SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2018, Vol. 61 032101:1–032101:11
doi: 10.1007/s11432-016-0616-4

# Right or wrong collision rate analysis without profiling: full-automatic collision fault attack

An WANG[1,2], Yu ZHANG[1], Weina TIAN[3], Qian WANG[4],
Guoshuang ZHANG[5] & Liehuang ZHU[1*]

[1] *School of Computer Science, Beijing Institute of Technology, Beijing* 100081*, China;*
[2] *State Key Laboratory of Cryptology, P.O. Box 5159, Beijing,* 100878*, China;*
[3] *College of Bioengineering, Beijing Polytechnic, Beijing* 100176*, China;*
[4] *Department of Electrical and Computer Engineering, University of Maryland, College Park* MD 20740*, USA;*
[5] *Science and Technology on Information Assurance Laboratory, Beijing* 100072*, China*

Received 3 October 2016/Accepted 24 November 2016/Published online 22 May 2017

**Abstract** In CHES 2010, Fault Sensitivity Analysis (FSA) on Advanced Encryption Standard (AES) hardware circuit based on S-box setup-time acquired by injecting clock glitches is proposed. Soon after, some improvements of FSA were presented such as colliding timing characteristics from Moradi et al. However, the acquisition of timing characteristics requires complex procedure due to the very gradual decrease of clock glitch cycle and the heavy requirements of setup-time samples. In HOST 2015, Wang et al. presented template-based right or wrong collision rate attack to improve the efficiency of FSA, but its profiling and plaintexts-choice procedures required too many encryptions. In this paper, we fix only one specific clock glitch cycle, and take the right or wrong collision rate as a collision distinguisher. So, the whole process is a non-profiling collision attack which can be executed automatically without massive pre-computations and interactions between PC and signal generator. According to the experiments, 256 encryptions are enough for exactly deciding whether two plaintext bytes can induce an S-box collision. Compared with the existing power analysis and FSA-based attacks on AES hardware, it costs negligible time (about 6.65 s) and storage space (only one byte), and no offline computations for finding the collision between two masked S-boxes. Furthermore, our study shows that the signal-to-noise ratio in FSA-based attacks is much higher than power-based attacks.

**Keywords** fault sensitivity analysis, right or wrong collision rate analysis, masking, collision attack

**Citation** Wang A, Zhang Y, Tian W N, et al. Right or wrong collision rate analysis without profiling: full-automatic collision fault attack. Sci China Inf Sci, 2018, 61(3): 032101, doi: 10.1007/s11432-016-0616-4

## 1 Introduction

Since the work of Kocher in 1996 [1], many cryptanalysts have focused on side-channel attacks on cryptographic circuits. The secret key can be leaked by side-channel information such as timing, power, electromagnetic wave, sound, and so on. Based on various side-channel information, some side-channel techniques such as collision [2–5], correlation coefficient [6–9], template [10], differential fault [11], and so on were considered.

Side-channel collision attack decreases the information of secret key by detecting collisions between intermediate values during algorithm implementations. Schramm et al. [5] first proposed side-channel

[*] Corresponding author (email: liehuangz@bit.edu.cn)

collision attacks against block ciphers in 2003, which detected internal collision in a same S-box of DES for two encryptions and executed differential analysis. Subsequetly, a collision attack on MixColumns operation of AES was presented [4]. Then Bogdanov et al. [2] gave the linear collision attack by directly building an equation about the key bytes according to byte collision. One year later, Bogdanov et al. [3] showed the voting-based collision detection by combining the statistical techniques. In the past three years, the non-linear collision attack and near collision attack are proposed [12, 13].

In CHES 2010, Li et al. [14] showed a fault attack named Fault Sensitivity Analysis (FSA) against cryptographic circuits which utilized the dependence between the input of an AES S-box and critical timing delay of combinational circuit. For an S-box in 10th round of AES, the attacker can get the setup-time corresponding to the S-box inputs by illegal clock injection and setup-time violation, and further recover the key byte by the method similar to the correlation power analysis. In CHES 2011, Moradi et al. combined correlation-enhanced collision attack [15] with fault sensitivity analysis, and gave new collision attack on two masked S-boxes of AES hardware implementation by colliding timing characteristics [16]. In 2013, the fault rate induced by the clock glitch is employed for collision detection on masked AES S-boxes [17]. Soon after, some other models are discussed [18, 19].

However in practical FSA attacks, the collection efficiency and demand of setup-time samples are not satisfying, in especial for masked implementation. To find a collision between two masked S-boxes, millions of encryptions are required because:

- In correlation coefficient based attacks, many setup-time samples should usually be recorded [14]. For example in the method of colliding timing characteristics [16], inputs of the two target S-boxes should traverse 0–255, and 512 setup-time samples must be acquired in total.

- In order to get the accurate setup-time samples corresponding to each input of an S-box, encryptions should be repeated for many times under different (gradually decreasing) glitch cycles [14]. Usually, more than 100 glitch cycles are required for each accurate sample.

- For each glitch cycle, it should be tested whether the correct rate of output byte is higher than a certain threshold [16]. So, at least, fifty encryptions are usually required.

Furthermore, in order to automatically execute the tedious process above, a software should be specially designed to control the signal generator. So, a sophisticated function would be expected as a better collision distinguisher to find a collision fast and conveniently.

**Our contribution.** In this paper, a side-channel collision attack based on right or wrong collision rate analysis on masked AES hardware implementations is presented.

