Clockwise Collision Analysis – Overlooked Side-Channel Leakage Inside Your Measurements

Yang Li, Daisuke Nakatsu, Qi Li, Kazuo Ohta, and Kazuo Sakiyama

Department of Informatics, The University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
{liyang,nakatsu_d,riki}@ice.uec.ac.jp, {ota,saki}@inf.uec.ac.jp

Abstract. This paper presents a new side-channel attack technique called *clockwise collision* analysis. For the cryptographic implementations using synchronous digital circuit with a loop architecture, signal transitions as well as the side-channel leakage relates to not only the input data in the current cycle, but also the status in one-cycle before. The clockwise collision utilizes the fact that little computation is required in the second clock cycle when the inputs for two consecutive clock cycles are the same. In contrast, the previously known *computational collision* utilizes the fact that the computation of the same input value leads to similar side-channel leakage. By experimentation, we demonstrate the feasibility and vulnerability for this novel clockwise collision analysis both by injecting faults and by analyzing the power consumption.

Keywords: Side-channel attack, clockwise collision, fault, power consumption

1 Introduction

In the recent decade, the emerge of side-channel attacks (SCAs) are considered as the most practical attacks that threat the security of some electronic devices, e.g., smart cards. The side-channel attacks are based on the information leakage through power consumptions [5], timing [4], electro-magnetic wave [3] and so on. The essence of these attacks comes from the difference of the signal transitions for the physical circuit due to the difference of the proceeded data. Thus, the side-channel leakages such as the power consumption are data-dependent. Based on the data-dependence of these leakages, attackers can statistically process the measurements to distinguish some special features with a correct partial key guess.

For the active attacks, fault-based attacks have also been massively researched. The most famous one is the differential fault analysis (DFA), that was proposed in 1996 [1]. The DFA attack has been demonstrated to be very effective to attack block ciphers such as AES [10]. Except from DFA, safe error attack (SEA) [14] and differential behavioral analysis (DBA) [12] are also well-known fault-based attacks. At CHES 2010, fault sensitivity analysis (FSA) attack was proposed in which the fault injections are used to measure the critical path delay of the S-box

calculation [6]. Later, the attack has been improved by combining it with the distinguisher from [8] and breaks all the AES implementations on SASEBO-R [9]. SASEBO-R is the LSI embedded type R of the Side-channel Attack Standard Evaluation Board (SASEBO-R) [11].

As far as we know, the previous SCAs are utilizing the data-dependency of the signal transitions in crypto circuit, more specifically the dependency on the processed data in the current clock cycle. However, not only the current processed data that the signal transitions as well as the side-channel leakage relates to, but also the processed data in one-cycle before is related. The processed data in one-cycle before determines the final status in the last cycle, which is the initial status of the calculation in the current cycle.

This paper discusses and verifies how to effectively involve the initial circuit status into the side-channel attack, and how it affects the current existing research results. Specifically, we focus on the case, when for a small component (e.g., S-box), the input data for two consecutive clock cycles collide, which we call clockwise collision. Due to the final status of the 1st clock cycle is already the calculation result for the 2nd clock cycle, we expect few signal transitions occur in the 2nd clock cycle. Therefore, little dynamic power is consumed and a setup time violation is difficult to be triggered. Experimentally, we verify our attack concepts, and discuss how the clockwise collision analysis can affect the existing SCAs.

The following of this paper is organized as follows. Section 2 briefly explains some of the related previous work. In Sect. 3, we explain the concept of clockwise collision analysis. Sections 4 and 5 show how we verify the vulnerabilities by experimentations. In Sect. 6, we have some discussions and Sect. 7 concludes the paper.

2 Previous Works

2.1 Power-based Attacks

The most well-known side-channel attack is the differential power analysis (DPA) that was proposed by Paul Kocher in 1999 [5]. In the original DPA attack, power traces are divided into 2 groups according to the value of an intermediate value bit, then the difference of means is used as the distinguisher to identify the correct key guess. Later, the DPA attack was improved to correlation power analysis (CPA) by using a new correlation-based distinguisher and introducing the Hamming distance model [2]. In CHES 2010, the correlation-enhanced collision based distinguisher was introduced by Moradi *et al.* [8].

