Group Verification Based Multiple-Differential Collision Attack

Changhai Ou[1,2], Zhu Wang[1(✉)], Degang Sun[1(✉)], Xinping Zhou[1,2], and Juan Ai[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{ouchanghai,wangzhu,sundegang,zhouxinping,aijuan}@iie.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China

Abstract. Bogdanov and Kizhvatov proposed the concept of test of chain, but they didn't give a practical scheme. Wang et al. proposed fault tolerant chain to enhance test of chain and gave a practical scheme. However, the attack efficiency of Correlation enhanced Collision Attack (CCA) is much lower than that of Correlation Power Analysis (CPA). A combination of CCA and CPA in fault tolerant chain proposed by Wang et al. may be unreasonable. Most importantly, when the threshold Thr_{Δ} introduced in Sect. 2.3 is large, the key recovery becomes very complex. Fault tolerant chain is unapplicable to this situation. In order to solve these problems, we propose a kind of new chain named group verification chain in this paper. We combine our group verification chain with MDCA and propose Group Verification based Multiple-Differential Collision Attack (GV-MDCA). Experiments on power trace set downloaded from the website DPA *contest v4* show that our group verification chain significantly improves the efficiency of fault tolerant chain.

Keywords: Group verification · Group verification chain · Collision attack · MDCA · GV-MDCA · DPA *contest v4* · Side channel attack

1 Introduction

There exist many kinds of leakages such as power consumption [7] and electromagnetic [2] when the cryptographic devices are on operation. Side channel attacks can be used to efficiently recover the key and pose serious threats to cryptographic implementation security. Side channel collision attack was firstly introduced in [13] against DES and extended in [12]. Nonlinear S-boxes are usually chosen as attack points. The linear parts such as MixColumns of AES, are also targeted in collision attack [12].

One advantage of collision attack is that it can help conquer the random masking of some AES implementations [3] and DES implementations [6]. Moradi et al. proposed MDCA based on binary voting and ternary voting [4]. Subsequently, he proposed CCA [9], which established the relationship among several key bytes using the collisions between different S-boxes. It is very efficient for

K.-Y. Lam et al. (Eds.): ICICS 2016, LNCS 9977, pp. 145–156, 2016.
DOI: 10.1007/978-3-319-50011-9_12

CCA to attack the masking schemes such as Rotating S-boxes Masking (RSM) [10]. CCA directly uses the correlation coefficients between two columns of two different S-boxes, it doesn't relay on any hypothesis power leakage model. In 2012, Bogdanov and Kizhvatov combined CPA with collision attack, which was more efficient than both stand-alone CPA and collision attack [5]. Moreover, the concept of **test of chain** was given. However, there was no practical scheme given in their paper. Wang et al. proposed fault tolerant chain in [15]. As far as we know, fault tolerant chain is the only one practical scheme to enhance test of chain. So, in this paper, we just compare our scheme with fault tolerant chain.

Let k_a and k_b denote the a^{th} and the b^{th} key bytes respectively. Taking AES for example, CCA considers the relationship between two key bytes. However, any key byte falling outside the threshold Thr_k will result in very complex key recovery, since the attacker does not know which one is error. The scheme of Wang et al. can identify the specific error key byte. However, the scheme may be not a good one. Firstly, the efficiency of CCA is much lower than that of CPA, a combination of CCA and CPA is unreasonable. Secondly, the threshold Thr_{Δ} ($\Delta_{(k_a,k_b)} = k_a \oplus k_b$) of any two key bytes k_a and k_b is always set to 1, a lot of correct Δ values fall outside the threshold. This leads to failure of key recovery. Thirdly, the scheme uses only one $\Delta_{(k_a,k_b)}$ to identify the value of key byte k_b. If both $\Delta_{(k_a,k_b)}$ and k_b are wrong, but they still satisfy that $k_b = \Delta_{(k_a,k_b)} \oplus k_a$. Then, the scheme of Wang et al. will regard k_b as the correct key byte, which leads to the failure of key recovery. Actually, the probability of this situation is about 10% when $Thr_k = 2$ and reaches more than 70% when $Thr_k = 8$ (see Fig. 4).

