating the collision of two traces.

The collision between two S-boxes of AES can be taken, for example. Because of the 8-bit input, the number of reference trace should be set to 256. In the stage of reference traces generation, 256 different plaintext bytes corresponding to the same S-box are input into the device, respectively. Each plaintext byte is encrypted for m' times, and the m' traces are averaged. So, 256 reference traces are acquired, which are denoted by $\overline{\tau}_i,\ i = 0, 1, \dots, 255$. If m' is big enough, the noise will be negligible.

In online stage, m traces are acquired corresponding to operations 1 and 2, respectively. Let τ_1 and τ_2 denote the two averaged traces.

In voting stage, for each reference trace $\overline{\tau}_i$, binary comparison is carried out first:

$$F(\tau_1, \tau_2, \overline{\tau}_i) = \Psi^{\mathrm{BC}}(\tau_1, \overline{\tau}_i) \cdot \Psi^{\mathrm{BC}}(\tau_2, \overline{\tau}_i). \tag{12}$$

When i traverses from 0 to 255, the total vote can be summed:

$$\Theta^{\mathrm{ITV}}(\tau_1, \tau_2) = \sum_{i=0}^{255} F(\tau_1, \tau_2, \tau_i). \tag{13}$$

Then, the collision can be decided according to the following threshold. The whole process is described in Figure 15:

$$\Psi^{\mathrm{ITV}}(\tau_1, \tau_2) = \begin{cases} 0 \ \ (\text{noncollision}), & \text{if } \Theta^{\mathrm{ITV}}(\tau_1, \tau_2) = 0, \\ 1 \ \ (\text{collision}), & \text{if } \Theta^{\mathrm{ITV}}(\tau_1, \tau_2) = 1, \\ -1 \ \ (\text{error}), & \text{if } \Theta^{\mathrm{ITV}}(\tau_1, \tau_2) > 1. \end{cases} \tag{14}$$

Remark. $\Theta^{\mathrm{ITV}}(\tau_1, \tau_2) > 1$ may mean that the threshold of least square method is too loose. Sometimes, the noise of reference traces may cause this problem. Therefore, more reasonable parameters should be chosen.

4.2. Efficiency Comparison.

We discuss two efficiency comparisons for evaluating our new attacks in this section.

4.2.1. Comparing Improved Ternary Voting Test with Ternary Voting Test.

In the stage of reference traces generation, both ternary voting test with profiling and our improved test acquire N^{TV} traces. But average is not employed by the old method, while the improved one executes an average for every m' traces (let $N^{\mathrm{TV}} = 256m'$). For the ternary voting test without profiling, the $160m$ reference traces are from the

Table 1: Comparison between three ternary voting tests.

Scheme	Reference traces generation	Online	Voting test
Ternary voting without profiling [9]	0	$m(C_{\text{acquire}} + C_{\text{average}})$	$160mC_{\text{vote}}$
Ternary voting with profiling [9]	$256m'C_{\text{acquire}}$	$m(C_{\text{acquire}} + C_{\text{average}})$	$256m'C_{\text{vote}}$
This paper	$256m'C_{\text{acquire}} + 256m'C_{\text{average}}$	$m(C_{\text{acquire}} + C_{\text{average}})$	$256C_{\text{vote}}$

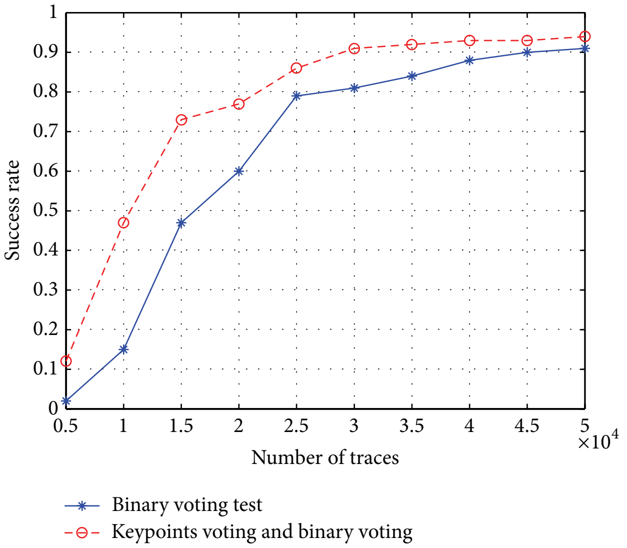


Figure 13: The relation between number of traces and success rate for binary voting test with/without keypoints voting test.