2.2 Setup-Time Violation based Attacks

The differential fault analysis was first proposed by Biham and Shamir in 1997 [1]. Assuming a fault with some known property can be injected during the calculation, and then the key can be recovered by examining whether the key-guess

based intermediate values have a correct difference. The most practical and used fault model for attacking 128-bit AES is injecting 1-byte random fault before the MixColumns operation in the 8th round [10].

At CHES 2010, a combination of passive and active attacks has been proposed called fault sensitivity attack. In the FSA attack, the setup time violation is used to measure the difference of the critical path delays according to different inputs to the S-box calculation.

3 Concept of Clockwise Collision Analysis

3.1 Iterative Cipher and Clock System

This paper focuses on the cryptographic ciphers with the iterative looping structure, where a small calculation, i.e., a round operation, is repeated by several times to complete the whole calculation. Many modern ciphers are designed to be implemented iteratively, e.g., DES, AES, the modular exponentiation algorithms of RSA. The iteration of the calculation is usually synchronized by a clock signal. Usually, at each positive edge of the clock signal, the calculation result of last clock cycle for the combinational circuit is hold by the register, and be used as the input of the calculation in the coming clock cycle.

3.2 Leakages Relate to Two Inputs

The process of calculation in each cycle for a digital circuit can be considered as a sequence of signal transitions from an initial status to a final status. In a certain clock cycle, the final circuit status is determined according to the input value, which is processed via gates according to the determined logic. Before the next positive edge of the clock signal, all the signals become stable with a calculation result. In the next clock cycle, a new input will be processed by the circuit.

Note that, the final circuit status in a certain cycle is the initial status of the calculation for the next cycle. For the same circuit processing the same data in one clock cycle, the signal transitions inside the circuit can be totally different. The reason is that the initial circuit statuses for two calculations can be totally different. Basically, the initial status is determined by the calculation in one clock cycle before. As a conclusion, the signal transitions and the side-channel leakage relates to the processed data in two consecutive clock cycles.

Most of the current popular SCAs are ignoring involving the initial status of circuit in their attack. In our opinion, there are two reasons for it. First of all, when statistically analyzing the leakages, the secret key can be recovered effectively even without considering the difference caused by the initial status. Second, usually only 1-round calculation near the public information, i.e., plaintext or ciphertext, are involved in the SCAs to restrict the key search space in the post analysis. If 2 rounds' calculations are involved in the attack, more bits of the secret key are required to calculate the related intermediate values.

Therefore, the computational complexity of the "divide and conquer" and "guess and determine" attack procedure increases greatly.

However, information theoretically, the ignorance of information inside leakage means the deficiency of the attack efficiency. The secret key should be able to be recovered with fewer measurements from the cryptography device when more information can be exploited in the post analysis. In this paper, on the understanding the mentioned difficulties, we try to practically exploit the possibility and benefits of involving two cycles' data in the SCAs.

3.3 A Special Case: Clockwise Collision

In this paper, we consider how the side-channel attack can exploit the information about the input for the previous clock cycle. Since this new leakage information has been overlooked in the previous attacks, we expect the enhancement of the existing side-channel attacks.

As a first step, we consider a special case where the processed data for two consecutive clock cycles are the same, which we call a clockwise collision. In this case, the initial status of the current clock cycle is exactly what the calculation wants to achieve. In other words, the calculation is done before the starting. As a result, we expect some special side-channel leakage is distinguishable and can be related to a practical secret key recovery.

In this paper, we consider the clockwise collision analysis both by active attacks (fault-based attacks) and passive attacks (power-based attacks).

- For the fault-based attack, we expect that the setup-time violation is very difficult to be triggered in the second cycle for a clockwise collision. Due to the clockwise collision, there are little signal transitions during the second cycle. The wires connecting the registers have the stable correct value right after the beginning of the second cycle. As a result, unless the period between two positive clock edges is shorter than the timing of several XOR calculations, the setup violation is hard to be triggered.
- For the power-based attack, we expect that the power consumption for the second clock cycle is lower when clockwise collision occurs. Since the initial status of the circuit is already the final result, few signal transitions occurs, thus little dynamic power is required to be consumed. On the other hand, for cryptographic components designed for the confusion, e.g., S-box, even a little change of the input will cause the bit-flip for almost half of the output bits, which leads to a big amount of power consumption. We expect the clockwise collision can be detected by observing whether the power consumption is lower than normal.

For the following two sections, we verify our attack concept step by step.