In this paper, we propose group verification chain to enhance fault tolerant chain. We then combine MDCA with our group verification chain and propose Group Verification based MDCA (GV-MDCA). Two schemes named Frequency based GV-MDCA (FGV-MDCA) and Weight based GV-MDCA (WGV-MDCA) are given. Our scheme can successfully search the correct key in large thresholds and significantly improve the attack efficiency.

This paper is organized as follows. MDCA, CCA, Bogdanov and Kizhvatov's test of chain and fault tolerant chain proposed by Wang et al. are briefly introduced in Sect. 2. In Sect. 3, group verification chain is introduced. FGV-MDCA and WGV-MDCA are given in this section. Experiments are performed on power trace set *secmatv*1 downloaded from the website DPA *contest v*4 [1] in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 Preliminaries

Bogdanov and Kizhvatov proposed linear collision attack in [5]. AES performs the 16 parallel SubBytes operations within the first round. A collision occurs if there are two S-boxes within the same AES encryption or with several AES encryptions accepting the same byte value as their input. $K = \{k_j\}_{j=1}^{16}$, $k_j \in F_{2^8}$ is the 16-byte subkey in the first round of AES. $P^i = \{p_j^i\}_{j=1}^{16}$, $p_j^i \in F_{2^8}$, are plaintexts, where i = 1,2,... is the number of AES execution. If

$$S(p_{j_1}^{i_1} \oplus k_{j_1}) = S(p_{j_2}^{i_2} \oplus k_{j_2}), \tag{1}$$

a collision happens. The attacker obtains a linear equation

$$p_{j_1}^{i_1} \oplus p_{j_2}^{i_2} = k_{j_1} \oplus k_{j_2} = \varDelta_{(k_{j_1}, k_{j_2})}. \tag{2}$$

Each equation is named a step of a chain [5].

2.1 Multiple Differential Collision Attack

The attacker will encounter a problem when the side channel collision theory is used in side channel attack. That is, how to detect collisions. Actually, the attacker can do this by comparing power traces of two S-boxes. For example, Bogdanov set a differential threshold in his MDCA. If the correlation coefficient of these two power traces was larger than the differential threshold, he deemed that a collision happened.

2.2 Correlation Enhanced Collision Attack

Moradi et al. divided power trace sections of each S-box into 256 classes according to their plaintext α from 0 to 255 [9]. Then, they averaged the power traces in each class and obtained 256 averaged power traces. Let M_j^α denote the averaged power trace of the j^{th} Sbox where the j^{th} plaintext byte are equal to α.

The value $\varDelta_{(k_a, k_b)} = k_a \oplus k_b$ is a constant, since the key used in the cryptographic device is constant. Hence, a collision occurs whenever the a^{th} and b^{th} plaintext bytes show the same difference. Moradi et al. guessed the difference $\varDelta_{(k_a, k_b)}$ and verified their guess by detecting all collisions $p_a = \alpha$ and $p_b = \alpha \oplus \varDelta_{(k_a, k_b)}$ for all $\alpha \in GF(2^8)$ [9]. To detect the correct $\varDelta_{(k_a, k_b)}$, they calculated the correlation coefficient of M_a^α and $M_b^{\alpha \oplus \varDelta_{(k_a, k_b)}}$ for all $\alpha \in GF(2^8)$. The correct difference $\varDelta_{(k_a, k_b)}$ of two key bytes k_a and k_b is then given by:

$$\underset{\varDelta_{(k_a, k_b)}}{\mathbf{argmax}} \rho(M_a^\alpha, M_b^{\alpha \oplus \varDelta_{(k_a, k_b)}}). \tag{3}$$

The correlation coefficients are computed for each $\alpha \in GF(2^8)$. The correct $\varDelta_{(k_a, k_b)}$ corresponds to the maximum correlation coefficient.

2.3 Test of Chain

Bogdanov and Kizhvatov defined test of chain in [5]. Suppose that the attacker uses CPA to obtain the 16 guessing key byte sequences $\{\xi_i | i = 1, 2, \cdots, 16\}$ of AES algorithm (as shown in Fig. 1). Specifically, he uses CCA to calculate $\varDelta_{(k_a, k_b)}$ between any two key bytes k_a and k_b. He then sorts correlation coefficients for all possible guess key byte values in descending order.