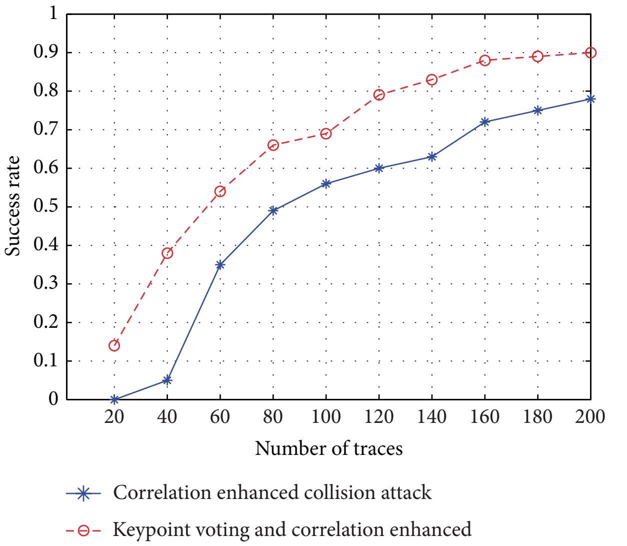


Figure 14: The relation between number of traces and success rate for correlation-enhanced collision attack with/without keypoints voting test.

m traces in online stage (one completed AES trace includes 16 × 10 sections corresponding to 16 S-boxes in 10 rounds). So, in its first stage, no reference traces are acquired.

In online stage, all three methods have the same operations. In voting stage, ternary voting test without/with profiling and our improved scheme carry out $160m$, N^{TV}, and 256 judgments, respectively, from their corresponding referees.

Assume that the complexity of acquiring a trace, averaging m traces, and a judgment is C_{acquire}, mC_{average}, and C_{vote}, respectively. Table 1 shows the complexity comparison of three methods. The complexity of the old method without/with profiling is $(160m - 256)C_{\text{vote}} - 256m'(C_{\text{acquire}} + C_{\text{average}})$ and $256(m' - 1)C_{\text{vote}} - 256m'C_{\text{average}}$ greater than the new one. In a high-performance oscilloscope, average is usually executed by hardware, whose complexity is negligible. Even if average is executed by computer, $C_{\text{vote}} \gg C_{\text{average}}$ also holds. Moreover, $m \gg m'$ usually. Therefore, our method is more efficient than the old ones.

4.2.2. Comparing Improved Ternary Voting Test with Binary Comparison. Let $\tau_1 = (\tau_{1,1}, \ldots, \tau_{1,n}) \in R^n, \tau_2 = (\tau_{2,1}, \ldots, \tau_{2,n}) \in R^n$ denote the two averaged traces to be decided, in which every point $\tau_{i,j}$ can be expressed as $\tau_{i,j} = s_{i,j} + \sigma_{i,j}$. Here $s_{i,j}$ means the ordinate value without noise, and $\sigma_{i,j}$ is a Gaussian noise whose expectation and variance are 0 and σ_R^2, respectively. Furthermore, we assume that $s_{i,j}$ forms a tolerance of Δs arithmetic progression when the input of S-box traverses from 0 to 255.

When collision takes place, the Euclidean distance from binary comparison follows noncentral chi-squared distribution [20]. If enough keypoints are chosen, it follows normal distribution:

$$\text{Distance}_{\text{BC}} \sim N\left(2n\sigma_R^2, 8n\sigma_R^4\right). \tag{15}$$

According to the three-sigma rule [5], this distance lies within the range of $(2n\sigma_R^2 - 6\sqrt{2n}\sigma_R^2, 2n\sigma_R^2 + 6\sqrt{2n}\sigma_R^2)$ with very high probability.