- We take the occurring probability of right or wrong collision as collision distinguisher between two S-boxes. So, no pre-computations and offline post-computations are required in template setup and collision detection, respectively.

- Under FPGA environment, we choose a specific glitch cycle, and inject it into two S-boxes of 10th round. Our experiment shows that an S-box collision/non-collision will definitely emerge with only 256 encryptions. So, in online stage, it requires much less samples than the existing side-channel attacks because of the significant advantage in signal-to-noise ratio.

- For an attack, our proposal requires only one glitch cycle. So, the whole collision detection can be executed automatically. Furthermore, an optimal glitch can be chosen easily by a slight modification of the attack system.

**Organization.** The remainder of this paper is organized as follows. In Section 2, we briefly recall the previous work on masking and fault sensitivity analysis. Section 3 introduces the distinguisher for collision detection, and describes the concrete attack process. Subsequently, we show the experiments of practical collision attack on masked AES in Section 4. Finally, Section 5 concludes this paper.

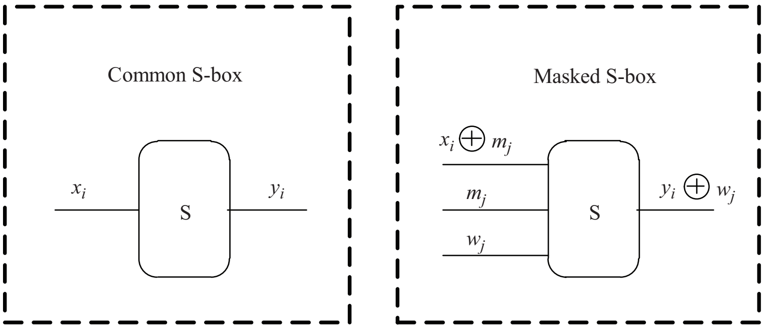

**Figure 1** The common and masked S-box.

## 2 Preliminary

### 2.1 S-box and masking

The Advanced Encryption Standard (AES) algorithm has been widely used in practice, which includes 10 rounds, and each one has 16 S-boxes. To increase the throughput, 16 S-boxes are usually implemented in parallel architecture. Sometimes, a combinational circuit consisting of only 4 S-boxes may run for 4 times in order to finish the 16 S-boxes [20]. Due to the limited area, S-box is usually implemented by the method of tower field [21–23]. Morioka et al. [24] proposed a resource-limited AES design with an optimized S-box circuit architecture, whose critical timing delay depended on the Hamming weight of inputs.

Masking technique is a conventional countermeasure against side-channel attack, which can randomize the intermediate values during the encryption. For the past few years, some masked S-box implementations of AES have been designed [25–27], in which the input and output of S-box are masked. For the validity of computation, both masked value and two masks are input into S-box, and one masked value is output, which is described in Figure 1. For reasons of efficiency, the masks may be reused between different S-boxes or rounds.

**Collision detection of masked S-boxes.** In this paper, we focus on two S-boxes which are masked by the same input mask value $m$ and output mask value $w$. So, the collision detection between two masked S-boxes $S_1, S_2$ still means to find $x_1 = x_2$ (or $y_1 = y_2$).

### 2.2 Fault sensitivity analysis

For combinational circuit, its outputs will be steady after a short time if some input bits transit. The delay time is named setup-time. In CHES 2010, Li et al. showed that an illegal clock could bring a setup-time violation. This is because flip-flops were triggered before the previous output of combinational circuit was fixed to a correct value [14]. This illegal clock is generated by a clock glitch. When this scene happens in last round of AES, the attacker can determine whether an S-box produces a fault according to the correctness of ciphertexts. So, if the attacker decreases clock glitch cycle gradually until the S-box exactly makes an error, this cycle will approach to the S-box setup-time for the current input value.

Because of the data dependency of the schemes proposed in [24], a method based on correlation coefficient can be used for key recovery. On the one hand, we consider the Hamming weights of S-box inputs as a vector. On the other hand, the setup-time samples for these inputs compose another vector. Thus, the key guess corresponding to the maximum correlation coefficient between the two vector is the right one. This process is called fault sensitivity analysis.

### 2.3 Fault sensitivity analysis on masking

The fault sensitivity analysis can also break masking scheme. In 2011, Moradi et al. [15] got the 256 setup-time samples $T_1^o$: $(\Delta t^{o=0}, \Delta t^{o=1}, \ldots, \Delta t^{o=255})$ of masked S-box by a great deal of averages, and employed the correlation-enhanced method for measuring the correlation between two groups of samples

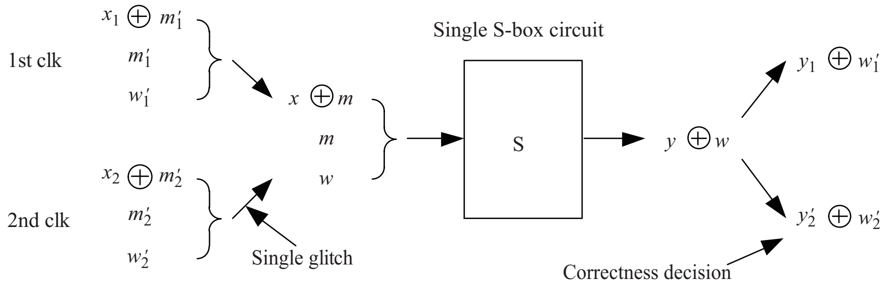

**Figure 2** Fault rate analysis in 2013 (only for serial S-boxes).

corresponding to two S-boxes [16]. As a result, the key difference $\Delta$ corresponding to the maximum correlation coefficient is correct. This attack is called colliding time characteristics which is described in Algorithm 1.