Each vertical line denotes a sorted guessing key byte. Each black point in Fig. 1 denotes a possible guessing key byte value. Each line from ξ_a to ξ_b

($1 \leq a < b \leq 16$) denotes a step of a chain. For example, the red line from ξ_2 to ξ_3 denotes that the sixth guessing value of the second key byte and the first guessing value of the third key byte are in Thr_k, the corresponding $\Delta_{(k_2,k_3)}$ of these two guessing values is in Thr_Δ, too. Thr_k and Thr_Δ here are defined as the threshold of key byte values and the threshold of Δ. As shown in Fig. 1, there are 10 black points within Thr_k on each vertical line. So, Thr_k is set to 10 here. Thr_Δ is set in the same way. We only consider the guessing values in Thr_k on each list ξ_i. They are the most possible candidates of the key byte k_i.

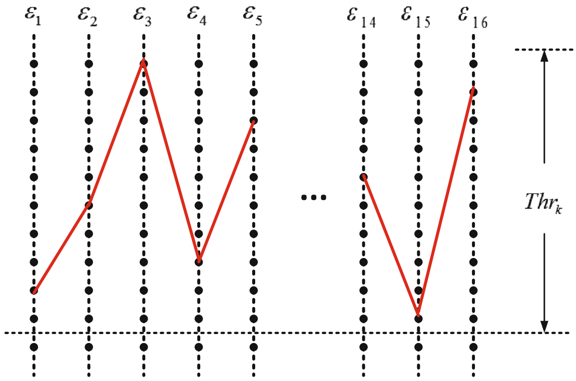

Fig. 1. Bogdanov and Kizhvatov's test of chain.

The chain is accepted if all key bytes of it are within the top m candidates. The chain is rejected if at least one key byte of it falls outside the m top candidates. m here is equal to Thr_k. In order to recover the full 16-byte key, the attacker usually hopes that a chain includes 15 steps as introduced in [5]. However, if a chain includes 15 steps, it is too long. The attacker has to calculate Δ between any two adjacent guessing key bytes. The complexity of computation is larger than exhausting all possible keys in Thr_k. For example, if Thr_k and Thr_Δ are set to 10 and 5 respectively, the attacker has to enumerate 10^{16} guessing keys in Thr_k by using brute-force attack. However, each step of a chain brings extra computation. The attacker has to enumerate all Δs in Thr_Δ. So, the computation complexity becomes $10^{16} * 5^{15}$.

2.4 Fault Tolerant Chain

Bogdanov and Kizhvatov did not give a practical scheme of their test of chain in [5]. The computation complexity of their test of chain is even greater than that of brute-force attack. For a chain, there may be several steps in the path from the free variable to the end. If an error happens in one of these steps, the key bytes computed in the following steps will be wrong, which will result in the failure of attack. Unfortunately, this kind of errors happen with non-negligible probability and lead to low efficiency of Bogdanov and Kizhvatov's attack.

Wang et al. constructed a new chain named fault-tolerant chain [15]. In their scheme, $k_i (i \geq 2)$ depends on only one key byte (i.e. k_1) instead of the other 14 key bytes. There are 15 chains from k_1 to k_i $(i = 2, \cdots, 16)$ (as shown in Fig. 2). Each chain only includes a step. This scheme greatly reduces the computation of test of chain proposed by Bogdanov and Kizhvatov [5]. Specifically, if the attacker enumerates all possible keys in the thresholds, the complexity is only $15 * (Thr_k)^2 * (Thr_\Delta)$ compared to $(Thr_k)^{16} * (Thr_\Delta)^{15}$ of test of chain.

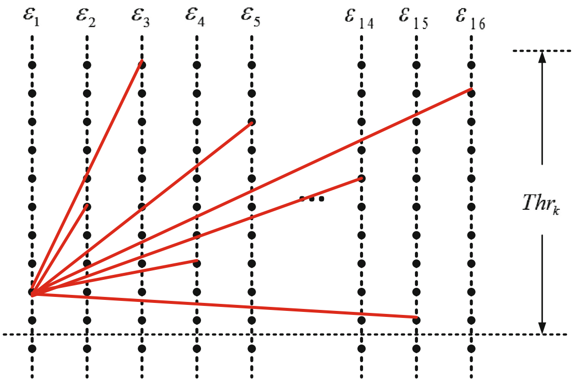

Fig. 2. Fault tolerant chain proposed by Wang et al.

Another advantage of fault tolerant chain is that, if k_i is wrong (under the threshold line), the attacker can still attempt to recover other ones. The error key bytes can still be exhausted. However, the threshold Thr_Δ of CCA is always set to 1 in their paper. Specifically, correlation coefficients of $\Delta_{(k_1,k_i)}$ of two key bytes k_1 and k_i are calculated using CCA. The 256 Δ values are sorted according to these correlation coefficients. Then the Δ corresponding to the maximum correlation coefficient is chosen as the candidate. Other Δ values in the $\Delta_{(k_1,k_i)}$ sequence are not taken into consideration. Actually, enlarging the threshold will lead to very complex key recovery. If $k_i = k_1 \oplus \Delta_{(k_1,k_i)}$ is under the threshold, they deduced that the chain is wrong. Subsequently, exhaustion is performed to find the correct key byte.

Moreover, if k_i and $\Delta_{(k_1,k_i)}$ are wrong and $k_i = k_1 \oplus \Delta_{(k_1,k_i)}$ is still satisfied, the attacker will regard the wrong guessing k_i as the correct key byte value. Actually, this kind of error happens with a high probability and increases with Thr_k. This is the main reason why the success rate declines in Fig. 4.

3 Group Verification Based MDCA

The Thr_Δ is always set to 1 in fault tolerant chain [15]. They did not discuss how to efficiently recover the key when $Thr_\Delta > 1$. In fact, enlarging Thr_Δ will result in very complex key recovery because of very huge key search space. In this case, the scheme of Wang et al. can't be applied any more.

3.1 Group Verification Chain

In this section, we introduce group verification chain, which can be used under the condition that both Thr_k and Thr_Δ are set largely. Group verification here is defined as the mutual verification among key bytes. Let ξ_i^k and $\xi_{\gamma+1}^t$ denote the k^{th} and t^{th} guessing key values in ξ_i and $\xi_{\gamma+1}$. $\Delta_{(i,\gamma+1)}^m$ denotes the m^{th} value in the $\Delta_{(i,\gamma+1)}$ sequence. If the equation

$$\xi_i^k \oplus \xi_{\gamma+1}^t = \Delta_{(i,\gamma+1)}^m \tag{4}$$

is satisfied, then we say that $\xi_{\gamma+1}^t$ can be verified by ξ_i^k. In our group verification chain, we do not care if ξ_i^k, $\xi_{\gamma+1}^t$ and $\Delta_{(i,\gamma+1)}^m$ are the correct key byte values and Δ value. Each candidate value can be verified by guessing values of other key bytes just like voting. When the support of a guessing key byte value is greater than the differential threshold, we deem that this is a good candidate.

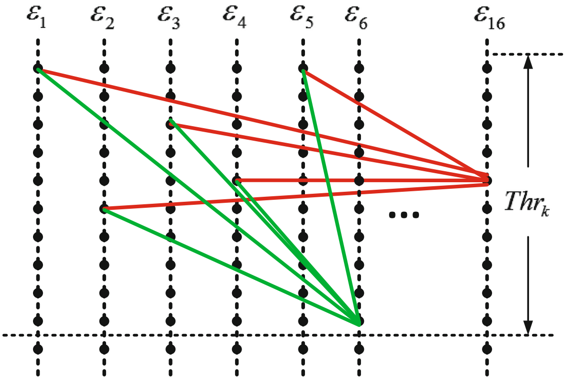

Fig. 3. Group verification chain.

Suppose that we use key byte values in $\xi_1, \cdots, \xi_5$ to verify guessing key byte values in $\xi_6, \cdots, \xi_{16}$ of AES algorithm (as shown in Fig. 3). 120 sequences of $\Delta_{(k_a,k_b)}$ between any two key bytes k_a and k_b are also calculated. The correct key byte values are effectively supported that the Eq. 4 is satisfied for most key values and Δs. Finally, the attacker gets the correct key.

3.2 Frequency Based GV-MDCA

Bogdanov proposed MDCA using binary voting and ternary voting [4], in which multiple difference is used for collision detection between any two S-boxes. Specifically, it is used to compare the power traces of two S-boxes to judge if collision has happened. In this paper, we deem that each S-box can be used as a key byte vote for other S-boxes.

There are 16 guessing key byte sequences $\{\xi_i | i = 1, 2, \cdots, 16\} \in GF(2^8)$ corresponding to S-boxes 1,$\cdots$, 16 of AES algorithm. Let Y^{FGV} denote a decision threshold of possible key byte values. Suppose that we use the $1 \cdots \gamma$ key bytes to verify the $(\gamma + 1)^{th}$ key byte. Then, a Frequency based GV-MDCA (FGV-MDCA) can be defined as:

$$\Psi^{FGV}_{\xi^t_{\gamma+1}} = \begin{cases} 1(\textbf{collision}), & \text{if } \Phi^{FGV}_{\xi^t_{\gamma+1}} > Y^{FGV} \\ 0(\textbf{nocollision}), & \text{if } \Phi^{FGV}_{\xi^t_{\gamma+1}} < Y^{FGV} \end{cases} \tag{5}$$

where $\Psi^{FGV}_{\xi^t_{\gamma+1}}$ denotes that $\xi^t_{\gamma+1}$ is a candidate byte value in the sequence $\xi_{(\gamma+1)}$ and

$$\Phi^{FGV}_{\xi^t_{\gamma+1}} = \sum_{i=1}^{\gamma} \Theta(\xi^k_i, \xi^t_{\gamma+1}). \tag{6}$$

$\Theta(\xi^k_i, \xi^t_{\gamma+1})$ here is defined as

$$\Theta(\xi^k_i, \xi^t_{\gamma+1}) = \begin{cases} 1, & \text{if } \xi^k_i \oplus \xi^t_{\gamma+1} = \Delta^m_{(i,\gamma+1)} \\ 0, & \text{else.} \end{cases} \tag{7}$$

The frequency of the correct byte values will be higher than these of wrong guessing key byte values in FGV-MDCA. This also shows that the correct key byte values obtain more support in the process of mutual verification, which makes them become obvious. The attacker can effectively restore the key by observing the frequencies of key byte values. He does not need to enumerate all possible key values within the threshold.

3.3 Weight Based GV-MDCA

Test of chain and fault tolerant chain introduced in Sect. 2, and our FGV-MDCA introduced in Sect. 3.2 use thresholds Thr_k and Thr_Δ. All possible keys within the thresholds are searched with the same probability. Obviously, this is unreasonable. The key values ranked in the front of $\{\xi_i | i = 1, 2, \cdots, 16\}$ should be enumerated with higher priority. A more accurate and higher efficient key search scheme named Weight Based GV-MDCA (WGV-MDCA) is proposed.

The attacker obtains key byte sequences and Δ sequences. The key byte values in the tops the sequences should be given higher weights. CPA is much more powerful than that of CCA in most cases. If we use moderate number of power traces, most of the correct byte values will be in the top of key byte sequences $\{\xi_i | i = 1, 2, \cdots, 16\}$. However, a number of correct Δ values are not in the top of their corresponding sequences when the same number of power traces are used. So, Δ values here become the most important factor of attack efficiency. So, we weigh key byte values referring to $\varDelta$ sequences. Specifically, $\varTheta(\xi_i^k, \xi_{\gamma+1}^t)$ here is defined as

$$\varTheta(\xi_i^k, \xi_{\gamma+1}^t) = \begin{cases} Thr_{\varDelta} - m, & \text{if } \xi_i^k \oplus \xi_{\gamma+1}^t = \varDelta_{(i,\gamma+1)}^m \\ 0, & \text{else.} \end{cases} \tag{8}$$

For example, if $Thr_{\varDelta}$ is set to 10 and m in Eq. 8 is 6 ($\varDelta_{(i,\gamma+1)}^6$ is the sixth guessing value of the corresponding sequence of $\varDelta_{(i,\gamma+1)}$). Then, $\varTheta(\xi_i^k, \xi_{\gamma+1}^t)$ is 4. By using WGV-MDCA, the difference between the correct key and wrong keys becomes more obvious compared to FGV-MDCA.

3.4 The Differential Threshold

It is very hard to get a good value of Y^{FGV} in both FGV-MDCA and WGV-MDCA. This value is very different in these two schemes. We normalize each reordered sequence. If the attacker gives a large value to Y^{FGV}, the correct key byte value may be deleted. If he gives a small value to Y^{FGV}, there will be a lot of guessing keys be to enumerated. We set the differential threshold Y^{FGV} of our group verification chain to $\frac{1}{3}$. This value is achieved through experience.

4 Experimental Results

Our experiments are performed on an Rotating S-boxes Masking (RSM) [8] protected AES-256 implemented on the Side-channel Attack Standard Evaluation Board (SASEBO). 10000 power traces are downloaded from the webset of DPA *contest v4* [1]. CCA is used to find the time samples of each S-box in the first round. To enhance the attack ability of CCA, Template Attack (TA) is combined with CCA. Then, we extract 4 interesting points from time interval of about a clock cycle suggested in [11].

We only compare our group verification chain with the fault tolerant chain proposed by Wang et al. [15]. Since fault tolerant chain is so far the only one practical scheme. $\xi_1, \cdots, \xi_7$ are used to verify guessing key byte values on $\xi_8, \cdots, \xi_{16}$. $\xi_{10}, \cdots, \xi_{16}$ are used to verify guessing key byte values on $\xi_1, \cdots, \xi_7$. Experimental results under different thresholds Thr_k, $Thr_{\varDelta}$ and different numbers of power traces are given in Sects. 4.1, 4.2 and 4.3.

4.1 Experimental Results Under Different Thresholds Thr_k

Firstly, we compare our group verification chain (FGV-MDCA and WGV-MDCA) with fault tolerant chain under different thresholds Thr_k. $Thr_{\varDelta}$ in fault tolerant chain is set to 1. This value is set to 5 in our FGV-MDCA and WGV-MDCA.

The success rate [14] of the 3 schemes are shown in Fig. 4. If $\varDelta_{(k_1,k_b)}$ of two key bytes k_1 and k_b is wrong and there exist one or several wrong k_b that satisfy

$k_b = k_1 \oplus \Delta_{(k_1,k_b)}$. Then, the scheme of Wang et al. will considers k_b as the correct key byte value. This is the main reason of failure of key recovery.

The success rate of fault tolerant chain decreases with the increase of Thr_k, which is very high when $Thr_k \leq 2$ (as shown in Fig. 4). Since the probability of wrong key byte k_b satisfying $k_b = k_1 \oplus \Delta_{(k_1,k_b)}$ is small. With the increase of Thr_k, this probability increases. When $Thr_k = 2$, fault tolerant chain can get a success rate of about 0.90. However, this value is only about 0.44 when $Thr_k = 5$. When $Thr_k > 13$, the success rate is smaller than 0.10. That is to say, The larger the Thr_k, the harder for the attacker to get success.

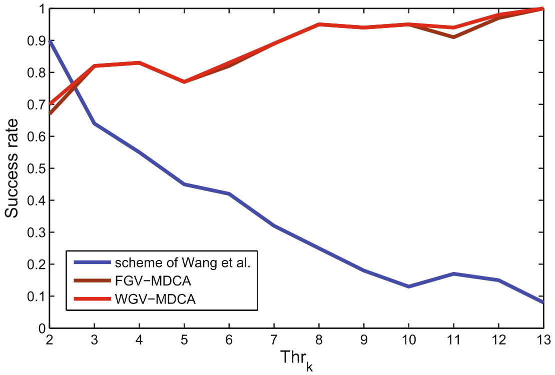

Fig. 4. Success rate under different Thr_k

The success rate of our FGV-MDCA and WGV-MDCA increases with Thr_k. When Thr_k is from 2 to 13, the success rate of FGV-MDCA and WGV-MDCA is from about 0.70 to about 1.00. When $Thr_k > 3$, the success rate of our FGV-MDCA and WGV-MDCA is greater than that of the scheme of Wang et al. This indicates that, the efficiency of group verification chain is slightly lower when Thr_k is small. With the increase of Thr_k, the correct key byte values fall within Thr_k and will be more effectively verified by group in our scheme.

Since Thr_Δ is set to 5, the success rate of FGV-MDCA and WGV-MDCA are very similar by only enlarging Thr_k.

4.2 Experimental Results Under Different Thresholds Thr_Δ

Secondly, we compare our FGV-MDCA and WGV-MDCA with the scheme of Wang et al. under different thresholds Thr_Δ. Thr_k here is set to 8. Thr_Δ of the scheme of Wang et al. is changed. Fault tolerant chain introduced in Sect. 2 can not be used in large Thr_Δ. We here enlarge this threshold. We then enumerate all possible chains that satisfy fault tolerant chain.

The success rate of the 3 schemes under different thresholds Thr_Δ are shown in Fig. 5. The success rate of fault tolerant chain is far lower than that of our

FGV-MDCA and WGV-MDCA. The success rate of fault tolerant chain does not significantly change with the increase of Thr_Δ. It ranges from 0.25 to 0.30 compared to from 0.8 to 1.00 of our FGV-MDCA and WGV-MDCA. The success rate of our FGV-MDCA and WGV-MDCA increase with Thr_Δ.

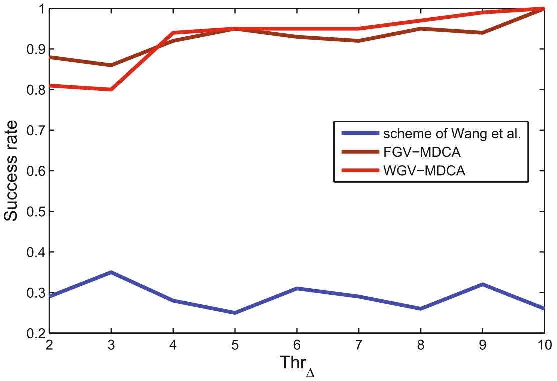

Fig. 5. Success rate under different thresholds Thr_Δ

When $Thr_\Delta = 2$, the success rate of FGV-MDCA and WGV-MDCA are about 0.87 and 0.82 respectively, both of which are significantly higher than that of the scheme of Wang et al. The success rate of FGV-MDCA is a little higher than that of WGV-MDCA when $Thr_\Delta < 4$. When $Thr_\Delta \geq 4$, the success rate of WGV-MDCA is higher than that of FGV-MDCA. The normalized weight of the correct key byte value in each reordered sequence $\{\xi_i | i = 1, 2, \cdots, 16\}$ in WGV-MDCA is more obvious than that in FGV-MDCA. This indicates that, the WGV-MDCA is more efficient than that of FGV-MDCA when Thr_Δ is large.

4.3 Experimental Results Under Different Numbers of Power Traces

Finally, we compare our FGV-MDCA and WGV-MDCA with the scheme of Wang et al. under the condition that different numbers of power traces are used. Thr_Δ is set to 5 and Thr_k is set to 8. Thr_Δ of the scheme of Wang et al. is still set to 1, since $Thr_\Delta > 1$ is very different from the fault tolerant chain.

When the number of power traces used in each repetition is from 60 to 170, the success rate of the 3 schemes are shown in Fig. 6. The success rate of the scheme of Wang et al. is far lower than that of our FGV-MDCA and WGV-MDCA. It ranges from 0 to 0.55 compared from 0.18 to 1.00 of our FGV-MDCA and from 0.37 to 1.00 of our WGV-MDCA. When the number of power traces used in each repetition is more than 150, the success rate of FGA-MDCA and WGA-MDCA is close to 1. However, the success rate of fault tolerant chain is only about 0.50 when about 170 power traces are used.

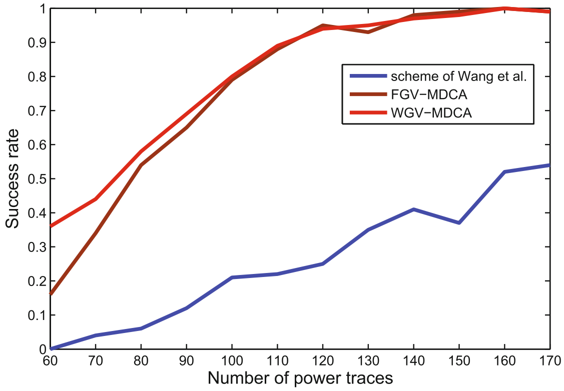

Fig. 6. Success rate under different numbers of power traces

The success rate of FGV-MDCA and WGV-MDCA is very close when more than 100 power traces are used in each repetition (as shown in Fig. 6). This is because, with the increase number of power traces, the locations of the correct key byte values and Δs fall in the top positions of $\{\xi_i | i = 1, 2, \cdots, 16\}$ and Δs sequences with higher probabilities.

5 Conclusions

In this paper, we propose group verification chain to enhance fault tolerant chain proposed by Wang et al. We combine MDCA and CCA to implement group verification chain and propose Group Verification based Multiple-Differential Collision Attack (GV-MDCA). Frequency based GV-MDCA (FGV-MDCA) and Weight based GV-MDCA (WGV-MDCA) are given. Experimental results performed on the power trace set of DPA *contest v*4 show that our group verification chain significantly improve the efficiency of fault tolerant chain.

Acknowledgment. This research is supported by the Nation Natural Science Foundation of China (No. 61372062).

References

1. Dpa contest. http://www.dpacontest.org/home/
2. Agrawal, D., Archambeault, B., Rao, J.R., Rohatgi, P.: The EM side—channel(s). In: Kaliski, B.S., Koç, K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 29–45. Springer, Heidelberg (2003). doi:10.1007/3-540-36400-5_4
3. Biryukov, A., Khovratovich, D.: Two new techniques of side-channel cryptanalysis. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 195–208. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74735-2_14

4. Bogdanov, A.: Multiple-differential side-channel collision attacks on AES. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 30–44. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85053-3_3
5. Bogdanov, A., Kizhvatov, I.: Beyond the limits of DPA: combined side-channel collision attacks. IEEE Trans. Comput. **61**(8), 1153–1164 (2012)
6. Handschuh, H., Preneel, B.: Blind differential cryptanalysis for enhanced power attacks. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. LNCS, vol. 4356, pp. 163–173. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74462-7_12
7. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999). doi:10.1007/3-540-48405-1_25
8. Moradi, A., Guilley, S., Heuser, A.: Detecting hidden leakages. In: Boureanu, I., Owesarski, P., Vaudenay, S. (eds.) ACNS 2014. LNCS, vol. 8479, pp. 324–342. Springer, Heidelberg (2014). doi:10.1007/978-3-319-07536-5_20
9. Moradi, A., Mischke, O., Eisenbarth, T.: Correlation-enhanced power analysis collision attack. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 125–139. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15031-9_9
10. Nassar, M., Souissi, Y., Guilley, S., Danger, J.: RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs. In: 2012 Design, Automation & Test in Europe Conference & Exhibition, DATE 2012, Dresden, Germany, 12–16 March 2012, pp. 1173–1178 (2012)
11. Rechberger, C., Oswald, E.: Practical template attacks. In: Lim, C.H., Yung, M. (eds.) WISA 2004. LNCS, vol. 3325, pp. 440–456. Springer, Heidelberg (2005). doi:10.1007/978-3-540-31815-6_35
12. Schramm, K., Leander, G., Felke, P., Paar, C.: A collision-attack on AES. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 163–175. Springer, Heidelberg (2004). doi:10.1007/978-3-540-28632-5_12
13. Schramm, K., Wollinger, T., Paar, C.: A new class of collision attacks and its application to DES. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 206–222. Springer, Heidelberg (2003). doi:10.1007/978-3-540-39887-5_16
14. Standaert, F.-X., Malkin, T.G., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 443–461. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01001-9_26
15. Wang, D., Wang, A., Zheng, X.: Fault-tolerant linear collision attack: a combination with correlation power analysis. In: Huang, X., Zhou, J. (eds.) ISPEC 2014. LNCS, vol. 8434, pp. 232–246. Springer, Heidelberg (2014). doi:10.1007/978-3-319-06320-1_18