Similarly known, in improved ternary voting test, the Euclidean distance between τ_1 and the reference trace $\overline{\tau}_1$ which is nearest to τ_1 follows normal distribution:

$$\text{Distance}_{\text{TV}} \sim N\left(n\sigma_R^2 + n\sigma_R'^2, 2n\sigma_R^4 + 2n\sigma_R'^4\right). \tag{16}$$

Here σ_R' denote the standard deviation of standard reference trace after being averaged by m' traces. If the standard deviation of original trace is σ_R'', then we have $\sigma_R' = \sigma_R''/\sqrt{m'}$ [9]. After being averaged by enough traces, that is, $m' \to \infty$, $\text{Distance}_{\text{TV}} \sim N(n\sigma_R^2, 2n\sigma_R^4)$. So this distance lies within the range of $(n\sigma_R^2 - 3\sqrt{2n}\sigma_R^2, n\sigma_R^2 + 3\sqrt{2n}\sigma_R^2)$ with probability of almost 1.

Both methods employ least square method and their noise follows the same distribution, so the same threshold

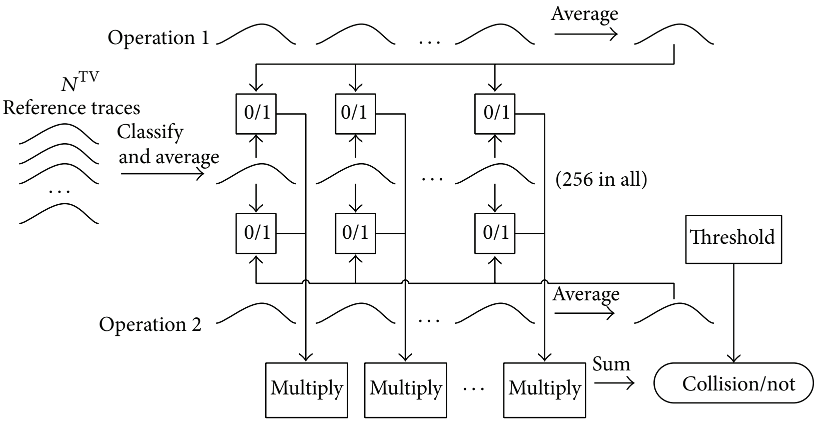


Figure 15: Flow chart of improved ternary voting test including only 256 votes.

should be chosen for collision detection. If we choose $2n\sigma_R^2 + 6\sqrt{2n}\sigma_R^2$ as threshold, the collision detection criterion of binary comparison will seem too loose, which is undesirable. According to the three-sigma rule, we suggest $n\sigma_R^2 + 3\sqrt{2n}\sigma_R^2$ as threshold, which can decide the collision more accurately. Therefore, our new method is more efficient than binary comparison.

Furthermore, we should discuss the case of false positives due to a large threshold when noncollision happens. Assuming two reference traces $\overline{\tau}_1$ and $\overline{\tau}_2$ are adjacent, the range of distance between τ_1 and $\overline{\tau}_2$ is

$$\left(\sigma_R^2\left(n+\frac{n\Delta s^2}{\sigma_R^2}-\frac{3\sqrt{2n}}{\sigma_R}\sqrt{\sigma_R^2+2\Delta s^2}\right),\right.$$
$$\left.\sigma_R^2\left(n+\frac{n\Delta s^2}{\sigma_R^2}+\frac{3\sqrt{2n}}{\sigma_R}\sqrt{\sigma_R^2+2\Delta s^2}\right)\right). \tag{17}$$

In order to avoid false positives, we must have

$$n\sigma_R^2+3\sqrt{2n}\sigma_R^2<\sigma_R^2\left(n+\frac{n\Delta s^2}{\sigma_R^2}-\frac{3\sqrt{2n}}{\sigma_R}\sqrt{\sigma_R^2+2\Delta s^2}\right). \tag{18}$$

In our practical experiments, we chose $n = 18$. So it can be simplified further to

$$\sigma_R^2-\Delta s^2+\sigma_R\sqrt{\sigma_R^2+2\Delta s^2}<0. \tag{19}$$

Assuming $l = \sigma_R/\Delta s$, we have $0 < l < 0.5$. Therefore, when m averages are executed such that the noise is reduced to $\sigma_R < \Delta s/2$, collision can be decided correctly with high probability.

In our experiment, the standard deviation of original traces $\sigma_R'' \approx 5\Delta s$. For two inputs of S-box and two groups of traces (each group included 200 traces), we executed improved ternary voting test and binary comparison, respectively. Figure 16 shows the relation between the success ratio of collision detection and number of averaged traces. Obviously, in our attack, only 100 traces can ensure that the error takes place with negligible probability, which is about 1/4 of the requirement of traditional method.