**Algorithm 1** Correlation timing attack on round 10 of AES

```
Input: $T_1^o$: $(\Delta t^{o=0}, \Delta t^{o=1}, \ldots, \Delta t^{o=255})$; $o = \mathrm{Sbox}(i) \oplus k_1$
Input: $T_2^o$: $(\Delta t^{o=0}, \Delta t^{o=1}, \ldots, \Delta t^{o=255})$; $o = \mathrm{Sbox}(i) \oplus k_2$
for $0 \leqslant \Delta \leqslant 255$ do
    $\mathrm{Cor}(\Delta) = \mathrm{Correlation}(T_1^o, T_2^{o \oplus \Delta})$
end for
return $\arg\max_{\Delta} \mathrm{Cor}(\Delta)$
```

**Shortcoming of FSA-based attacks.** For a fixed $x$, in order to determine the setup-time of a changing $x \oplus m$, an attacker shortens gradually the glitch cycles until the fault rate of outputs is more than a threshold. So, for each setup-time sample, the attacker must repeat encryptions many times. In more details, in the experiments of [16], getting one setup-time sample of unmasked S-box needs about 10000 captures, and 1000000 captures were required in the case of masked S-boxes. Furthermore during the colliding time characteristics, computing the correlation coefficient needs 512 samples of setup-time in total.

### 2.4 Fault rate analysis on masking

Fault Rate Analysis (FRA) [17] proposed in 2013 focuses on two S-boxes implemented by single S-box circuit in serial. The attacker injects a clock glitch with some sophisticated cycles into the second S-box, which is described in Figure 2. If no fault happens, i.e., there are no transitions in the second S-box, a collision can be detected. This collision distinguisher is not sensitive to the glitch cycle, so it shows efficiency to a certain degree. However, the shortcoming of fault rate analysis is that, it is only appropriate for the serial implementation of S-box, and in vain for the parallel implementation.

### 2.5 Template attack based on right or wrong collision rate

In HOST 2015, Wang et al. [19] gave the definition of right or wrong collision (RW collision for short), and employed it for a template-based fault attack.

**Definition 1** (RW collision). Assume that a same clock glitch is injected into the two S-boxes $S_1, S_2$. If either of the following cases takes place, it is called RW collision (else RW non-collision):

- Both of the $y_1 \oplus w$ and $y_2 \oplus w$ are right computed;
- Both of the $y_1 \oplus w$ and $y_2 \oplus w$ are wrong computed.

The number of RW collisions can be recorded by a counter. So, the probability of RW collision (or the number of RW collisions) $\rho$ can be taken as a collision distinguisher, which is called RW collision rate. Figure 3 shows an example of this process.

**Shortcoming of template attack based on RW collision rate.** This attack from Wang et al. is a template-based one which executes 256 × 256 encryptions for template setup and costs 256 × 4 bytes

| Exp. index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_1$ output | R | W | W | W | R | W | R | W | W | W | ... |
| $S_2$ output | R | W | R | W | R | W | W | R | R | W | ... |
| RW collision? | Y | Y |  | Y | Y | Y |  |  |  | Y | ... |

Template setup/match

**Figure 3** The example of RW collision rate attack in HOST 2015.

at least for their storage. Before the template match, more than 300000 pre-encryptions are required for choosing the appropriate plaintexts. Furthermore, an assumption that the adversary has a same device whose secret key is under control is usually needed for template attacks.

## 3 Collision attack based on right or wrong collision rate

### 3.1 Basic idea

The inefficiency of FSA-based attack [14, 16] is due to the gradual decrease of glitch cycle. So, we try to fix the glitch cycle during the whole attack. Specifically, if the correctness of S-box computation is used as leakage function, and a well-chosen counter function is employed as collision distinguisher, the complexity in online stage will be decreased significantly since the glitch cycle does not require to be changed. And the cost in offline stage will be very little since the only computation module consists of two comparators and a counter.

From another point of view, template-based RW collision rate attack [19] costs many pre-encryptions on template setup and plaintext choice. So, we try to construct a collision attack to avoid the profiling process.

For the masked AES, we consider the two S-boxes $S_1, S_2$ in Round 10 of AES. In Figure 4, $x_1, x_2, y_1, y_2$ denote the unmasked inputs and outputs of $S_1, S_2$, and $m$, $w$ denote the input and output mask values respectively. Under the same glitch cycle, if $x_1 = x_2$, then $S_1$ and $S_2$ may reflect the same correctness. And for the case of non-collision, there may be some differences due to the influence of mask. So, in noiseless environment, the former probability should be 1, and the latter probability should be strictly less than 1.

**Remark.** The correctness of $y_1 \oplus w$ and $y_2 \oplus w$ can be decided by the output ciphertexts. This is because, $y_i$ is wrong if and only if $y_i \oplus w$ is wrong when the mask $w$ is correctly computed. In fact, the path from $w$ to $y_i \oplus w$ is much shorter than the critical path of S-box (discussed in Subsection 4.2). So, $w$ joins the whole S-box computation correctly, and has no influence on the correctness determination of $y_i \oplus w$.

### 3.2 Attack scenario

In this section, we explain the correctness of our attack first. It is based on the same assumptions as [16] that 256 setup-time values corresponding to 256 S-box inputs $x_i \oplus m$ are not exactly equal (not limited to Hamming weight model). In Section 4, we verify our attack in practice.

**Correctness of collision distinguisher.** The collision distinguisher is correct if and only if the RW collision rate in collision is strictly greater than non-collision. This comparison can be explained as follows.

When a glitch cycle is repeatedly injected into a combinational circuit which has the same inputs, the outputs may be different because of the noise of clock glitch. But in the same environment, the

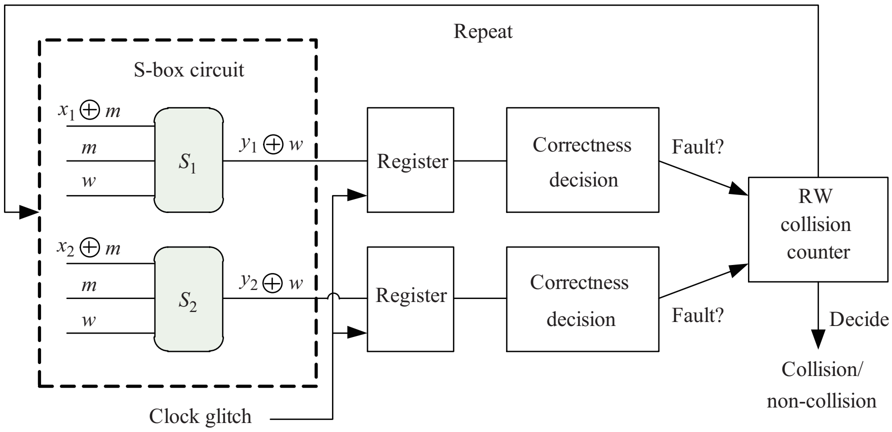

**Figure 4** The main idea of right or wrong collision rate analysis (for both serial and parallel S-boxes).

expectation of correct rate from abundant experiments trends to a constant. For simplicity, we consider the case without noise, and let $P_j$ denote the correct rate when $x_i \oplus m = j$.

If $x_1 = x_2$, two S-boxes produce an RW collision with the following probability.

$$\rho_1 = \frac{1}{256}\sum_{i=0}^{255}[{P_i}^2 + (1 - P_i)^2].$$

If $x_1 \oplus x_2 = a \neq 0$, this probability is

$$\rho_2 = \frac{1}{256}\sum_{i=0}^{255}[P_i \cdot P_{i\oplus a} + (1 - P_i)(1 - P_{i\oplus a})].$$

For any $a$, there exists a conjugacy: if $x_1 \oplus m = i$, then $x_2 \oplus m = i \oplus a$. Conversely, if $x_1 \oplus m = i \oplus a$, then $x_2 \oplus m = i$ holds. Therefore, given $x_1, x_2$ and $x_1 \neq x_2$, the set $\{x_1 \oplus m, x_2 \oplus m\}$ $(m = 0, 1, 2, \ldots, 255)$ has 128 possibilities, which traverse 0–255 exactly. So, the 256 items of $\rho_1$ and $\rho_2$ can be divided into 128 pairs respectively. For each pair, we have

$$P_i^2 + (1 - P_i)^2 + P_{i\oplus a}^2 + (1 - P_{i\oplus a})^2 \geqslant 2[P_i \cdot P_{i\oplus a} + (1 - P_i) \cdot (1 - P_{i\oplus a})].$$

As a result, the sum of 128 inequations implies $\rho_1 \geqslant \rho_2$, in which the equality never holds in practice obviously.

**Attack process.** After determining a proper threshold between $\rho_1$ and $\rho_2$ by pre-computation, the distinguisher can decide collision or not, which is described in Algorithm 2. Subsequently, the relationship between the two key bytes corresponding to $S_1$ and $S_2$ can be determined. Similarly, repeating online and key recovery stage of Algorithm 2, the attacker can shrink the 128-bit key space to $2^8$ according to the linear collision attack [2].

**Remark.** In order to correspond to the correctness discussion, this algorithm shows a profiling attack in which a predetermined threshold $\rho_0$ is chosen. In fact, our proposal can also be used for the non-profiling attack. During the change of ciphertexts, especially for the chosen-ciphertext attack, there may exist one collision in each 256 ciphertexts on average, whose distinguisher would be higher than those of the other 255 non-collisions. So, non-profiling method can also be competent for RW collision rate analysis attack.

## 4 Experiments and efficiency

### 4.1 Simulation experiments

We have made some simulation experiments in Matlab based on the Hamming weight model. We assume that nine Hamming weights (0, 1, ..., 8) of $x_i \oplus m$ correspond to the setup-time of S-box 10 ns, 11 ns, ...,

**Algorithm 2** RW collision rate analysis on two masked S-boxes

```
Pre-computation stage:
1: Determine the maximum and minimum setup-time $t_1$ and $t_2$ of S-box circuit.
2: Choose a glitch cycle $t$ such that $t_2 < t < t_1$.
3: Determine threshold $\rho_0$ of distinguisher in the case that glitch cycle is $t$.
Online stage:
1: Under the normal clock, encrypt a random plaintext $P$ correctly.
   (Two ciphertext and subkey bytes corresponding to $S_1, S_2$ of round 10 are denoted by $c_1, c_2, k_1, k_2$).
2: Inject a glitch whose cycle is $t$ into round 10 and encrypt $P$ for $n$ times.
3: Compare the outputs of step 2, $n$ pairs $(c_1', c_2')$, with the correct value $c_1$ and $c_2$.
4: Compute the probability (or the number) $\rho$ of RW collision.
5: If $\rho \geqslant \rho_0$, then $x_1 = x_2$, else $x_1 \neq x_2$.
6: If collision is decided, carry out key recovery stage, else repeat online stage.
Key recovery stage:
1: For a decided collision, $k_1 \oplus k_2 = c_1 \oplus c_2$ holds.
```

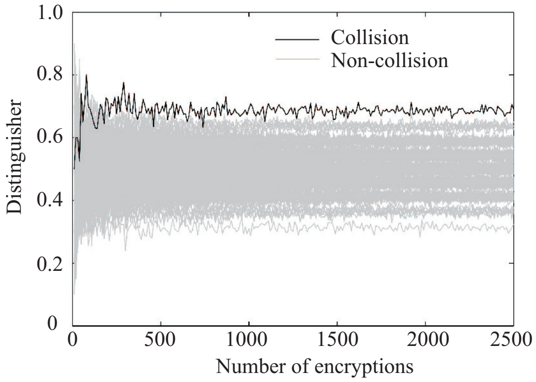

**Figure 5** The relation between the number of encryptions and RW collision rate in simulation.

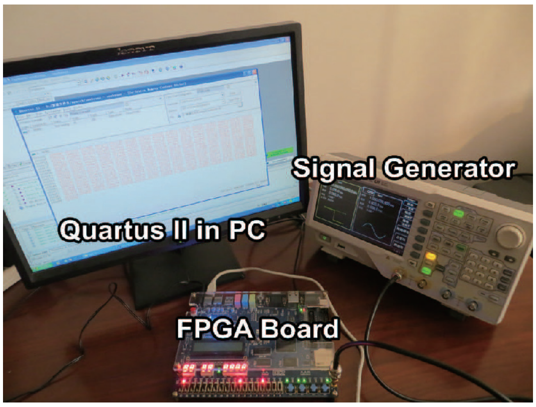

**Figure 6** (Color online) The experiment platform of proposal attack.

18 ns respectively. For simplicity, we assume that the values of $m, w$ have no influence on the distinguisher. The glitch cycle is chosen as a random variable which follows a normal distribution whose expectation is 14 ns and standard deviation is 1.3 ns. So in our simulation environment, the signal-to-noise ratio can be expressed as 1:1.3. We choose $x_1 = 1$ and let $x_2$ traverse 0–255. For each value of $x_2$, encryption is carried out for 2500 times (during each encryption, $m$ is a random value of 0–255) under the chosen glitch, and 2500 RW collision (or non-collision) samples can be recorded. So, a curve which shows the relation between the number of encryptions and probability can be acquired, which is described in Figure 5. Obviously, in our simulation environment, 2000 encryptions are sufficient to detect a collision. The black and 255 gray curves represent the case of collision and non-collision respectively.

## 4.2 Experiments in FPGA

We made a practical experiment with the help of computer, RIGOL DG4102 signal generator, and DE2-115 development board with Cyclone IV EP4CE115 FPGA chip, which is showed in Figure 6. The total expense of these devices is no more than \$1000, which is much cheaper than power and laser attacks.

Figure 7 describes the circuit structure of proposal attack. First, the masked S-box of AES based on the combinational field [25] is implemented. Figure 8 shows its brief architecture. It is clear that the path from $w$ to output $y \oplus w$ is very short, and the critical path is longer obviously. So, the computations which $w$ joins will make no mistakes under the clock glitch cycle close to the critical timing delay.

Second, the glitch module we expected was implemented by an on-chip glitchy-clock generator [28]. The output correctness of the masked S-box was determined by a normal lookup-table S-box and a comparator

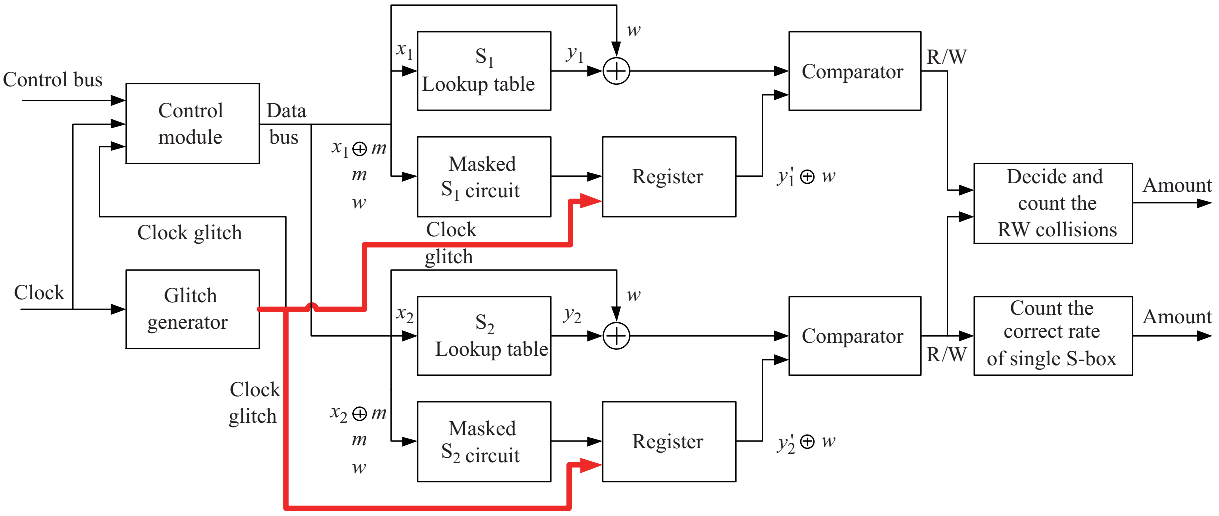

**Figure 7** (Color online) The circuit structure of our experiments in which the thick lines mean the clock with glitch.

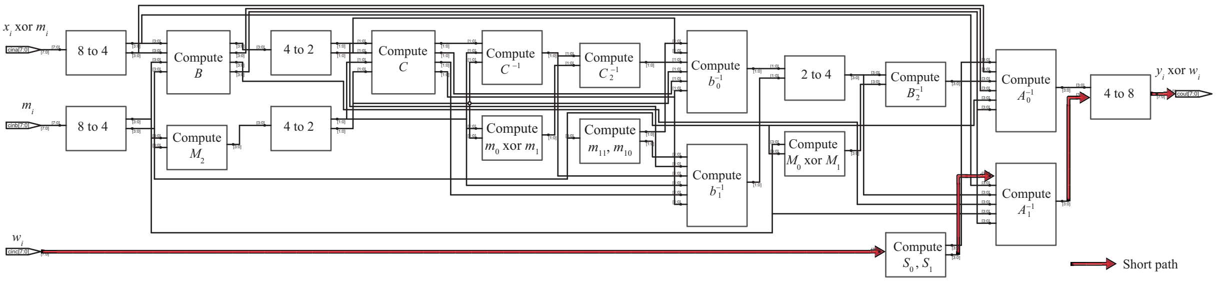

**Figure 8** (Color online) The sketch map of hardware architecture of masked S-box.

Finally, a counter recorded the number of right or wrong collisions, and the data were transmitted to PC by "in system memory content editor" in Quartus II.

We can distinguish collision from non-collision distinctly if $x_1$ traverses from 0 to 255. For each pair $(x_1, x_2), x_1, x_2 \in \{0, 1, \ldots, 255\}$, we choose $m$ from 0–255 randomly, and make one experiment (each experiment consists of 256 encryptions). Then we can get a total RW collision rate corresponding to the pair $(x_1, x_2)$, which is labelled to Figure 9 as a point. In Figure 9, $x$-axis means $x_1$ value. For each $x_1$, the above star corresponds to the case of collision, i.e., $x_2 = x_1$, and the below 255 grey points correspond to non-collisions, i.e., the other 255 $x_2$. Obviously, for each pair $(x_1, x_2)$, attacker can distinguish collision from non-collision distinctly with very high probability. Compared with other attacks, the signal-to-noise ratio in practical RW collision analysis is much higher than that in most power analysis (about 1:3–1:5) and in our simulation above (1:1.3).

**Remark.** In practical attacks, the adversary can only study the setup-time of the whole Round 10, instead of the setup-time of only S-box in our experiment. However, all of the other calculations in Round 10 are XOR operations which are data independent [14]. So, the critical path of S-box can be reflected by the critical path characteristics of the whole Round 10 accurately.

### 4.3 Choice of glitch cycle

The optimal glitch cycle can be chosen by the following process. First, under a certain glitch cycle, choose a random plaintext, encrypt it for many times, and counter the correct rate of only one S-box (instead of two S-boxes). Then decrease or increase the glitch cycle gradually such that the correct rate of single S-box is 50%. Now, this glitch cycle can be taken for the subsequent attack. This procedure can be executed very easily because one can directly twist the frequency knob of the signal generator gradually. Usually, no more than 1000 encryptions is enough for decide an appropriate clock glitch.

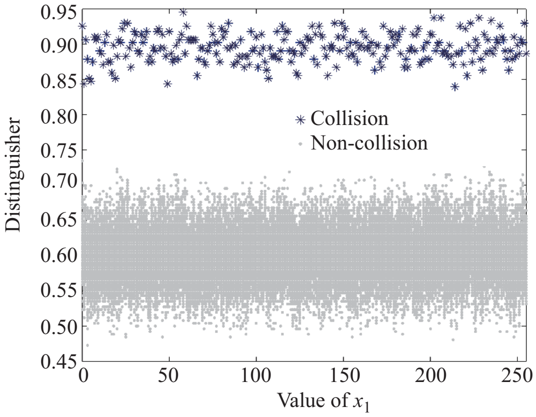

**Figure 9** (Color online) Distinct distinguisher values between collision (stars) and non-collision (points) in practice.

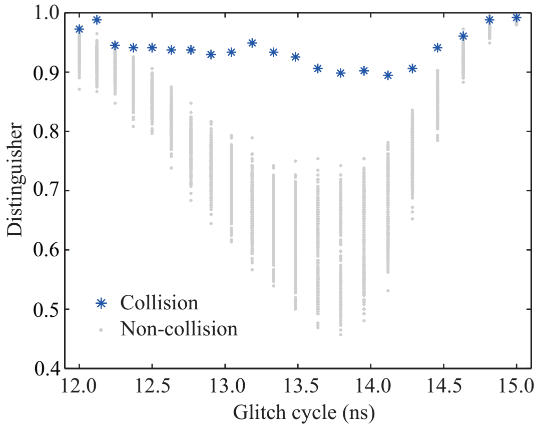

**Figure 10** (Color online) The relation between the distinguisher and glitch cycles in practice.

The glitch cycle corresponding to 50% is the best for collision detection because in most cases, the value

$$(P_i^2 + (1 - P_i)^2) - (P_i P_{i\oplus a} + (1 - P_i)(1 - P_{i\oplus a}))$$

achieves the largest one for more $i \in \{0, 1, \ldots, 255\}$. So, the distinguisher corresponding to this clock glitch cycle brings optimal success rate. This computation is very easy to be implemented in our experimental circuit. Only a counter and a register are required extra, which is described in Figure 7.

According to the process above, we made an experiment for $x_1 = 33\text{H}$. The corresponding setup-time of a single S-box is about 13.80 ns when the correct rate of this S-box is 50%. Then we mount a collision detection for $x_2 = 0, 1, 2, \ldots, 255$ in different glitch cycles. The glitch cycles are chosen by the following method. First, the original frequencies is set to 20, 20.25, 20.5, $\ldots$, 25 MHz which is gotten from the signal generator. Then they are converted to 66.67, 67.5, 68.33, $\ldots$, 83.33 MHz by the PLL module in FPGA. In other words, the corresponding glitch cycles are 15, 14.81, 14.63, $\ldots$, 12 ns. The distinguisher values in different cycles are shown in Figure 10. It is clear that the distinguisher seems to show high signal-to-noise ratio in 13.80 ns or so.

### 4.4 Efficiency comparison

Table 1 shows the comparison among several collision attacks on S-box masking. Our attack has remarkable advantages in encryption time and space, computation complexity, etc. Specifically in order to find a collision with the success rate of 95%, for on-line encryption, the fault-based CTC (colliding timing characteristics [16]) and FRA (fault rate analysis [17]) should respectively execute $10^6$ and 80000 encryptions in FPGA which cost about 100 and 8.1 s, and record 512 and 40 samples which occupy 2048 and 80 bytes. However, our attack needs to execute 256 collision detections for $x_2 = 0, 1, 2, \ldots, 255$. So, all the 256 collision detections cost $0.0256 \times 256$ s for $256 \times 256$ encryptions and only one byte for the counter storage, which is negligible for CTC and less than FRA.

Power-based attacks usually spend 100 ms or longer for recording a power trace, which is much longer than an FPGA encryption. So, CEPA (correlation-enhanced power analysis [15], which needs to detect collision only once) and CCPA (collision-correlation power analysis [8], which consists of 256 collision detections) executed in FPGA cost more than 500 s for 5000 encryptions and $200 \times 256$ s for $2000 \times 256$ encryptions respectively. What's more, their space costs $2 \times 10^6$ bytes and $1.6 \times 10^7$ bytes are much higher than that of our proposal.

The offline complexity row gives computation complexity in PC for collision determination. $\mathcal{C}_{\rho_n}$ means the computation complexity of correlation coefficient between two vectors, each of which includes $n$ samples. In our attack, no computations are needed. Even if the attacker want to compute the RW collision probability, only one division may be computed.

**Table 1** The efficiency comparisons of six methods for finding a collision

| Item | CEPA [15] | CCPA [8] | CTC [16] | FRA [17] | T-RWCRA [19] | This paper |
|---|---|---|---|---|---|---|
| Attack channel | Power | Power | Fault | Fault | Fault | Fault |
| Num of plaintexts | 256 | 256 | 256 | 256 | $256 \times 10$ | 256 |
| Num of encryptions | $> 5 \times 10^3$ | $> 2 \times 10^3 \times 256$ | $10^6$ | $8 \times 10^4$ | $256 \times 10$ | 65536 |
| Num of pre-encryptions | 0 | 0 | 0 | $< 1000$ | $> 300000$ | $< 1000$ |
| Enc time (s) | $> 500$ | $> 200 \times 256$ | 100 | $< 8.1$ | $> 30$ | $< 6.65$ |
| Recorded samples | 512 | $> 4 \times 10^3$ | 512 | 40 | 256 (templates) | 0 |
| Space occupation (bytes) | $2 \times 10^6$ | $> 1.6 \times 10^7$ | 2048 | 80 | 1024 + 4096 | 1 |
| Offline complexity | $256\ C_{\rho_{256}}$ | $256\ C_{\rho_{2000}}$ | $256\ C_{\rho_{256}}$ | 0 | 0 | 0 |
| Num of glitches | – | – | 50 | 40 | 1 | 1 |
| Serial/parallel availability | Both | Serial | Both | Serial | Both | Both |
| Extra assumption | Need | No need | Need | No need | No need | No need |

During CTC, experiments should be made under 50 kinds of glitch cycles in order to get the fault sensitivity. In FRA, 40 kinds of glitch cycles are needed for the average of 40 fault rates. But in our method, only one glitch is enough for the setup-time violation tests of the two S-boxes. For the availability, the new proposal can break both serial and parallel implementations of AES hardware. Moreover, CEPA and CTC usually need an assumption of "imperfect mask" which means that averaging directly may leak the unmasked information [15]. However, our attack does not need this kind of assumptions.

## 5 Conclusion

In this paper, we propose a non-profiled right or wrong collision rate analysis on protected AES. Based on a clock glitch injection to two masked S-boxes, the correctness of each S-box is employed for building a collision distinguisher named RW collision rate. In our experiments, only 256 setup-time samples are enough for deciding a collision or non-collision between the specific inputs of two masked S-boxes. So, the signal-to-noise ratio in FSA-based attacks is obviously much higher than that in power-based attacks. Furthermore, our attacks are applicable to other block ciphers due to the similar S-boxes implementations.