4 Fault-based Clockwise Collision Analysis

4.1 Experiment Platform and Setup

We first choose the AES-comp implemented in the LSI on SASEBO-R to perform the following experiments. This AES have a 128-bit data path with 16 parallel S-boxes. These S-box components are implemented using a multiplication inversion circuit defined over the composite field GF(((2²)²)²) and don't have any side-channel attack countermeasures. For AES-comp, each AES round requires a clock cycle to finish, so that AES-128 requires 10 cycles to finish.

As shown in the middle of Fig. 1, by using **SB** to denote the S-box circuit, we focus on 4 S-boxes, i.e., **SB0**, **SB1**, **SB2** and **SB3**, in the final AES round. We use C, K and I to denote the ciphertext, round key and the round input, respectively. The superscript and subscript are used to denote the byte position and round number, respectively. The relationship between the byte position and the AES block state is shown in the left part of Fig. 1. For **SB0**, the condition for its clockwise collision is $I_9^0 = I_{10}^0$, which occurs with probability 1/256.

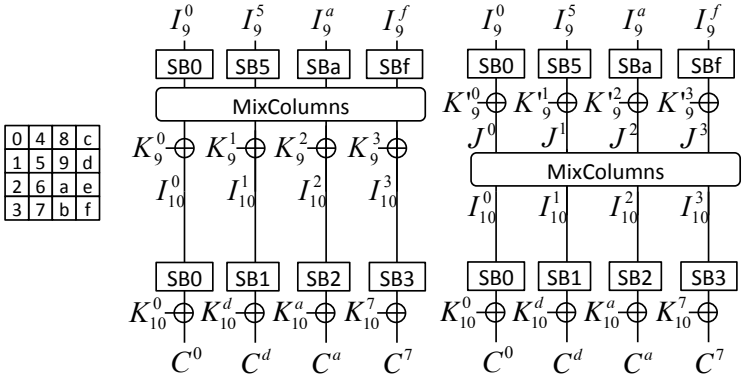


Fig. 1. Model of 4 bytes in the final 2 AES rounds.

In order to simplify the following analysis, we exchange the sequence of the MixColumns and AddRoundKey in the 9th round. Therefore the added 9th round key becomes K'^0_9, K'^1_9, K'^2_9 and K'^3_9 that satisfy $K'^0_9||K'^1_9||K'^2_9||K'^3_9 = MC^{-1}(K_9^0||K_9^1||K_9^2||K_9^3)$. For the clockwise collision at **SB0**, i.e., $I_9^0 = I_{10}^0$, we have $S(I_9^0) = S(I_{10}^0)$, thus $J^0 \oplus K'^0_9 = C^0 \oplus K_{10}^0$. As a useful conclusion, when the clockwise collision occurs for **SB0**, $J^0 \oplus C^0$ is a fixed value as $K'^0_9 \oplus K_{10}^0$. The similar relationship can be found for other S-boxes according to its position.

In order to perform a fault injection at a precise round operation, we use a similar fault injection mechanism used in the fault sensitivity analysis. A fully controllable short clock cycle is provided to the target round operation in the LSI core. However, all the rest clock cycles as long as the LSI interface part are given a normal clock.

4.2 Verification of Attack Concept

Two Clocks' Data Dependence As a first step, we use the faulty behaviors to verify whether the signal transitions of S-box calculation relate to both the currently processed data and the input one-cycle before. From Fig. 1, we see that the circuit of S-box doesn't directly connect to the register, but to the MixColumns operation and the AddRoundKey (several XOR) operation. Only for the final round, the MixColumns calculation is passed.

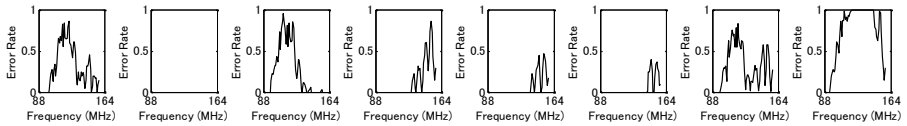
Specifically, we focus on the most significant S-box **SB0** due to its unchanging position for the ShiftRows operation. We want to verify that if both I_9^0 and I_{10}^0 are the same for two plaintexts, the signal transitions for **SB0** in the 10th round are very similar. On the other hand, for two plaintexts that leads to different I_9^0 but the same I_{10}^0, totally different signal transitions for **SB0** may occur.

The faulty ciphertexts under the setup-time violations are used to observe the signal transitions inside the S-box. Denote the S-box output by $S(I_{10}^0)$, we have that $C^0 = S(I_{10}^0) \oplus K_{10}^0$. Then a setup-time violation can be triggered at the final AES round using a clock cycle with an abnormal short period. In this case, the faulty value is coming from the S-box calculation. The value K_{10}^0 should be correct since its value comes directly from registers. While the S-box calculation has a much longer critical path delay, therefore, the faulty ciphertext C' is caused by a faulty output of S-box $S'(I_{10}^0)$, thus $C' = S'(I_{10}^0) \oplus K_{10}^0$. The difference between C and C' (ΔC) equals to the difference between $S(I_{10}^0)$ and $S'(I_{10}^0)$ ($\Delta S(I_{10}^0)$). Thus, the change of $\Delta S(I_{10}^0)$ against faulty clock with different frequencies can be observed from ΔC. That could be an indicator of the internal signal transitions inside the S-box calculation.

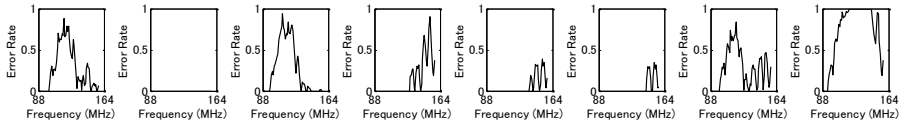
In our experiment, we gradually change the clock frequency from 88MHz to 164MHz for the glitch cycle, and for each step of frequency change, we repeat the calculation for 100 times. Thus, we can calculate the error (bit-flip) rate for each bit for each ciphertext byte. Here we show 4 groups of such results in Fig. 2. Each row in Fig. 2 have 8 sub-figures corresponding to each bit of a ciphertext byte, from left to right is from the MSB to the LSB.

Note that, the error rate is zero at a high frequency fault since the LSI ignores these clock glitches. The 1st and 2nd row in Fig. 2 correspond to the faulty behavior for two plaintext that both leads the 9th and 10th round inputs be 0x$B1$ and 0x$B5$. We can see that the faulty behaviors for all the bits are very similar to each other. On the other hand, if only the S-boxes input for the final round are same (the second row and the third row), the difference of faulty behaviors could be really huge. This result proves that the signal transitions inside the circuit are related to the processed data for two consecutive clock cycles.

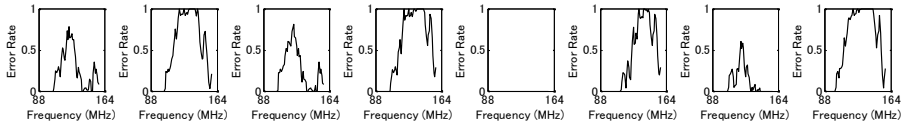
Free from Setup-time Violation Following the experimentation, we'd like to test the ΔC when the clockwise collision occurs. As shown in Fig. 2(d), we can see that when the clockwise collision occurs, the setup-time violation fault cannot be triggered in our experiment. Some plaintext may leads to a S-box calculation that is not very sensitive to a fast clock. But only when the clockwise collision



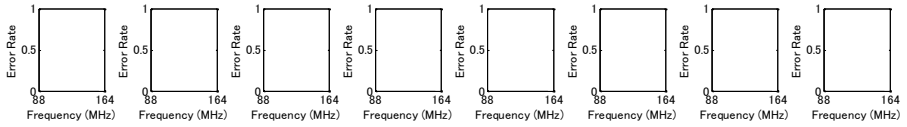
(a) 9th input: 0x$B1$, 10th input: 0x$B5$



(b) 9th input: 0x$B1$, 10th input: 0x$B5$



(c) 9th input: 0xDA, 10th input: 0x$B5$



(d) 9th input: 0x79, 10th input: 0x79

Fig. 2. Error rate for different clocks from MSB to LSB (from right to left).

occurs, the fault cannot be injected for all the tested clock frequencies. This could be used to detect the clockwise collision and relate to a key recovery attack.

4.3 Attack Scenario

We explain how to apply an attack to AES with a very loose fault model.

Fast Collision Detection When considering the practical attack, the previous experiments cost too much for too many times of fault injections. One can test the existence of the clockwise collision by calculating every plaintext twice. One is for the faulty-free ciphertext, and the other one is with a very strong fault. Under this strong fault, most of the S-boxes will generate a faulty value, while still no fault occurs for the clockwise colliding S-boxes.

Our experiment used $100,000$ random plaintexts, which means about $100,000 * 16/256 \simeq 6400$ clockwise colliding S-boxes occurs. We first use a normal clock to obtain the fault-free ciphertexts. Then a faulty clock at 148 MHz is used to obtain the faulty ciphertexts. In Table 1, we show the distributions of colliding and faulty S-boxes. The most important thing is that when the clockwise collision occurs, no fault was injected.

Table 1. Clockwise collision and faulty S-box distributions

Colliding/Faulty or not	# of occurrences (proportion)
Colliding S-boxes without fault injected	6341(0.4%)
Colliding S-boxes with fault injected	**0(0.0%)**
No collision with fault injected	1483120(92.7%)
No collision without fault injected	110539(6.9%)

Key Recovery After the ciphertexts with clockwise collision are known, the key recovery procedure is simple. For 2^{32} key candidates for 4 bytes of the final round key, attackers calculate the value of J^0, and checks whether the $J^0 \oplus C^0$ is a fixed value for the ones with collision. After repeating the same procedure for 4 groups of key bytes, both K_9 and K_{10} can be recovered. One can further use the key schedule to verify the recovered key.

From Table 1, one may notice that for the fault-free S-boxes, lots of them don't correspond to colliding S-boxes. We give 3 solutions to show that it is not a problem to the attack. First, one can repeat performing the fault injections to the fault-free S-boxes for several times to filter the non-collision cases out. Second, one can check whether $J^0 \oplus C^0$ covers all the 256 values for the ciphertexts without collision. Third, one can check whether $J^0 \oplus C^0$ has a frequently occurring value (with a probability larger than 1/256) for the ciphertexts with collisions.

Note that the fault injection used in this attack can trigger most of the S-boxes to be faulty. As far as we know, there is no DFA attack can respond to this type of fault model. And it is a loose model for attackers to achieve since there is no requirement for the faulty value and the number of faulty S-boxes.

5 Power-based Clockwise Collision Analysis

In this section, we analyze how the clockwise collision be utilized based on power.

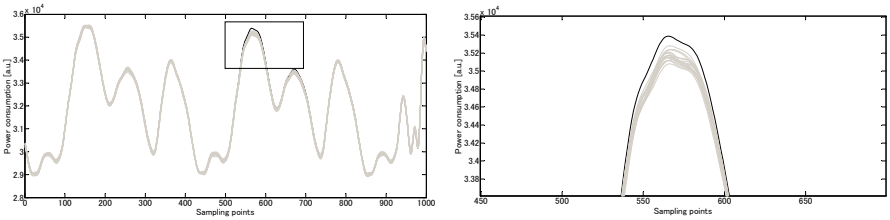
5.1 Experiment Platform and Setup

In our power-based attacks, we use the same AES-comp implementation to verify our attack. However, we used the power consumption from the FPGA-based implementations. The first set of data is collected using SASEBO-G, and the second set of data is from the DPA contest version 2[13].

Generally, the power-based attacks that belong to passive attacks is weaker than the active attacks such as fault-based ones. Especially, the AES-comp implementation we attacked is with a 128-bit data path, so the measured power consumption always corresponds to the sum of that for 16 S-boxes in parallel. The power consumption for each S-box cannot be directly separated and analyzed independently.

5.2 Verification of Attack Concept

To verify the attack concept, we need to see whether or not the power consumption is low enough to be detected for the 2nd clock cycle of a clockwise collision. In the first experiment, we obtained the power traces for $100,000$ random plaintexts from SASEBO-G. Based on the known plaintexts and key, we can check whether a clockwise collision occurs for a certain byte position of a certain plaintext. Thus, for each byte position, we collect the power traces that corresponding to the collision and we calculate the mean power trace for them. We also calculated the mean power trace for all the power traces, and plot 17 mean power traces in Fig. 3. As shown in Fig. 3(b), we can see that 16 mean traces corresponding to collisions are clearly lower than the mean of all traces at the timing for the final AES round. Note that we also verified the similar phenomenon for the first two rounds of AES.



(a) Last 2 rounds (b) Magnified part

Fig. 3. Mean of all traces (black) and mean of colliding traces for each S-box (gray).

5.3 Attack Scenario

In the experiment shown in Sec. 5.2, we have shown that the power consumption is statistically lower than normal ones when the clockwise collision occurs. The next question is whether or not this phenomenon can be related to a practical attack, especially for a set of noisy data.

A Straight Forward Attack Before proposing a practical attack scenario for power-based clockwise collision analysis, we first give a straightforward attack algorithm based on the last power-based experiment.

1. Collect power traces for random plaintexts.
2. Have a key guess of 5 key bytes, e.g., $K_{10}^0, K_{10}^d, K_{10}^a, K_{10}^7, K_9^0$, calculate 1 byte of the 9th round input e.g., I_9^0. Select all the plaintexts that lead to the clockwise collisions, e.g., $I_9^0 = I_{10}^0$. Calculate the difference between the mean power traces for the traces without the clockwise collisions and the ones with.

3. Repeat step 2 for 2^{40} key candidates, choose the key guess corresponding to the largest difference as the correct key guess.

This straightforward attack algorithm have three problems.

First of all, for 2^{40} key candidates, the correct key may not correspond to the lowest mean of power consumption. To answer this question, we did another experiment to show that the probability is low for a random set of power traces leads to a lower mean power trace. In our experiment, we compute the mean power trace for $400 \simeq 100000/256$ randomly selected power traces and compare it with the mean trace for the colliding traces. We repeat this procedure for 10000 times and find that none of the random set of power traces have a lower mean than that for the colliding traces. In fact, given some statistical status of the power traces, one can mathematically estimate the probability that the mean for a random selected set of power traces have a lower mean than that of the colliding traces.

Second, 2^{40} is a comparably large complexity for a practical side-channel attacks. For each key guess, attacker needs to find out which plaintexts are corresponding to the clockwise collisions.

The final problem is also the most important one. When given a set of random power traces, attackers always first use the exiting powerful attacks, e.g., DPA, CPA. Note that, for power-based attacks, previous attacks can be applied to these power traces, while for fault based attacks, none of the existing DFA attacks can be applied to the fault model used in the clockwise collision analysis. Therefore, it is reasonable that the power-based clockwise collision analysis is used by being combined with previous power analysis.

Another important fact is that clockwise collision analysis uses a different information source from the one used in the previous power analysis. Previous power analyses are based on the computational collision, which means the side-channel leakages are similar when the processed data are the same. While clockwise collision analysis focus on the physical signal transitions for two consecutive clocks. Both methods can reveal the sensitive information from the measured data, but two information sources do not overlap. In other words, for a set of limited measurements, it is most information-theoretically efficient if both the traditional computational collision based attack and the clockwise collision based attack are applied.

Compared to traditional attacks, the clockwise collision relates to the data for two clocks, so that it is related to more intermediate values and the more secret bits. This fact brings more complexity to apply the clockwise collision analysis but also benefits when it is used to help verifying and identifying the secret key.

A Practical Attack Procedure Here we propose a reasonable attack algorithm that combines traditional attacks (e.g., CPA attack) and the clockwise collision based attacks.

1. For a set of power traces, attackers first perform the traditional attack, e.g., CPA attack. For each byte of the final round key, attackers obtain a rank

of credibility for each key candidate according the results from transitional attack, e.g., the correlation coefficient in the CPA attack.
2. Focus on 4 key bytes that corresponds to a 9th round MixColumns operation, e.g., K_{10}^0, K_{10}^d, K_{10}^a, K_{10}^7, attacker test the key candidates in the sequence of the credibility from high to low.
3. For each 4-byte key candidate, attackers can check 4 byte of clockwise collision. Following the example of recovering $K_{10}^0, K_{10}^d, K_{10}^a, K_{10}^7$, attacker first calculate the value of J^0 for each ciphertext. Then all the traces can be divided into 256 groups according to the value of $J^0 \oplus C^0$ and the mean trace for each group can be calculated. For the correct 4-byte key candidate, the value of $J^0 \oplus C^0$ for the colliding traces is a certain fixed value. So we expect that the correct key guess can be distinguished by check whether 1 out of the 256 mean traces has an obvious lower power consumption. In contrast, for the wrong key guess, all 256 mean traces have the similar power consumptions. Hereafter, we call this distinguisher the lowest-mean test. For each 4-byte key candidates, this lowest-mean test can be applied by 4 times corresponding to 4 byte positions.

Application to DPA Contest Data We applied our attack algorithm to the DPA contest data. For several sets of 20000 power traces, we first apply a straightforward CPA attack to obtain the credibility of each key candidate for each key byte. For about half of the data sets, the 128-bit key can be identified by CPA attack directly, while the rest half doesn't. Using the clockwise collision analysis, the correctness of the key can be verified without knowing a plaintext-ciphertext pair. For each 4 final-round key bytes, from high credibility to low one, attackers can perform the lowest-mean test to verify each key candidate. Also the recovered 9th round key bytes can be used to verify the key correctness using key scheduling.

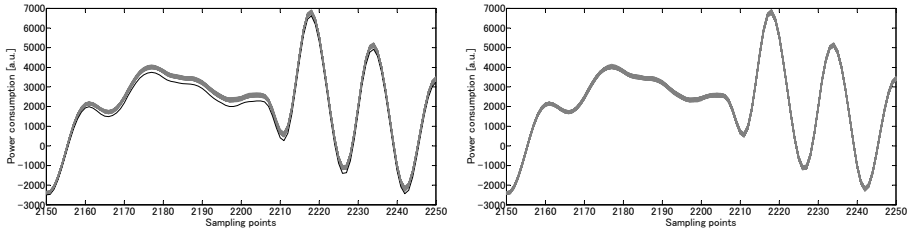
As shown in Fig. 4, in the case of correct key guess, 1 mean trace is obviously lower than others. While for the wrong key guess, all the mean traces are similar to each other. In the lowest-mean test, we can use the difference between the mean of all traces and the mean of the lowest mean trace as a distinguisher. In Fig. 5, we show the evolution of this lowest-mean distinguisher against the number of used traces. We can see that about 4000 traces is enough to identify the correct key guess from other 624 key guesses in this example.

6 Discussions

This section explains some interesting research topics and open questions we realized in this research.

6.1 Mathematical Analysis of Attack Efficiency

Similar to previous side-channel attacks, it is possible to estimate the necessary number of the power traces for the power-based clockwise collision analysis.



(a) Correct key guess (b) Wrong key guess

Fig. 4. Mean traces for 256 groups of power traces according to $J^0 \oplus C^0$.

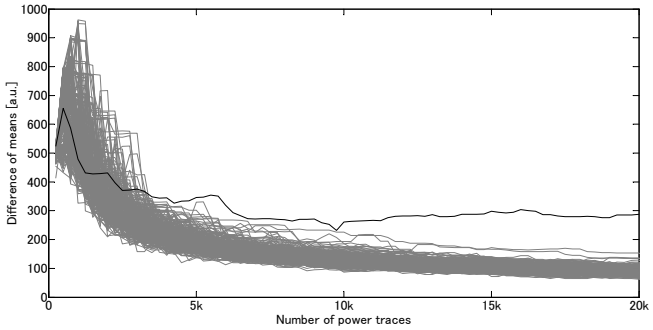


Fig. 5. The evolution of the lowest-mean test against the number of used traces.

To explain the basic idea, the probability distribution of all the traces and the colliding traces are plot in Fig. 6.

Consider the power traces follows a normal distribution with mean μ and variance σ^2, and N random traces are used in the attack. Denote the mean of the colliding traces by μ_1, where $\mu_1 < \mu$. Then based on Chebyshev's inequality, the probability that the mean of randomly selected $n = \frac{N}{256}$ traces being smaller than μ_1 is bounded as shown in

$$Pr(\frac{\sum_{i=1}^{n} X_i}{n} < \mu_1) \leq \frac{\sigma^2}{2(\mu - \mu_1)^2 n}. \tag{1}$$

Based on Eq. 1, one can see that improving the accuracy of the attack comes from reducing the data noise and increasing the amount of data.

6.2 Push the Limits of Clockwise Collision Analysis

As a general vulnerability of the synchronous digital circuit with loop architecture, this paper first brings this topic to the academic. There are many possible approaches to push the limits of the clockwise collision based attacks. We list a few of them here.