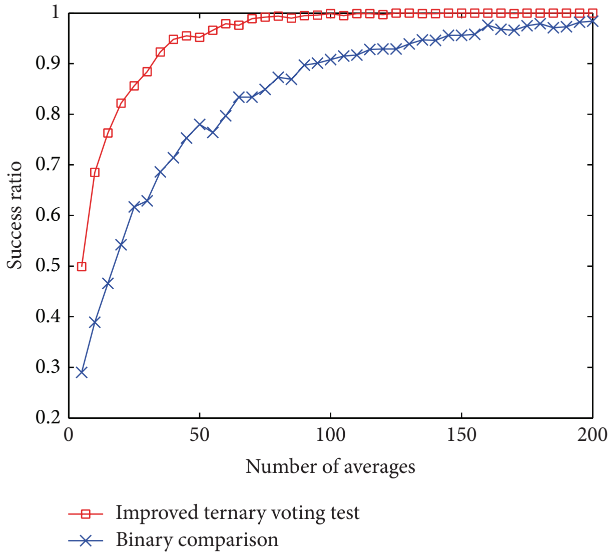


Figure 16: The relation between the success ratio of collision detection and number of averaged traces.

Remark. The improved ternary voting test can be combined with keypoints voting test. Specifically, every trace in the ternary voting test can be divided into n votes. Then a decision from a reference trace is replaced by n votes, and the threshold can be set to n. The combined method possesses better applicability for real environment and can overcome more problems such as local noise and inefficiency.

5. Discussions of Countermeasures

The attacks presented in this paper defeat the countermeasure of generating Gaussian noise. However, we think there are some countermeasures against our attacks.

(i) Random delays: the traditional countermeasure of random delays tries to complicate data alignment.

So these delays are inserted into the cryptographic operation either by special state machines or nondeterministic processors [21].

(ii) Dummy rounds: the AES algorithm includes ten rounds, but it can also be implemented by the way of more than ten rounds. Some dummy rounds which run the same operation and random operands can join the whole encryption from a random position. As a result, the attacker will get some invalid information with high probability due to the confusion of the dummy rounds.

(iii) Masking: the technology of masking [22] makes the power consumption of the cryptographic device independent of the intermediate values of the cryptographic algorithm by randomizing the intermediate values that are processed by the cryptographic chip. So, it can resist first-order collision attack completely.

Unfortunately, all these countermeasures cannot resist various side-channel attacks completely but just increase their difficulty.

6. Conclusions

In this paper, we propose keypoints voting test, improve ternary voting test, and discuss their countermeasures. According to our experiments, the two new methods show higher success ratio and efficiency than traditional attacks. In fact, the collision detection technologies can be used not only for collision attack of cryptographic devices, but also for all the situations in which the equality of some parameters needs to be verified, such as template attack [23] of power analysis and fault detection of differential fault analysis [24]. Therefore, our methods show broad applied values.

Collision attacks are usually appropriate to software since the variables are bytes and thus are more likely to be equal. However, due to the features on multivariant, keypoints voting test is also suitable for hardware implementation. Under the correct circumstances, more points in a trace can be studied for higher signal-to-noise ratio.

The voting test only discusses how to detect a collision fast. This kind of collision detection methods can be combined with other collision attack frameworks such as the unified and optimized linear collision attacks [8] so that collision attack can be mounted more efficiently.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the 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) and the Foundation of Science and Technology on Information Assurance Laboratory (no. KJ-13-101). Besides, the authors are grateful to the reviewers and the editor that contributed to the great improvement of the original version of this paper with their valuable comments and suggestions.

References

[1] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering: Design Principles and Practical Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2010.

[2] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, Fla, USA, 1997.

[3] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Advances in Cryptology, CRYPTO ’99*, vol. 1666 of *Lecture Notes in Computer Science*, pp. 388–397, Springer, Heidelberg, Germany, 1999.

[4] K. Schramm, T. Wollinger, and C. Paar, “A new class of collision attacks and its application to DES,” in *Fast Software Encryption, FSE ’03*, vol. 2887 of *Lecture Notes in Computer Science*, pp. 206–222, Springer, Heidelberg, Germany, 2003.

[5] A. A. Sveshnikov and R. Silverman, *Problems in Probability Theory, Mathematical Statistics and Theory of Random Functions*, Dover Publications, New York, NY, USA, 1979.

