Hypothesis Testing Based Side-channel Collision Analysis

DONG ZHENG, XIANG JIA, AND MEILING ZHANG
National Engineering Laboratory for Wireless Security, Xi’an University of Posts and Telecommunications, Xi’an 710121, China
Corresponding author: Meiling Zhang (e-mail: zhangmlwy@126.com).

This work was supported by the National Natural Science Foundation of China under Grant 61772418 and Grant 61602378.

ABSTRACT Side-channel collision analysis has become a research hotspot since its first publication in 2003. Compared with differential power analysis (DPA) and correlation power analysis (CPA), collision analysis does not need to know the intermediate value and does not depend on a specific power model. Collision analysis includes collision detection phase and key recovery phase. After collisions have been detected, the cryptographic key can be recovered from these collisions using such cryptanalytic techniques as linear collision attack and algebraic collision attacks. Therefore, the success rate of the entire attack depends largely on the success rate of collision detection.

In this paper, we propose an efficient collision detection method, which combines side channel information with hypothesis testing in statistics. With the traces gathering by repeatedly encrypting a certain plaintext that is unnecessary to be known, we are able to evaluate the cumulative distance between two collision S-boxes, then define a null hypothesis \( H_0 \) and calculate the rejection region accordingly. Given some traces collected from repeatedly encrypting any identical plaintext, collisions can be detected by determining whether the cumulative distance between the corresponding two S-boxes falls into the rejection region or not. We compare our work with those related to side channel collision attack in recent years from perspective of four hot issues. Comparing with current methods, our proposed collision detection method is simpler and more feasible. Furthermore, the experiments show that the more accurate the constant power is, the higher the success rate of collision detection is.

INDEX TERMS side-channel collision analysis, detect collisions, hypothesis testing, AES

I. INTRODUCTION

Artificial Intelligence (AI) technology has been rapidly applied in many fields including medical, commercial, communication, and urban management. Tsinghua University has developed a computational chip for reconfigurable neural networks; Huawei first used artificial intelligence mobile chips in global mobile phones [1]. However, security threats in AI chips, especially side channel analysis, should not be underestimated. Side-channel analysis was introduced by Kocher in the 1990s [2]. Since then, for more than twenty years, the research of side-channel attacks and defense countermeasures has become an important branch in cryptography. Power analysis attack is one of the most important and effective means of side-channel attacks. In 2004, Brier et al. [3] formalized the differential power analysis (DPA) [4] and provided a statistical way to associate a leakage model with power traces thanks to the Pearson correlation factor. In the presence of a masking countermeasure, high order techniques are proposed [5]–[7]. Nowadays, a growing number of methods are being applied to side-channel attacks, such as collision attack, mutual information analysis [8], and linear regression attack [9], etc.

Different from DPA, side-channel collision analysis is concern about whether there is a collision in the process of encryption or decryption. For block ciphers, it is necessary to find a process involving partial key operations, the outputs of which may be the same for different inputs. Previous collision attacks are mainly associated with the surjective mapping of a hash function, such as MD4 [10]. Schramm et al. first proposed a side-channel collision attack on block cipher DES [11]. They showed that it is feasible to find collisions at the output of a S-box, as S-box operation is surjective but not injective. They improved the efficiency of their attack mainly in two aspects. On one hand, the possibility of collisions can be improved by increasing the number of collision detections. On the other hand, the number of candidate secret keys can be
reduced by choosing plaintexts after a collision occurs. Later,
[12] applies the technique to block cipher AES. For the S-box
operation is bijective in AES, collisions are detectable in the
output bytes of MixColumn operation in the first round. In
[13] and [14], collision attacks are applied to a masked S-box
implementation. Moradi et al. discovered a leakage inside
the masked circuits due to uncontrolled hardware glitches
and implemented attacks. Clavier et al. utilized the re-use
of masks to build the relation of masked data in various S-
box operations. Linear collision attacks [15] and algebraic
collision attacks [16] for AES have been proposed which
require fewer measurements for the key recovery procedure.
All the above attacks work under the assumption that one-
byte collisions are detectable.

For methods of detecting internal collisions, most papers
on collision attacks utilize the Euclidean distance between
two traces. It is oblivious that the distance will be higher
for non-collisions and lower for collisions, but the actual
results are affected by electronic noise. In [17], the effi-
ciency of collision detection can be significantly improved
by reducing the dimension of side-channel traces based on
the properties of the Euclidean distance. In [13] and [18],
Pearson correlation factor is used to detect collisions. In [12],
continuous wavelet analysis is also investigated to achieve
better collision detection but its computational costs are
much higher compared with least-square method. In order
to make collision detection more reliable, the techniques of
binary voting and ternary voting are proposed in [19], and
attacked can apply voting to select correct collisions.

In this paper, a hypothesis testing based collision detection
 technique is proposed, which requires a very low number of
measurements for each plaintext but succeeds with a high
probability, and practical results on real power traces are also
presented. In order to count the success rate of our method,
we carry out 50,000 experiments and conduct a total of 6 mil-
lion collision detections. Two factors that affect the success
rate of detection have also been verified. The experimental
results show that the accuracy of collision detection of this
technique reaches 98.77% with only 11 measurements for
each same plaintext. For current methods to detect collision,
the main way to determine collision is traversing plaintexts or
setting threshold. The method of traversing plaintexts needs
more power traces, while the method of setting threshold
needs to estimate the decision threshold. Our technique is
to determine collision by setting threshold, which can be
well estimated. It has advantages over binary voting in some
aspects. Binary voting test requires two thresholds, but how
to estimate them accurately is not mentioned in [19]. For
each collision detection, the binary voting test requires a
voting process, but our scheme only requires a hypothesis
test while the null hypothesis is easy to construct. Moreover,
the technique is also suitable for other embedded devices of
block ciphers.

The remainder of the paper is organized as follows: Sec-
tion II introduces the basic knowledge of hypothesis testing,
two types of errors in hypothesis testing, and briefly mentions
liner collision attacks. Section III presents characteristics
of single point in power trace and the proposed hypothesis
testing based collision detection technique. In section IV, we
demonstrate our attacks and practical results on the AES
software implementation in both unprotected and protected
cases. We conclude this paper in Section V.

II. PRELIMINARIES
The proposed collision attack consists of collision detection
phase and key recovery phase. In the collision detection
phase, hypothesis testing is used to determine whether a
collision occurs. In the phase of key recovery, the method of
linear collision attacks is used to recover the entire encryption
key.

A. HYPOTHESIS TESTING
Significance-based hypothesis testing determines whether a
specified hypothesis can be accepted. Now suppose that \( \mu \)
is the mean of some sample values. To test whether the \( \mu \)
is equal to a certain value \( \mu_0 \), two assumptions are set up.
One is \( H_0 : \mu = \mu_0 \) called null hypothesis, the other is
\( H_1 : \mu \neq \mu_0 \) called alternative hypothesis. Then follow some
teps to test whether the null hypothesis is true at a predefined
significance level. To be more specific, let the above sample
values are gathered from a normal random variable \( X \) with
mean \( \mu \) and variance \( \sigma^2 \), i.e. \( X \sim N(\mu, \sigma^2) \).

Estimate \( \mu \) with sample mean \( \bar{X} \) to verify whether \( \mu \) is
equal to \( \mu_0 \). Given a carefully chosen constant \( c \), if \( \bar{X} - \mu_0 > c \),
then we reject \( H_0 \), i.e. \( \mu = \mu_0 \); otherwise accept it. In
one-tail test, let the significance level be \( \alpha \), then we have

\[
 p(\bar{X} - \mu > c) = \alpha. \tag{1}
\]

According to (1), the relationship between \( c \) and \( \alpha \) is
shown as follows.

\[
 c = \frac{\sigma}{\sqrt{n}} \cdot z_{1-\alpha}. \tag{2}
\]

Where \( n \) represents the number of the sample, \( z_{1-\alpha} \) is the
upper \((1-\alpha)\) point of standard normal distribution, and then
the rejection region of \( \mu_0 \) can be given by

\[
 \left( \bar{X} + \frac{\sigma}{\sqrt{n}} \cdot z_{1-\alpha/2}, \infty \right). \tag{3}
\]

The proposed method for collision detection consists four
steps:

- analyzing the cumulative distance distribution when
  colliding.
- setting the rejection region.
- calculating the cumulative distance between any two
  chosen S-boxes.
- testing whether the calculating result falls into the rejec-
  tion region or not.
B. TWO TYPES OF ERRORS

In hypothesis testing, there are two types of errors that are inevitable.

(1) Type I error is the incorrect rejection of a true null hypothesis, the probability of which can be assumed as the significance level \( \alpha \).

(2) Type II error is the failure to reject a false null hypothesis, which is denoted by \( \beta \).

Then we have

\[
\begin{align*}
    & \Pr\{\text{reject } H_0 | H_0 \text{ is true}\} = \alpha \\
    & \Pr\{\text{accept } H_0 | H_0 \text{ is not true}\} = \beta.
\end{align*}
\]

Due to the limitations of sample information, it is bound to produce the both two types of errors. Such being the case, the guiding principle of hypothesis testing is to control the probability of type I error no more than \( \alpha \), and then reduce the probability \( \beta \) of type II error by some methods.

In this paper, the type I error means that a collision indeed occurs but is not detected, and the type II error means that no collision occurs but the test result is collision. Both types of errors affect the success rate of the attack.

C. LINER COLLISION ATTACKS

In [12], to detect collisions, Schramm et al. compared power traces corresponding to the instances of time when the outputs of MixColumn operation are processed in the second round. In [15], the notion of a generalized internal collision for AES was introduced. As the S-box operation is applied 160 times in a single AES run, this gives us about 40 generalized collisions. The number of generalized collisions grows quadratically with the increase of the number of AES executions.

Liner collision attacks mean solving large number of linear equations of collision set. When a linear collision occurs, the attacker can get a binomial linear equation over \( GF(2^8) \), such as \( k_{j_1} \oplus k_{j_2} = p_{j_1} \oplus p_{j_2} \) (\( i_1, i_2 \) stands for serial numbers of AES executions, and \( j_1, j_2 \) stands for serial numbers of different S-boxes).

In this paper, we use liner collision attacks to recover the AES key and only consider the collisions in the first round. It should be noted that an extra plaintext-ciphertext pair is required to select the correct key from the set of key candidates. A detailed theoretical analysis of linear collision attacks is given in [15].

III. COLLISION DETECTION TECHNIQUE

For collision detection, there are two important factors should be considered. One, in the first place, is the success rate of collision detection which affects the success rate of the entire attack. The other is complexity that quantifies the degree of difficulty in achieving the technique.

As is known to all, the least-square method is the most commonly method in collision attacks. It detects collisions by calculating the Euclidean distance of two real-valued traces. Collision detection with continuous wavelet analysis works best, but its computational costs are much higher. Binary voting test can detect collisions in the presence of noise.

In this section, we first introduce the composition of the power trace and the characteristics of the single point distribution. Then, the collision detection technique using hypothesis testing is proposed.

A. POWER TRACE SINGLE POINT CHARACTERISTIC

The power analysis attack utilizes the fact that the power consumption of the cryptographic device depends on the operations performed by the device and the processed data. For each single point in the power trace, the characterization of power of the point can be denoted as [20]:

\[
P_{\text{total}} = P_{\text{op}} + P_{\text{data}} + P_{\text{el.noise}} + P_{\text{const}},
\]

where the four factors \( P_{\text{op}}, P_{\text{data}}, P_{\text{el.noise}} \) and \( P_{\text{const}} \) denote operation dependent power, data dependent power, electronic noise and constant power, respectively. Note that all the four factors are functions of time and the values may differ for different points. In addition, the AES implementation in this paper satisfies the property that all S-box operations are implemented in a similar way, and \( P_{\text{el.noise}} \) follows a normal distribution with mean \( 0 \). According to (5), the distribution of \( P_{\text{total}} \) is similar to the distribution of \( P_{\text{el.noise}} \) when performing the same operation on fixed data. In power analysis attack, an attacker can recover secret key by analyzing \( P_{\text{op}} \) and \( P_{\text{data}} \), but the larger the \( P_{\text{el.noise}} \), the more difficult the attacker can get the key.

When collision occurs, i.e. \( P_{\text{op}_x} + P_{\text{data}_x} = P_{\text{op}_y} + P_{\text{data}_y} \) (\( x \) and \( y \) represent different segments of S-box operations on the power trace), it’s obvious that there are two conditions largely affecting the success rate of detecting collisions.

- (c1) Distance between \( P_{\text{el.noise}_x} \) and \( P_{\text{el.noise}_y} \): The smaller distance between \( P_{\text{el.noise}_x} \) and \( P_{\text{el.noise}_y} \), the closer distance between \( P_{\text{total}_x} \) and \( P_{\text{total}_y} \) will be.

- (c2) Distance between \( P_{\text{const}_x} \) and \( P_{\text{const}_y} \): Make sure \( P_{\text{const}_x} = P_{\text{const}_y} \), which is difficult to achieve for the constant power is related to the actual hardware implementation.

For condition (c1), it is well known that \( P_{\text{el.noise}} \) in most cryptographic devices obeys a normal distribution. The distance between \( P_{\text{el.noise}_x} \) and \( P_{\text{el.noise}_y} \) can be reduced to one over \( \sqrt{\lambda} \) by averaging technique, which measurements encryption of each plaintext \( \lambda \) times, then averages the traces collected and takes the corresponding results as the trace for this plaintext.

For condition (c2), it is not realistic to know whether \( P_{\text{const}_x} = P_{\text{const}_y} \) due to the lack of detailed implementation of the device. However, our experimental results show that the constant power for different encryption moments can be estimated by averaging the traces collected from encrypting some random (un)known plaintexts. In order to decrease the influence caused by \( P_{\text{const}_x} \neq P_{\text{const}_y} \), the constant power is subtracted for each point. By this means, collision detection
is more accurate. Therefore, in this paper, when a collision occurs during the S-box operations, we have

\[ P_{\text{total}_x} - P_{\text{total}_y} = P_{\text{el.noise}_x} - P_{\text{el.noise}_y}. \]  

Thus, \( P_{\text{total}_x} - P_{\text{total}_y} \) follows the same distribution as \( P_{\text{el.noise}_x} - P_{\text{el.noise}_y} \), i.e., a normal distribution.

In the case of non-collision, different data consume different data dependent power. So, it’s oblivious that the following formula holds.

\[ |P_{\text{data}_x} - P_{\text{data}_y}|_{\text{data}_x \neq \text{data}_y} > |P_{\text{data}_x} - P_{\text{data}_y}|_{\text{data}_x = \text{data}_y} = 0. \]  

Based on this attribute, the collision and the non-collision can be distinguished. As it is difficult to make a general statement of the data dependent power of a cryptographic device, the distance distribution in the non-collision case is much more complex than that in collision case, which makes the latter easier to analyze. The collision detection technique using hypothesis testing only considers the Euclidean distance distribution in the case of collision.

**B. HTCD: HYPOTHESIS TESTING COLLISION DETECTION**

In this subsection, we first analyze relevant parameters of the distance distribution in the case of collision, including mean value \( \mu \), constant \( c \) (as stated in (2)) and unilateral rejection region (as stated in (3)). Then, hypotheses are built according to the distance of traces of any two S-box operations, and the unilateral rejection region is used to determine which hypothesis \( H_0 \) or \( H_1 \) is accepted. If \( H_0 \) is has accepted, it means that a collision have occurred, otherwise has not. How to estimate these parameters in the case of collision will be discussed below.

First, for each plaintext, encrypt it \( n \) times to obtain a set of \( n \) power traces that are indexed by \( i \). Then, it is necessary to locate the 16 S-box operations of the first round in the trace by the means of variance detection. After locating the positions of S-boxes, the segment from the output of XOR operation to the output of S-box operation is intercepted for each S-box. Each segment contains \( J \) points that are indexed by \( j \) and is used to the following collision detection. For clarity, we use \( X \) and \( Y \) to denote two segments corresponding to two of the 16 S-boxes. And the subscripts of \( X_{ij} \) and \( Y_{ij} \) refer to the \( j^{th} \) point in the two segments that picked up from the \( i^{th} \) trace, where \( 1 \leq i \leq n \), and \( 1 \leq j \leq J \).

As mentioned in the previous subsection, the distribution of a single point in power traces follows a normal distribution. In the case that a collision happens between \( X \) and \( Y \), the experiments show that the mathematical expectation and variance of \( X_{ij} \) and \( Y_{ij} \) are almost the same. So the distributions of \( X_{ij} \) and \( Y_{ij} \) can be expressed as:

\[ X_{ij} \sim N(\mu_j, \sigma_j^2), \]
\[ Y_{ij} \sim N(\mu_j, \sigma_j^2). \]

For each measurement of the same plaintext, it is believed that the distribution of \( X_{ij} \) (\( Y_{ij} \)) is irrelevant to the index of traces, i.e., \( i \), thus:

\[ X_{ij} \sim N(\mu_j, \sigma_j^2), \]
\[ Y_{ij} \sim N(\mu_j, \sigma_j^2). \]

Let \( W_{ij} = X_{ij} - Y_{ij} \),

\[ W_{ij} = X_{ij} - Y_{ij} \sim N(0, 2\sigma_j^2). \]

Then, the cumulative distance \( V \) can be estimated as:

\[ V = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{J} (X_{ij} - Y_{ij})^2 \]
\[ = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{J} W_{ij}^2 \]
\[ = \frac{1}{n} \sum_{i=1}^{n} U_i, \]  

where \( U_i = \sum_{j=1}^{J} W_{ij}^2 \), for \( i \in [1, n] \).

Therefore, in the case of a collision, the mean and variance of \( U_i \) can be estimated by

\[ \begin{cases} E(U_i) \approx \sum_{j=1}^{J} 2\sigma_j^2 = \mu_0 \\ D(U_i) \approx 8 \left( \sum_{j=1}^{J} \sigma_j^2 \right)^2 = \sigma_0^2. \end{cases} \]  

According to the Central Limit Theorem, the sum of multiple independent and identical distributions obeys a normal distribution, i.e.

\[ V \sim N(\mu, \sigma^2), \]

where \( \mu = \mu_0, \sigma^2 = \sigma_0^2/n \). After completing the estimation of the cumulative distance distribution in the case of a
In the case of collision, it is necessary to determine the \( n \) required for the same plaintext. As can be seen from the foregoing discussion, given the target probability \( \alpha \) and the constant \( c \), it can be determined by the (10) derived from Figure 2.

\[
P(V - \mu > c) = \alpha
\]

Rewrite (10) as

\[
P\left( \frac{V - \mu}{\sigma} > \frac{c}{\sigma} \right) = \alpha.
\]

According to the definition of the quantile, \( n \) can be deduced,

\[
c = \frac{\sqrt{n} \cdot c}{\sigma_0} = z_{1-\alpha},
\]

\[
n = \frac{\sigma_0^2 \cdot z_{1-\alpha}^2}{c^2}.
\]

In turn, given \( n \) and \( \alpha, c \) can be estimated by equation (13):

\[
c = \frac{\sigma_0 \cdot z_{1-\alpha}}{\sqrt{n}}.
\]

Note that the quantile \( z_{1-\alpha} \) can be looked up in the standard normal distribution quantile table.

After completing the estimation of the relevant parameters in the case of collision, we can get the unilateral rejection region \( (\mu + c, \infty) \) and the corresponding probability of false rejection is \( \alpha \). The unilateral rejection region is the core of the entire hypothesis testing, and greatly affects the accuracy of collision detection. For each plaintext, the number of collision detection tests is 120, i.e. combinatorial number \( C_{120}^{2} \). Let \( d_q \) \(( q \in [1, 120] )\) represent the corresponding 120 distance values \( V \) in (8).

As shown in Figure 3, for each \( d_q \), we formulate the null hypothesis \( H_0: d_q = \mu \) and the alternative hypothesis \( H_1: d_q \neq \mu \), and then assess the truth of \( H_0 \). If \( d_q \) falls into the unilateral rejection region, \( H_0 \) is rejected which means that a collision does not happen between the two S-boxes corresponding to \( d_q \). Then repeat hypothesis testing for other \( d_q \) until enough different collisions are found to recover the encryption key.

According to the birthday paradox, the probability that a collision happens reaches 50% with 19 \( d_q \)-tests and 99.5% with 50 \( d_q \)-tests, while the least-square method using the Euclidean distance requires 256 measurements to detect collision.

As shown in Chapter 3 of [15], the expected number of generalized collisions within the first round after 8 measurements is around 28.12. From these generalized collisions, the number of connected components is around 1.43 and the success probability of recovering the entire key can reach 99.90%.

IV. PRACTICE ATTACK

In this section, practical attacks are performed on the platform of ChipWhisperer in order to verify the efficiency of HTCD in the actual attack. ChipWhisperer is an open-source platform for hardware embedded security research [21]. It integrates measurement equipment, target device, capture software and attack software. It has benefits for students and researchers who have a low-cost laboratory. In our experiments, the version we are using is CW1173 ChipWhisperer-Lite, equipped with SAKURA-G experiment board and a CW303 XMEGA as the target test board. The attacks are presented on a well-known software AES implementation written in C. The internal clock generator is running at a frequency of 7.37MHz, and we only sample the power in the first round with 3000 sample points.

A. ATTACK PROCEDURE

The entire attack procedure consists of the online acquisition phase and the offline attack phase. The power traces collected in the online acquisition phase fall into three parts: (Part1) power traces for estimating constant power consumption, (Part2) power traces for estimating the rejection region, and (Part3) power traces of attack. Among them, the power traces of Part1 are collected from encrypting some different plaintexts that is unnecessary to be known. And these traces can be reused to locate the 16 S-box operations in the first round by variance detection. According to (9) and (14), the parameter, i.e. \( \mu \) and \( \sigma^2 \) that are used to estimate the rejection region and related to power traces, can be derived by variance of each point in the traces. Therefore, the traces of Part2 should be gathered from repeatedly encrypting an identical (un)known plaintext. However, for the power traces of attack, the corresponding plaintext must be clear. The traces of Part3 consist of \( \{\tau_i\}_{i=1,...,n,T=1,...,T} \) which are gathered from measuring \( n \) encryptions for each of \( T \) different plaintexts.

In the offline attack phase, we locate the 16 S-box operations of the first round in the power trace, estimate the constant power, calculate variance of each point to determine the unilateral rejection region of collision, and then assess the truth whether the chosen two S-boxes collide, and finally recover the encryption key.

Since implementations of the 16 S-box operations in the first round are similarly, the positions of the 16 S-box operations in the power trace can be clearly distinguished, as shown in Figure 4, by using variance detection [13]. We
Given two traces $\tau$ determine the unilateral rejection region follows (9) and (14).

Part traces in $\tau$ are averaged to estimate the constant power trace. All power traces of $\tau$ minus the constant one are new traces for $\tau$. Meanwhile, the traces of $\tau$ from Part $1$ and $2$ are intercepted segments from the write down the point before S-box and the point after S-box, both of which are used to intercept segments from the traces of Part $2$ and Part $3$. Meanwhile, the traces of Part $1$ are averaged to estimate the constant power trace. All power traces in Part $2$ minus the constant one are new traces for $\tau$ to determine the unilateral rejection region follows (9) and (14).

B. ATTACK ON UNPROTECTED S-BOXES OF AES

Given two traces $\tau_{s_1, t_1} = (\tau_{s_1, 1}, \cdots, \tau_{s_1, t_1}) \in \mathbb{R}^J$ and $\tau_{s_2, t_2} = (\tau_{s_2, 1}, \cdots, \tau_{s_2, t_2}) \in \mathbb{R}^J$ that are intercepted from $\tau_{s_1, t_1}$ and $\tau_{s_2, t_2}$, and correspond to S-box $s_1$ and $s_2$, respectively. Let $Col\{d(\tau_{s_1, 1}, \tau_{s_2, 1})\}$ denote a decision function returning 0 (non-collision) or 1 (collision). The decision function can be expressed as:

$$Col\{d(\tau_{s_1, 1}, \tau_{s_2, 1})\} = \begin{cases} 0(\text{non-collision}), & \text{if } d(\tau_{s_1, 1}, \tau_{s_2, 1}) > \mu + c \\ 1(\text{collision}), & \text{if } d(\tau_{s_1, 1}, \tau_{s_2, 1}) \leq \mu + c \end{cases}$$

(15)

where $d(\tau_{s_1, 1}, \tau_{s_2, 1})$ is a function to calculate the cumulative distance, $\mu + c$ is the lower bound of the rejection region, and $t_1, t_2 \in [1, T]$. $d(\tau_{s_1, 1}, \tau_{s_2, 1})$ can be expressed as:

$$d(\tau_{s_1, 1}, \tau_{s_2, 1}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{J} (\tau_{s_1, i,j}^t - \tau_{s_2, i,j}^t)^2.$$

(16)

Figure 3 shows the workflow of hypothesis testing collision detection. Figure 5 shows cumulative distance distributions of collision and non-collision. The leftmost red part is the case of collision which has the smallest mean value, and is clearly separated from the other cases of non-collision marked black. Then, the probability of two types errors can be expressed as:

$$\begin{align*}
\alpha &= Pr\{Col\{d(\tau_{s_1, 1}, \tau_{s_2, 1})\} = 0|p_{s_1}^t \oplus k_{s_1} = p_{s_2}^t \oplus k_{s_2}\} \\
\beta &= Pr\{Col\{d(\tau_{s_1, 1}, \tau_{s_2, 1})\} = 1|p_{s_1}^t \oplus k_{s_1} \neq p_{s_2}^t \oplus k_{s_2}\}.
\end{align*}$$

(17)

In order to verify the success rate of this detection technique, we conduct 50,000 experiments, a total of 6 million collision tests, and collect relevant information which is shown in Table 1.

From Table 1, the probability of Type I error, i.e. $Pr_f$ in the $4^{th}$ column, is close to the significance level $\alpha$ in the first column, especially as $\alpha$ gets smaller and smaller. Therefore,
The type II error will increase with the increase of $\alpha$. Without average, the probability of success rate decreases. As can be seen from Figure 6, the success rate of collision detection increases with the decrease of $\alpha$. It is necessary to balance the impact of $\alpha$ and $\beta$ on collision detection when a practice attack is made. In our experiments, we can achieve a success rate of 98.77%.

C. TWO FACTORS THAT AFFECT THE SUCCESS RATE OF COLLISION DETECTION

Two factors that affect the success rate of collision detection are also considered. As can be seen from Figure 6, the success rate increases with the increase of $1 - \alpha$ when the average method is used. Without average, the probability of the type II error will increase with the increase of $1 - \alpha$. The reason is if more and more non-collision cases are being misinterpreted as collisions, which in turn affects the success rate of detection.

Figure 7 illustrates that the accuracy of the estimated constant power has large influence on the detection success rate. We conduct three sets of experiments, using 10 different plaintexts, 30 different plaintexts and 50 different plaintexts to estimate the constant power, respectively. The results show that the success rate in the case of 50 different plaintexts is higher than the other two cases, which indicates that accurately estimating the constant power is helpful to improve the correctness of the attack.

Collision detection is almost impossible without considering the constant power. Figure 8 shows cumulative distance distributions of collision and non-collision without subtracting constant power for each point. The red part is the cumulative distance distribution in the case of collision, which can not be distinguished from other non-collision cases marked black due to the constant power consumption may not actually be constant.

Figure 9 compares the evolution of success rate for hypothesis testing collision detection and the collision detection techniques mentioned in [19]. It is obvious that the performance of HTCD is better than the other two techniques. As a matter of fact that it is hard to decide the threshold, hypothesis testing collision detection is more efficient.

<table>
<thead>
<tr>
<th>Theory</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$z_{1-\alpha}$</td>
</tr>
<tr>
<td>0.2000</td>
<td>0.84</td>
</tr>
<tr>
<td>0.1600</td>
<td>0.99</td>
</tr>
<tr>
<td>0.1200</td>
<td>1.18</td>
</tr>
<tr>
<td>0.0700</td>
<td>1.48</td>
</tr>
<tr>
<td>0.0200</td>
<td>1.98</td>
</tr>
<tr>
<td>0.0100</td>
<td>2.25</td>
</tr>
<tr>
<td>0.0060</td>
<td>2.51</td>
</tr>
</tbody>
</table>

1 It was assumed that constant power is well estimated.
2 The value of $\lambda$ (mentioned in Subsection III-A) has been chosen that minimize the number of traces needed.
D. ATTACK ON MASKED S-BOXES OF AES

When applied to attack the masked S-boxes whose masks are re-used based on the collision attack in [14], hypothesis testing based collision detection attack (HTCD) also has great performance. We randomly select several plaintexts, and encrypt each of them for \( n \) times, and acquire the power traces corresponding to \( \text{SubBytes}(p_{s_1} \oplus k_{s_1}) \) and \( \text{SubBytes}(p_{s_2} \oplus k_{s_2}) \) respectively. By using HTCD, \( k_{s_1} \) and \( k_{s_2} \) can be recovered.

Figure 10 describes the re-used masking scheme, this masked S table is usually computed before the AES execution and stored in volatile memory. Figure 11 shows the evolution of the success rate for hypothesis testing based collision detection attack and the collision attack in [14] under different noise levels, both the two schemes do not need to traverse plaintexts. In the case where \( \sigma \) is equal to 0.0015 or 0.006, it is obvious that the performance of HTCD is better than collision correlation attack.

At last, Table 2 summarises several typical collision attacks and their attack capabilities. The typical collision attacks we select include linear collision attack (LCA) [15], collision correlation attack (CCA) [14], non-linear collision attack (NLCA) [22], near collision attack (NCA) [23], collision attack on linear layers (LLCA) [24], and scalable collision attack on linear layers (LLSCA) [24]. Four practical hotspot issues for side channel collision attacks are used to compare, including whether traversal of plaintexts is required, whether practically determine the threshold is required, whether the leak samples is easily identifiable, and whether re-used masking scheme can be broken.

Note that, these practical issues will affect the implementation of a collision detection scheme. First is to identify leakage samples accurately. For NLCA, it detects collisions processed under different operations, and it is a challenge to find the two leaking samples which refer to the leakage of such operations. Second is to minimize the amount of plaintexts required. For NCLA, NCA, LLCA and LLSCA, it is necessary to traverse plaintexts to detect collision more reliable. Third is to determine the threshold in a more reasonable way, rather than relying on practically determine. Last but not least, in order to protect cryptographic algorithms against side-channel attacks, masking techniques are used to blinding the information leakage. Among them, re-used masking scheme is a time- and cost-effective approach, and we need to make sure we can break it.
TABLE 2. Typical collision attack methods and their attack capabilities.

<table>
<thead>
<tr>
<th></th>
<th>LCA</th>
<th>CCA</th>
<th>NLCA</th>
<th>NCA</th>
<th>LLCA</th>
<th>LLSCA</th>
<th>HTCDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to identify leakage samples</td>
<td>✔</td>
<td>✔</td>
<td>✘</td>
<td>✔</td>
<td>✘</td>
<td>✘</td>
<td>✔</td>
</tr>
<tr>
<td>No need to traverse plaintexts</td>
<td>✘</td>
<td>✔</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✔</td>
</tr>
<tr>
<td>No need to practically determine the threshold</td>
<td>✘</td>
<td>✘</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Effective for re-used masking scheme</td>
<td>✘</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, a collision detection technique based on hypothesis testing is proposed, which is simple, feasible and has a high success rate. The rejection region is needed to be determined only once, and all collision detections can be performed. When applied to practice attack, we conduct 50,000 experiments with 98.77% detection success rate in the case of unprotected S-boxes, each experiment has 120 collision detections. When applied to attack the masked S-boxes whose masks are re-used, we also has great performance.

The power collision is analyzed from the perspective of power trace composition. When the same operations are implemented in the same way, processing the same data will lead to approximately equal power values; on the contrary, processing different data will lead to different power values. Then we pay attention to the remaining two parts of power, the constant power and the electronic noise. On one hand, when the variance of electronic noise is large, the probability of type II error will increase obviously. On the other hand, the attack efficiency can be improved by accurately estimating the constant power. In the end, we compare the works related to side channel collision attack in recent years from the perspective of four hot issues. The technique described in this work might turn out applicable to other symmetric constructions such as stream ciphers or lightweight block ciphers. Our technique also can overcome certain random masking schemes for block ciphers, such as re-used masking scheme.

ACKNOWLEDGMENT

The corresponding author Meiling Zhang, on behalf of all the authors, would like to thank the reviewers and the editors who contribute to the great improvement of the original version of this paper with their valuable comments and suggestions.

REFERENCES