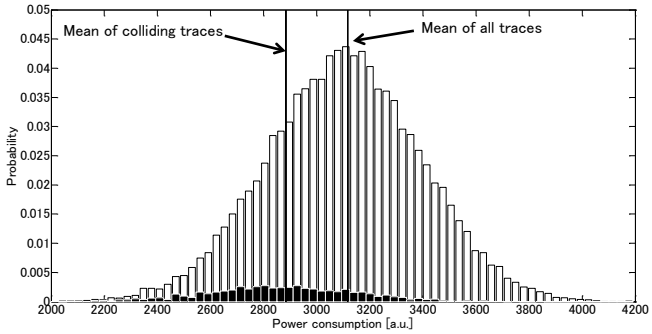


Fig. 6. Probability distribution for all the traces (white) and for colliding traces (black).

Template-based Fault Attack Using 1 Plaintext Without Fault Model In Sect.4, from Fig. 2, we expect the success of a template-based fault attack. Attacker builds 2^{16} templates to record the faulty behavior for each bit of the ciphertext byte for all the possible two round inputs. Then by testing a random plaintext, attackers can obtain the possible two round inputs for each S-box. Then key schedule can be used to verify the correctness of the recovered keys.

Possibility of New Record for DPA Contest As mentioned, the clockwise collision based attacks use a different information source from that used in the previous power-based attacks. So no matter what kind of techniques are used to reach the latest DPA contest record, the clockwise collision analysis can exploit more secret information from the power traces. Therefore, one can expect the possibility of a new DPA contest record by combining the new proposed attack with the previous techniques.

Possibility of Vulnerability of Existing SCAs Countermeasures It is also important to consider whether the clockwise collision analysis can be applied to the implementations with SCAs countermeasure. In our opinion, it is possible to apply the attack on some of the masking countermeasures. When the unmasked values collide, there is a possibility that both the masked values and the masks have collision simultaneously. On the other hand, if the unmasked values don't collide, the masked values and the masks cannot have collision at the same time. This could lead to a difference of the fault injection error rate and lead to a vulnerability. By far, we consider a random precharge can be used a countermeasure to the proposed attacks.

7 Conclusion

In this work, we discussed a general but overlooked vulnerability called clockwise collision. The intuition is to relate the initial circuit status to the existing side-channel attacks, and this paper focused on the case when the inputs for two

consecutive clocks collide. We successfully demonstrate our attack concept and attack approach both for active and passive attacks. In the discussion, we show lots of open questions for the future work. We expect that more vulnerability exists for current cryptographic implementations when two consecutive inputs are considered in the SCAs.

References

1. E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In B. S. K. Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 1997.
2. E. Brier, C. Clavier, and F. Olivier. Correlation power analysis with a leakage model. In M. Joye and J.-J. Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.
3. K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, D. Naccache, and C. Paar, editors, *CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.
4. P. C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In N. Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.
5. P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In M. J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.
6. Y. Li, K. Sakiyama, S. Gomisawa, T. Fukunaga, J. Takahashi, and K. Ohta. Fault sensitivity analysis. In Mangard and Standaert [7], pages 320–334.
7. S. Mangard and F.-X. Standaert, editors. *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*. Springer, 2010.
8. A. Moradi, O. Mischke, and T. Eisenbarth. Correlation-enhanced power analysis collision attack. In Mangard and Standaert [7], pages 125–139.
9. A. Moradi, O. Mischke, C. Paar, Y. Li, K. Ohta, and K. Sakiyama. On the power of fault sensitivity analysis and collision side-channel attacks in a combined setting. In B. Preneel and T. Takagi, editors, *CHES*, volume 6917 of *Lecture Notes in Computer Science*, pages 292–311. Springer, 2011.
10. G. Piret and J.-J. Quisquater. A differential fault attack technique against spn structures, with application to the aes and khazad. In C. D. Walter, Çetin Kaya Koç, and C. Paar, editors, *CHES*, volume 2779 of *Lecture Notes in Computer Science*, pages 77–88. Springer, 2003.
11. Research Center for Information Security (RCIS). Side-channel attack standard evaluation board (SASEBO). `http://www.rcis.aist.go.jp/special/SASEBO/CryptoLSI-en.html`.
12. B. Robisson and P. Manet. Differential behavioral analysis. In P. Paillier and I. Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 413–426. Springer, 2007.
13. Telecom ParisTech french University. DPA Contest v2 2009/2010: Introduction. `http://www.dpacontest.org/v2/index.php`.
14. S.-M. Yen and M. Joye. Checking before output may not be enough against fault-based cryptanalysis. *IEEE Trans. Computers*, 49(9):967–970, 2000.