[6] A. Bogdanov, “Improved side-channel collision attacks on AES,” in *Selected Areas in Cryptography, SAC ’07*, vol. 4876 of *Lecture Notes in Computer Science*, pp. 84–95, Springer, Heidelberg, Germany, 2007.

[7] A. Moradi, O. Mischke, and T. Eisenbarth, “Correlation-enhanced power analysis collision attack,” in *Cryptographic Hardware and Embedded Systems, CHES ’10*, vol. 6225 of *Lecture Notes in Computer Science*, pp. 125–139, Springer, Heidelberg, Germany, 2010.

[8] B. Gérard and F. X. Standaert, “Unified and optimized linear collision attacks and their application in a non-profiled setting,” in *Cryptographic Hardware and Embedded Systems, CHES ’12*, vol. 7428 of *Lecture Notes in Computer Science*, pp. 175–192, Springer, Heidelberg, Germany, 2012.

[9] A. Bogdanov, “Multiple-differential side-channel collision attacks on AES,” in *Cryptographic Hardware and Embedded Systems, CHES ’08*, vol. 5154 of *Lecture Notes in Computer Science*, pp. 30–44, Springer, Heidelberg, Germany, 2008.

[10] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*, Cambridge University Press, 2nd edition, 2009.

[11] T. Guneysu and A. Moradi, “Generic side-channel countermeasures for reconfigurable devices,” in *Cryptographic Hardware and Embedded Systems, CHES ’11*, vol. 6917 of *Lecture Notes in Computer Science*, pp. 33–48, Springer, Heidelberg, Germany, 2011.

[12] E. Brier, C. Clavier, and F. Olivier, “Correlation power analysis with a leakage model,” in *Cryptographic Hardware and Embedded Systems, CHES ’04*, vol. 3156 of *Lecture Notes in Computer Science*, pp. 16–29, Springer, Heidelberg, Germany, 2004.

[13] J. Daemen and V. Rijmen, *The Design of Rijndael*, Springer, Berlin, Germany, 2002.

[14] J. -S. Coron and I. Kizhvatov, “Analysis and improvement of the random delay countermeasure of CHES 2000,” in *Cryptographic Hardware and Embedded Systems, CHES ’10*, vol. 6225 of *Lecture Notes in Computer Science*, pp. 95–109, Springer, Heidelberg, Germany, 2010.

[15] H. Kim, S. Hong, and J. Lim, “A fast and provably secure higher-order masking of AES S-box,” in *Cryptographic Hardware and*

Embedded Systems, CHES 2010, vol. 6917 of *Lecture Notes in Computer Science*, pp. 95–107, Springer, Heidelberg, Germany, 2011.

[16] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE '04)*, vol. 2004, pp. 246–251, IEEE Computer Society, February 2004.

[17] E. Prouff and M. Rivain, "A generic method for secure S-box implementation," in *Workshop on Information Security Applications, WISA '07*, vol. 4867 of *Lecture Notes in Computer Science*, pp. 227–244, Springer, Heidelberg, Germany, 2008.

[18] X. Yang, Z. Li, A. Wang, and S. Wen, "Design research of the des against power analysis attacks based on FPGA," *Microprocessors and Microsystems*, vol. 35, no. 1, pp. 18–22, 2011.

[19] Altera, Product Specification: Cyclone FPGA Family Data Sheet, October 2003, http://www.altera.com.

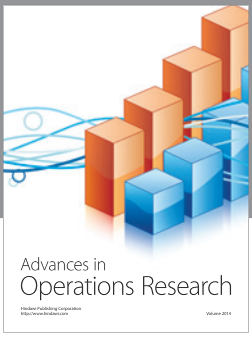
[20] A. Bogdanov and I. Kizhvatov, "Beyond the limits of DPA: combined side-channel collision attacks," *IEEE Transactions on Computers*, vol. 61, no. 8, pp. 1153–1164, 2012.

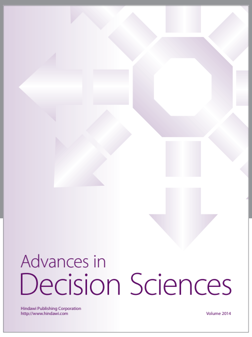
[21] J. Irwin, D. Page, and N. P. Smart, "Instruction stream mutation for non-deterministic processors," in *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP '02)*, pp. 286–295, IEEE Computer Society, Los Alamitos, Calif, USA, 2002.