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

**Conflict of interest** The authors declare that they have no conflict of interest.

### References

1 Kocher P. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Advances in Cryptology—CRYPTO'96. Berlin: Springer, 1996. 104–113

2 Bogdanov A. Improved side-channel collision attacks on AES. In: Selected Areas in Cryptography. Berlin: Springer, 2007. 84–95

3 Bogdanov A. Multiple-differential side-channel collision attacks on AES. In: Cryptographic Hardware and Embedded Systems—CHES 2008. Berlin: Springer, 2008. 30–44

4 Schramm K, Leander G, Felke P, et al. A collision-attack on AES combining side channel- and differential-attack. In: Cryptographic Hardware and Embedded Systems—CHES 2004. Berlin: Springer, 2004. 163–175

5 Schramm K, Wollinger T J, Paar C. A new class of collision attacks and its application to DES. In: Fast Software Encryption. Berlin: Springer, 2003. 206–222

6 Brier E, Clavier C, Olivier F. Correlation power analysis with a leakage model. In: Cryptographic Hardware and Embedded Systems—CHES 2004. Berlin: Springer, 2004. 16–29

7 Bogdanov A, Kizhvatov I. Beyond the limits of DPA: combined side-channel collision attacks. IEEE Trans Comput, 2012, 61: 1153–1164

8 Clavier C, Feix B, Gagnerot G, et al. Improved collision-correlation power analysis on first order protected AES. In: Cryptographic Hardware and Embedded Systems—CHES 2011. Berlin: Springer, 2011. 49–62

9 Oswald E, Mangard S, Herbst C, et al. Practical second-order DPA attacks for masked smart card implementations of block ciphers. In: Topics in Cryptology—CT-RSA 2006. Berlin: Springer, 2006. 192–207

10 Chair S, Rao J R, Rohatgi P. Template attacks. In: Cryptographic Hardware and Embedded Systems—CHES 2002. Berlin: Springer, 2003. 13–28

11 Biham E, Shamir A. Differential fault analysis of secret key cryptosystems. In: Advances in Cryptology—CRYPTO'97. Berlin: Springer, 1997. 513–525

12 Ege B, Eisenbarth T, Batina L. Near collision side channel attacks. In: Selected Areas in Cryptography—SAC 2015 Cryptology. Berlin: Springer. 2015. 277–292

13 Ye X, Chen C, Eisenbarth T. Non-linear collision analysis. In: Radio Frequency Identification: Security and Privacy Issues. Berlin: Springer, 2014. 198–214

14 Li Y, Sakiyama K, Gomisawa S, et al. Fault sensitivity analysis. In: Cryptographic Hardware and Embedded Systems, CHES 2010. Berlin: Springer, 2010. 320–334

15 Moradi A, Mischke O, Eisenbarth T. Correlation-enhanced power analysis collision attack. In: Cryptographic Hardware and Embedded Systems, CHES 2010. Berlin: Springer, 2010. 125–139

16 Moradi A, Mischke O, Paar C, et al. On the power of fault sensitivity analysis and collision side-channel attacks in a combined setting. In: Cryptographic Hardware and Embedded Systems—CHES 2011. Berlin: Springer, 2011. 292–311

17 Wang A, Chen M, Wang Z Y, et al. Fault rate analysis: breaking masked AES hardware implementations efficiently. IEEE Trans Circ Syst, 2013, 60: 517–521

18 Ren Y T, Wang A, Wu L J. Transient-steady effect attack on block ciphers. In: Cryptographic Hardware and Embedded Systems—CHES 2015. Berlin: Springer, 2015. 433–450

19 Wang Q, Wang A, Wu L J, et al. Template attack on masking AES based on fault sensitivity analysis. In: Proceedings of IEEE International Symposium on Hardware Oriented Security and Trust (HOST 2015), Washington, 2015. 96–99

20 Mangard S, Aigner M, Dominikus S. A highly regular and scalable AES hardware architecture. IEEE Trans Comput, 2003, 52: 483–491

21 Canright D. A very compact S-box for AES. In: Cryptographic Hardware and Embedded Systems—CHES 2005. Berlin: Springer, 2005. 441–455

22 Paar C. Efficient VLSI architectures for bit-parallel computation in Galois fields. Dissertation for Ph.D. Degree. Essen: University of Essen, 1994

23 Rudra A, Dubey P K, Jutla C S, et al. Efficient Rijdael encryption implementation with composite field arithmetic. In: Cryptographic Hardware and Embedded Systems—CHES 2001. Berlin: Springer, 2001. 171–184

24 Morioka S, Satoh A. An optimized S-box circuit architecture for low power AES design. In: Cryptographic Hardware and Embedded Systems—CHES 2002. Berlin: Springer, 2003. 172–186

25 Canright D, Batina L. A very compact "perfectly masked" S-box for AES. In: Applied Cryptography and Network Security. Berlin: Springer, 2008. 446–459

26 Genelle L, Prouff E, Quisquater M. Thwarting higher-order side channel analysis with additive and multiplicative maskings. In: Cryptographic Hardware and Embedded Systems—CHES 2011. Berlin: Springer, 2011. 240–255

27 Kim H, Hong S, Lim J. A fast and provably secure higher-order masking of AES S-box. In: Cryptographic Hardware and Embedded Systems—CHES 2011. Berlin: Springer, 2011. 95–107

28 Endo S, Sugawara T, Homma N, et al. An on-chip glitchy-clock generator for testing fault injection attacks. J Cryptogr Eng, 2011, 1: 265–270