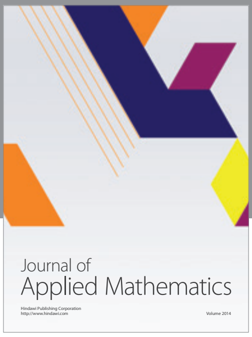
[22] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer, Heidelberg, Germany, 2007.

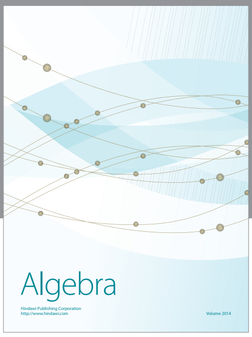
[23] S. Chari, J. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems, CHES '02*, vol. 2523 of *Lecture Notes in Computer Science*, pp. 13–28, Springer, Heidelberg, Germany, 2003.

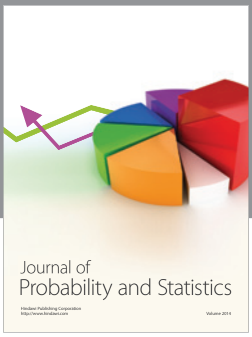
[24] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," in *Advances in Cryptology, CRYPTO '97*, vol. 1294 of *Lecture Notes in Computer Science*, pp. 513–525, Springer, Heidelberg, Germany, 1997.


Advances in
Operations Research

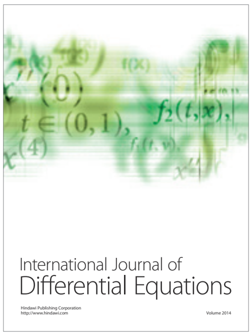

Advances in
Decision Sciences


Journal of
Applied Mathematics

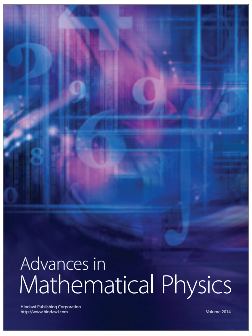

Algebra

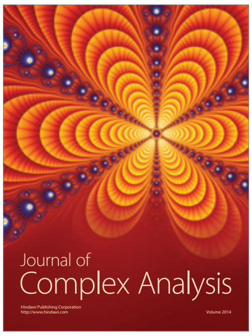

Journal of
Probability and Statistics

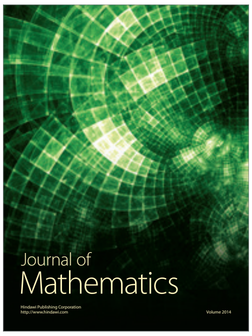
The Scientific
World Journal

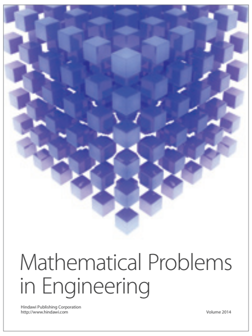

International Journal of
Differential Equations

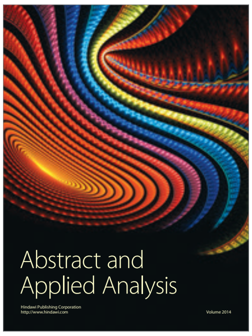
International Journal of
Combinatorics

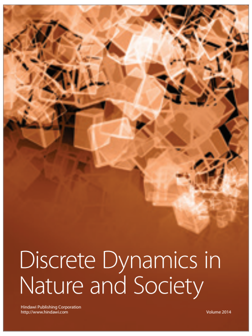

Advances in
Mathematical Physics

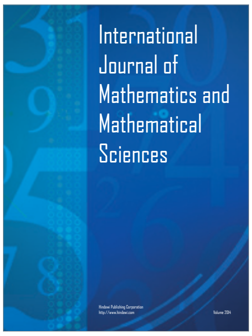

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

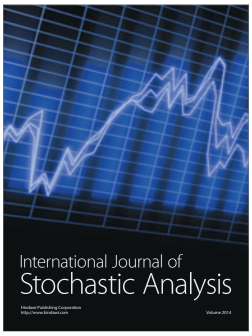

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization