SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2017, Vol. 60 042401:1–042401:14
doi: 10.1007/s11432-016-0071-7

Improving DFA attacks on AES with unknown and random faults

Nan LIAO[1], Xiaoxin CUI[1]*, Kai LIAO[1], Tian WANG[1], Dunshan YU[1] & Xiaole CUI[2]*

[1]*Institute of Microelectronics, Peking University, Beijing* 100871*, China;*
[2]*Key Lab of Integrated Microsystems, Peking University Shenzhen Graduate School, Shenzhen* 518055*, China*

Received April 5, 2016; accepted May 23, 2016; published online December 14, 2016

Abstract Differential fault analysis (DFA) aiming at the advanced encryption standard (AES) hardware implementations has become a widely research topic. Unlike theoretical model, in real attack scenarios, popular and practical fault injection methods like supply voltage variation will introduce faults with random locations, unknown values and multibyte. For analyzing this kind of faults, the previous fault model needed six pairs of correct and faulty ciphertexts to recover the secret round-key. In this paper, on the premise of accuracy, a more efficient DFA attack with unknown and random faults is proposed. We introduce the concept of theoretical candidate number in the fault analysis. Based on this concept, the correct round-key can be identified in advance, so the proposed attack method can always use the least pairs of correct and faulty ciphertexts to accomplish the DFA attacks. To further support our opinion, random fault attacks based on voltage violation were taken on an FPGA board. Experiment results showed that about 97.3% of the attacks can be completed within 3 pairs of correct and faulty ciphertexts. Moreover, on average only 2.17 pairs of correct and faulty ciphertexts were needed to find out the correct round-key, showing significant advantage of efficiency compared with previous fault models. On the other hand, less amount of computation in the analyses can be realized with a high probability with our model, which also effectively improves the time efficiency in DFA attacks with unknown and random faults.

Keywords AES, DFA attacks, unknown and random faults, efficient, theoretical candidate number, voltage violation

Citation Liao N, Cui X X, Liao K, et al. Improving DFA attacks on AES with unknown and random faults. Sci China Inf Sci, 2017, 60(4): 042401, doi: 10.1007/s11432-016-0071-7

1 Introduction

Data security has become one of the most important issues in finance, communication and military fields. The widespread use of smart cards, mobile handsets and intelligence appliances increases the risk of information leakage. Thus protection against accidental or intentional attacks on crypto-hardware has become a popular research topic. In modern crypto-circuit designs, strict cryptographic algorithms are applied to provide reliable computing environments. However, the rapid development of physical attacks makes mathematical algorithms not secure enough to protect sensitive information. During the physical attacks, side channel information like power consumption, electromagnetic leakage and timing

* Corresponding author (email: cuixx@pku.edu.cn, cuixl@pkusz.edu.cn)

information which is correlated to the secret key will be utilized to disclose the secret key [1]. Attacks that target straightly the hardware implementation of cryptographic applications have become the main threats in recent years.

Since more and more countermeasures based on software and hardware have been proposed to resist side channel attacks like power attack and electromagnetic attack, fault attack has become a powerful and popular attack method against cryptographic applications [2]. The fault attack was first introduced by Boneh et al. in 1997 [3]. By using the computational errors occurring during the executions of the cryptographic algorithm, they break an RSA implementation and recover the secret key with both a correct and a faulty signature of the same message. After that, fault attack was implemented successfully on symmetric block ciphers data encryption standard (DES) and the concept of differential fault analysis (DFA) was introduced in 1996 [4]. In 2000, DFA attacks on elliptic curve cryptosystems were proposed by Biehl et al. [5]. After October 2000, the advanced encryption standard (AES) substituted DES as the global standard for sensitive data encryption [6]. Since then, AES has quickly become the most popular cryptographic algorithm implemented on cryptographic hardwares especially on smart cards. To test the security of cryptographic smart cards using AES, DFA attacks on AES undoubtedly becomes a widely research field both in academic and industrial circles.

In recent years, several DFA attacks on AES have been reported in literature. The attack presented in [7] recovers the key with 250 faulty ciphertexts by inducing byte level faults in the input of the ninth round of AES encryption. In [8], 256 faulty encryptions are needed to determine a 128-bit secret key based on a very liberal fault model. By using a byte level fault induction between the 8th MixColumn and the 9th MixColumn, Dusart et al. break the full 128-bit secret key by analyzing 40 faulty ciphertexts in [9]. In [10], the number of faulty ciphertexts needed was further reduced. As long as the single fault is located between 8th MixColumn and the 9th MixColumn, only 8 faulty ciphertexts are enough to recover the whole 128-bit secret key. Multi-byte fault attack was presented in [11]. The authors proposed two models: 1500 faulty ciphertexts are required if all the 4 bytes of one column are influenced by the occurred fault; while the attacker only needs 6 faulty ciphertexts to discover the secret key if at most 3 bytes of one column are corrupted.

It can be concluded that if the fault model is based on single-byte fault and the fault location is in a specific round, the number of faulty ciphertexts needed is quite small. However, in real scenarios such attacks are feasible only with sophisticated equipment like laser beam. Utilizing the laser beam can accurately control the fault type and location, but this method is too complicated and high-cost [12]. In the real attack, less costly and non-invasive fault injection methods like supply voltage variation and clock glitch injection are more practical and popular [13,14]. However, these methods introduce faults with random locations, unknown values and multibyte (e.g., 1, 2 or 3 bytes). Therefore multi-byte fault models are required for the accuracy, meanwhile it also indicates more pairs of correct and faulty ciphertexts needed and more complicated mathematical operations to recover the correct round-key, which sacrifices efficiency to guarantee accuracy. On the other hand, if the first two faulty ciphertexts to be analyzed are both affected by single-byte fault, the efficiency can be effectively improved by applying the single-byte model, which makes distinguishing the fault type the most critical step in the fault attacks with unknown and random faults. To solve this problem, a novel fault model aiming at random fault attack is proposed in this paper, which takes both accuracy and efficiency into consideration. Based on the fault model, the four-byte round-key can be recovered correctly with only 2 pairs of correct and faulty ciphertexts.

2 Previous DFA attacks on AES with unknown and random faults

Most of the fault attacks aiming at AES implementation are based on the principle of setup time violation. Raising the clock frequency or lowering the supply voltage will make some logic paths fail to setup properly, leading to faulty results of the encryption. Here the specific fault type and location depend on the hardware structure. For AES implementation, the encryption processes contribute to the longest

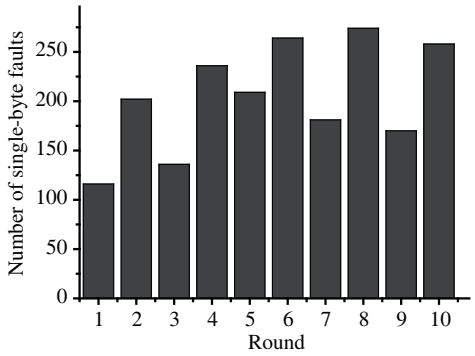

Figure 1 The distribution of single-byte faults in random fault attack.

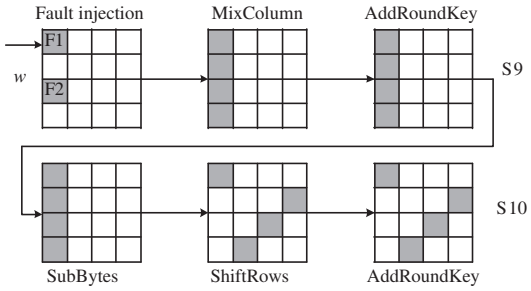

Figure 2 The fault propagation of 9th round models.

critical path delay, which contains a fixed sequence of operations: SubBytes, ShiftRows, MixColumns and AddRoundKey. Therefore setup failures will appear in the encryption processes first, influencing the 16-byte internal-state data. Finally the ciphertexts will be corrupted by the faults. Furthermore, since the critical path is highly data-dependent and each round consists of the same operations, faults will appear randomly in each round. For better demonstration, a low voltage fault attack was carried out on an FPGA-based AES implementation of which the encryption process is realized by iteration structure. 7622 total faults were collected in this attack. To verify the analysis above, the distribution of single-byte faults is depicted in Figure 1. The fault location is obtained by inversing the faulty ciphertexts with the known key and calculating the difference between the faulty and the correct intermediate values in each round. The round at which the single-byte difference appears is considered to be the round injected by single-byte fault. It can be seen that 2046 single-byte faults out of 7622 total faults spread randomly at all the ten rounds, which coincides with the analysis above. Moreover it can be predicted that a similar distribution also applies for multi-byte faults.

Based on the analysis above, there is a certain probability that the fault will be injected in the first 8 rounds in the random fault attacks, which would corrupt all the 16-byte ciphertexts. In this situation, the fault analysis is infeasible since the attacker cannot determine the fault type and location. Therefore in the previous fault attacks with unknown and random faults, faults in the last two rounds are used to recover the key since these faults only influence certain bytes of the ciphertext. For higher efficiency, models based on the fault injection between the 8th MixColumns and the 9th MixColumns are chosen in most of the random fault attacks. Dusart's single-byte model and Moradi's multi-byte model are two typical 9th round models. These two models express the same fault propagation, which is shown in Figure 2.

As shown in Figure 2, due to the missing of MixColumns step in the 10th round, the faults only influence 4 of the 16 bytes of the ciphertext. Furthermore, it is worth noting that the distribution of the 4 bytes is decided by the fault position. Define the intermediate value before the MixColumns in the 9th round as w. If the faults are injected in the first column of w, the faulty bytes in the ciphertext will be ($S10_{0,0}$, $S10_{1,3}$, $S10_{2,2}$, $S10_{3,1}$), regardless of the number of faults. Thus for the fault position in columns 2, 3 and 4, the corresponding fault distributions will be ($S10_{0,1}$, $S10_{1,0}$, $S10_{2,3}$, $S10_{3,2}$), ($S10_{0,2}$, $S10_{1,1}$, $S10_{2,0}$, $S10_{3,3}$) and ($S10_{0,3}$, $S10_{1,2}$, $S10_{2,1}$, $S10_{3,0}$) respectively. Based on this characteristic, the attacker can easily select out the exploitable faulty ciphertexts to implement differential fault attacks.

As for the differential analysis, since the operations of ShiftRows, MixColumns and SubBytes are fixed and only the last round-key affects the output ciphertext, the attacker can make a hypothesis on the target 4-byte round-key ($K10_{0,0}$, $K10_{1,3}$, $K10_{2,2}$, $K10_{3,1}$). When a pair of correct and faulty ciphertexts is prepared, the attacker inverts the ciphertexts to the states w and $\widetilde{w}$ ($\widetilde{w}$ stands for the faulty one). The difference $\delta = w \oplus \widetilde{w}$ can be easily obtained as shown in (1). Here the effects of AddRoundKey in the 9th round can be eliminated since the operations from w to $S10$ are all linear except the SubBytes in the 10th round. Based on the model hypothesis, the attacker can check if the obtained difference is

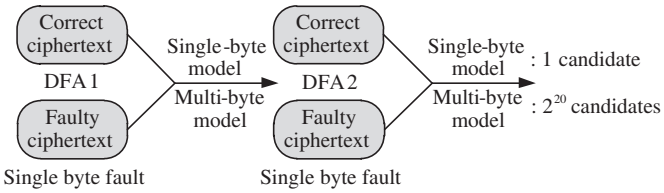

Figure 3 DFA on two ciphertexts infected by single byte fault.

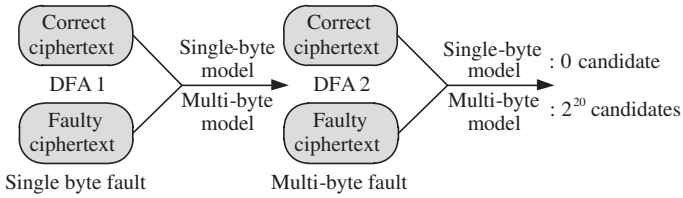

Figure 4 DFA on two ciphertexts infected by multi-byte fault.

composed of single byte or multi bytes. If the difference matches the hypothesis, the 4-byte round-key hypothesis is selected as a candidate. With a certain number of faulty ciphertexts, the corresponding 4-byte round-key can be uniquely determined with a reasonably high probability,

$$\begin{aligned}\delta &= w \oplus \widetilde{w}\\ &= \text{InvMixColumn}(K9 \oplus S9) \oplus \text{InvMixColumn}(K9 \oplus \widetilde{S}9)\\ &= \text{InvMixColumn}(S9 \oplus \widetilde{S}9),\\ S9 &= \text{InvSubBytes}(\text{InvShiftRows}(K10 \oplus S10),\\ \widetilde{S}9 &= \text{InvSubBytes}(\text{InvShiftRows}(K10 \oplus \widetilde{S}10).\end{aligned} \tag{1}$$

It is worth noting that although the two fault models express the same fault propagation, the number of faulty ciphertexts required to recover the secret keys is different. Single-byte fault model only needs 2 pairs of correct and faulty ciphertexts to recover the 4-byte round-key, while it requires 6 with multi-byte fault model. In extreme condition, 1500 pairs of correct and faulty ciphertexts are required if 4-byte faults injection is considered. But in real attack, this situation scarcely exists since it's almost impossible that 4 bytes flip simultaneously in one column when the attack is implemented in appropriate experimental conditions. Thus 4-byte fault injection will not be considered in our research.

As mentioned above, most of the random faults are induced by regulating the supply voltage or clock frequency. If the voltage or frequency is adjusted in an appropriate range, single-byte faults will appear with a high probability, which makes the fault analysis quite efficient with single-byte fault model. Imagine the situation that two consecutive exploitable faulty ciphertexts are both caused by single-byte fault as shown in Figure 3, which is sufficiently likely to happen in random fault attacks. By utilizing single-byte model, the 4-byte round-key can be uniquely determined with these two faulty ciphertexts. However, if multi-byte fault model is applied, there are still about 220 key candidates left after analyzing two pairs of correct and faulty ciphertexts. For recovering the round-key, another 4 faulty ciphertexts are required.

Although single-byte model expresses high efficiency, accuracy of the analysis should be put in higher priority. In the practical attack, we find that two-byte and three-byte faults still emerge with a certain probability. Take the situation in Figure 4 for instance. Faulty ciphertext 1 is caused by single-byte fault, while faulty ciphertext 2 results from multi-byte fault injection. Still using single-byte fault model, no key candidates will remain after the analysis, which means that the correct round-key cannot be picked out by this method. Thus, Moradi's multi-byte model is a better choice to guarantee the accuracy of the analysis in this situation. However, raising the accuracy is at the cost of sacrificing efficiency. To recover the 128-bit secret key, multi-byte fault model requires 24 faulty ciphertexts, while it demands only 8 faulty ciphertexts with single-byte fault model. In the real attack scenarios, it always needs to recover numerous keys. Multi-byte fault model, to some extent, will restrict the attack efficiency compared with

single-byte fault model. It is quite necessary to find a new fault model which considers both efficiency and accuracy for the fault attack research with unknown and random faults.

3 Proposed fault model

Before we present the proposed model, the reason why multi-byte fault model requires 6 pairs of correct and faulty ciphertexts while single byte model only needs 2 will be investigated first. As discussed above, the multi-byte fault model covers one-byte, two-byte and three-byte faults situations. The proportion of the number of covered faults to the number of all possible faults is defined as the fault coverage rate of the model, CR. Therefore the CR of the multi-byte fault model is

$$\mathrm{CR} = \frac{C_4^1 \times 255 + C_4^2 \times 255^2 + C_4^3 \times 255^3}{256^4 - 1} = 0.0155. \tag{2}$$

Averagely, every covered fault corresponds to one candidate 4-byte round-key. Thus it can be deduced that the proportion of the number of candidate keys to the number of all possible keys, $P_{\mathrm{candidates}}$, will be approximate to CR,

$$P_{\mathrm{candidates}} \approx \mathrm{CR} = 0.0155. \tag{3}$$

For analyzing the first pair of correct and faulty ciphertexts, the number of all possible round-keys is 256^4. Therefore the number of candidate keys will be about

$$N_{\mathrm{can\_key}} = N_{\mathrm{all}} \times P_{\mathrm{key}} = 256^4 \times 0.0155 = 66571993. \tag{4}$$

In the real attack, large amount of experiment results show that the $N_{\mathrm{can\_key}}$ is very close to 66571993.

When analyzing the second pair, the candidate keys will be selected out from the candidates of the first analysis. With another plaintext and the same secret key, P_{can} will be the same as the first one. Consequently the number of candidate keys will decrease to about 1031865. As shown in Figure 5, the theoretical candidate number will be reduced to about 0.06 after analyzing 6 pairs of correct and faulty ciphertexts, which is close to 0. It indicates that no candidate keys can remain except the correct one, which explains the reason why multi-byte fault model requires 6 pairs of correct and faulty ciphertexts to obtain the round-key.

As for the single-byte fault model, based on the single-byte fault injection hypothesis, the proportion of the number of covered faults to the number of all possible faults is

$$\mathrm{CR}_{1\mathrm{b}} = \frac{C_4^1 \times 255}{256^4 - 1} = 2.37 \times 10^{-7}. \tag{5}$$

Accordingly, the proportion of the number of candidate keys will be approximate to 2.37×10^{-7}:

$$P_{\mathrm{can\_1b}} \approx \mathrm{CR}_{1\mathrm{b}} = 2.37 \times 10^{-7}. \tag{6}$$

Thus the theoretical candidate number after each analysis will vary as Figure 6 shows. Thanks to the small $P_{\mathrm{can\_1b}}$, it only needs two faulty ciphertexts for the candidate number to reach 0.00024, a figure close to zero, which means that the correct round-key will be the only candidate that remains after analyzing two pairs of correct and faulty ciphertexts.

Likewise, the situations of two-byte and three-byte fault models can be deduced based on the same principle. For these two models, the corresponding P_{can} are 9.08×10^{-5} and 0.0154 respectively. Figures 7 and 8 exhibit the corresponding variation of the theoretical candidate number after each analysis. It can be clearly seen that two-byte fault model requires 3 faulty ciphertexts to recover the round-key; and this figure increases to 6 when the implementation is merely injected by three-byte faults. Therefore the attacker can choose the corresponding fault model to improve the attack efficiency if the fault type is known. However, as mentioned above, the caused changes of MixColumns input in the 9th round cannot be distinguished in the practical random fault attacks. Multi-byte fault model is necessary to guarantee

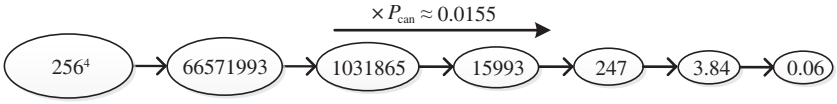

Figure 5 The theoretical candidate number after each analysis in multi-byte fault model.

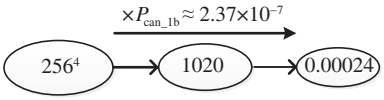

Figure 6 The theoretical candidate number after each analysis in single-byte fault model.

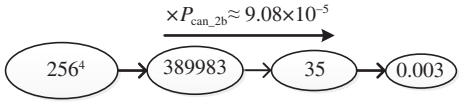

Figure 7 The theoretical candidate number after each analysis in two-byte fault model.

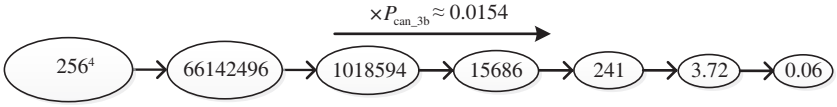

Figure 8 The theoretical candidate number after each analysis in three-byte fault model.

Figure 9 Two series of faulty ciphertexts.

the accuracy in previous analysis. So it always needs 6 faulty ciphertexts to find out the round-key no matter what kind of faults is injected to the implementation. In a way, traditional method limits the attack efficiency. To solve this problem, a new idea is proposed in this paper to make the best use of the different kinds of fault models.

Based on the analysis above, it can be learned that the number of faulty ciphertexts needed to recover the round-key depends on the theoretical candidate number. When this figure decreases close to 0, only the correct round-key can remain in the corresponding candidate set. By analyzing the reduction of the theoretical candidate number, we find that the P_{can} is a determining factor in this process. Multi-byte fault model requires 6 faulty ciphertexts due to its high P_{can}, which leads to a slow decreased trend of the theoretical candidate number. On the contrary, owing to a small $P_{\text{can\_1b}}$, single-byte fault model only needs 2 faulty ciphertexts to complete the attack. If small P_{can} can be exploited in the analysis with random faults, the number of faulty ciphertexts required can be reduced and accordingly the attack efficiency will be effectively improved, which is the key point of the proposed fault model.

Here two series of faulty ciphertexts are given in Figure 9. The figures in the frames stand for the fault types (1-byte, 2-byte or 3-byte faults). In the previous random fault attacks, 6 pairs of correct and faulty ciphertexts are needed since the fault type is unknown. For the comparison, suppose that the fault types are given to the attacker, thus the attacker can choose the corresponding fault model. Take series 1 for example, the first faulty ciphertext is influenced by a 2-byte fault and the corresponding $P_{\text{can\_2b}}$ is 9.08×10^{-5}. So, the theoretical candidate number after the first analysis will be 389983 and it decreases rapidly to 0.092 after the second analysis with a small $P_{\text{can\_1b}}$ of 2.37×10^{-7}, which is shown in Figure 10. As analyzed above, if the theoretical candidate number is close to 0, only the correct round-key can stay after the corresponding analysis. Therefore, on the basis of the hypothesis that the fault types are known to the attacker, only 2 pairs of correct and faulty ciphertexts are sufficient to recover the round-key for series 1. The same principle also applies with series 2 in Figure 11. The theoretical candidate number drops to 0.014 after analyzing three pairs of correct and faulty ciphertexts even though

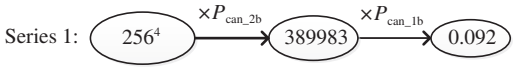

Figure 10 The theoretical candidate number after each analysis for series 1.

Figure 11 The theoretical candidate number after each analysis for series 2.

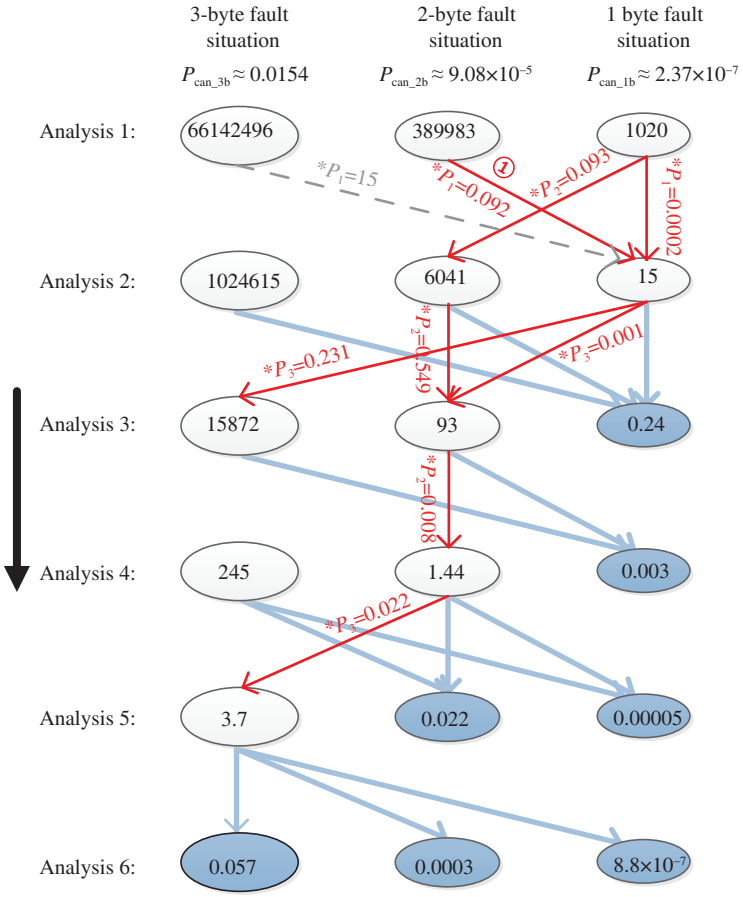

Figure 12 (Color online) The situations that can complete the attack within 6 pairs of correct and faulty ciphertexts.

the second ciphertext is injected by a 3-byte fault. Thus it can be concluded that knowing the fault type can help saving three or four faulty ciphertexts to analyze compared with multi-byte fault model.

For DFA attacks with unknown and random faults, it is worth summarizing all the possible situations like ciphertext series 1 and 2 that can complete the fault analysis with less than 6 pairs of correct and faulty ciphertexts. These situations may provide ideas to build up a new fault model. For clarity, all the possible situations are illustrated in Figure 12. It further refines the changing process of theoretical candidate key number. The theoretical key candidates are subdivided into three parts according to different fault types. Figures in oval frames in the left column correspond to the theoretical number of key candidates which satisfy 3-byte fault situation, while the middle and the right ones are mapped to the 2-byte fault and the 1-byte fault situations respectively. For example, the figure '15' in the second row means that theoretically there are 15 key candidates satisfying 1-byte fault hypothesis. As for the line connecting two oval frames, it represents one possible situation of faulty ciphertexts series and we define it as a path. Take the dotted line for instance, it depicts one situation that the first ciphertext is infected by a 3-byte fault and the second one results from 1-byte fault injection.

Analyzing one pair of correct and faulty ciphertexts will filter out some wrong candidates and decrease the size of possible key candidate set. Moreover the selected candidates will be regarded as the possible keys for the next analysis. By subdividing the theoretical candidate number into three parts according to

different fault types, we can find the detailed direction of the candidates and the corresponding candidate contribution between two analyses in Figure 12. Here the lines with arrow show the direction of the candidates and the number upon the lines presents the corresponding contribution (The contribution is calculated by multiplying the number of possible keys by the corresponding P_{can}). It can be found that there exist nine possible paths between two analyses. Every theoretical candidate number will contribute part of itself to the next three situations. Meanwhile, each theoretical candidate number derives from three parts of the possible keys, where each part corresponds to one different fault type situation. By studying all the possible paths, the overall analysis can be summed up in three situations based on Figure 12.

(1) The dotted line path with contribution much higher than 1. For the dotted line path, the corresponding contribution is much higher than 0, which is not sufficient to select out the correct round-key. Take the dotted line in Figure 12 for example, if the real ciphertext series just matches this path, there will be about 15 key candidates after the first two analyses, therefore the following calculation is necessary to further decrease the size of possible key candidate set. In a way, the dotted line paths are not the situations needed to be concerned in our model. For the briefness, the other dotted paths are not shown in Figure 12.

(2) The black solid line path with contribution close to 0. Different from the dotted line, the corresponding contribution of black solid line path is close to 0, which means that no candidates but the correct one can remain after the analysis. Take the path 1 in Figure 12 for instance, the contribution is only 0.092. In the real situation, if the faulty ciphertexts series correspond to this path, according to the discussion above, the correct round-key will be the only candidate after analyzing two pairs of correct and faulty ciphertexts and the DFA attacks can be completed in advance. Thus we call the black solid line paths as exploitable paths.

(3) Grey oval frames with theoretical candidate number close to 0. Some grey oval frames exist in the last four analyses. Unlike the white frames, the theoretical candidate number in the grey frames is already close to 0, which indicates that all the paths pointing to the grey frames will be the exploitable paths with theoretical candidate number near to 0. For example, if the third faulty ciphertexts to be analyzed is infected by 1-byte fault, no matter what kind of fault influences the last faulty ciphertext, the correct round-key will be the only candidate of 1-byte fault situation after analyzing three pairs of correct and faulty ciphertexts. It is worth noting that grey frames occupy all the situations after 6 differential analyses, which just coincides with the fact that the previous multi-byte fault model needs 6 pairs of correct and faulty ciphertexts to find out the correct round-key.

By now, all the possible situations in Figure 12 have been summarized. Based on the conclusion, we can easily find the exploitable paths in the overall analysis, which make it feasible to recover the correct round-key in advance and realize a more efficient attack method.

In the attack analysis, it is too complicated and inefficient to verify all the possible situations like series 1 and 2 one by one. Based on the analysis of Figure 12, we proposed a more efficient fault model. Figure 13 depicts the proposed attack method. After the first analysis, the selected key candidates are categorized into three parts according to the fault types in the DFA, that are 1-byte, 2-byte and 3-byte faults respectively. As shown in Figure 12, situations that can select out the correct round-key exist in the last five analyses, which correspond to the black solid line paths and grey oval frames. Thus for analyzing the second pair of correct and faulty ciphertexts, DFA will be implemented on these exploitable situations first and the result will be verified. If there is one candidate still remaining after the DFA, then this candidate is just right the only correct round-key and the analysis finishes in advance. If no candidates remain after DFA in these exploitable situations, DFA will be carried out on other situations in the current analysis to further decrease the number of key candidates for the next analysis. By repeating this proposed method for the rest analyses, the correct round-key will be filtered out with less than 6 faulty ciphertexts with a reasonable high probability. An advantage of this fault model is that it covers all the possible situations of each analysis, which guarantees that the correct round-key can be recovered with the least amount of calculation. On the other hand, since the grey oval frame is the terminal point of the paths pointing to it, it is not necessary to consider the paths starting from the grey oval frame,

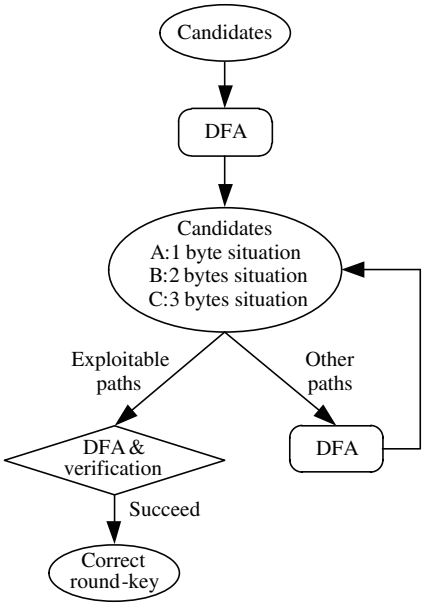

Figure 13 The proposed attack method.

Table 1 The corresponding fault type series of different required number of faulty ciphertexts

Required number of faulty ciphertexts	Fault type series
2	11,12,21
3	311, 312, 313, 321, 331, 221, 222, 231

which reduces the computation amount of the analysis.

In this fault model, fault analysis can be done in any one of the analyses towards the last five faulty ciphertexts and the number of faulty ciphertexts required in one specific attack depends on the fault type series. Table 1 presents the corresponding fault type series of different required number of faulty ciphertexts. Here only attacks that succeed with two or three faulty ciphertexts are considered since recovering the key with less faulty ciphertexts is the main purpose of the proposed model. As discussed above, if the attack is implemented in appropriate experimental conditions, the probability of single-byte fault infection will be much higher than the ones of two-byte fault and three-byte fault situations. Suppose the probability of the three situations are $P_1 = 0.89$ (single-byte), $P_2 = 0.10$ (two-byte), $P_3 = 0.01$ (three-byte). Then based on the data in Table 1, we can calculate the probability of attacks that succeed within 2 and 3 faulty ciphertexts.

(1) The probability of attacks that succeed with 2 faulty ciphertexts is

$$P_{2\text{ciphertexts}} = P_1 \cdot P_1 + P_1 \cdot P_2 + P_2 \cdot P_1 = 0.9701; \tag{7}$$

(2) The probability of attacks that succeed with 3 faulty ciphertexts is

$$\begin{aligned} P_{3\text{ciphertexts}} = {} & P_3 \cdot P_1 \cdot P_1 + P_3 \cdot P_1 \cdot P_2 + P_3 \cdot P_1 \cdot P_3 + P_3 \cdot P_2 \cdot P_1 \\ & + P_3 \cdot P_3 \cdot P_1 + P_2 \cdot P_2 \cdot P_1 + P_2 \cdot P_2 \cdot P_2 + P_2 \cdot P_3 \cdot P_1 \\ = {} & 0.0214; \end{aligned} \tag{8}$$

(3) Thus the probability of attacks that succeed with less than 3 faulty ciphertexts is

$$P_{2\text{ciphertexts}+3\text{ciphertexts}} = P_{2\text{ciphertexts}} + P_{3\text{ciphertexts}} = 0.9915. \tag{9}$$

Obviously, with this probability, fault analysis can be done with 2 faulty ciphertexts with quite a high probability of 97.01%. Considering the attacks that succeed with less than 3 faulty ciphertexts, the probability reaches up to 99.15%, which indicates that almost all the attacks can be completed successfully within 3 faulty ciphertexts. Thus it can be predicted that the proposed model will be a high-efficient and accurate fault model in the fault attacks with unknown and random faults.

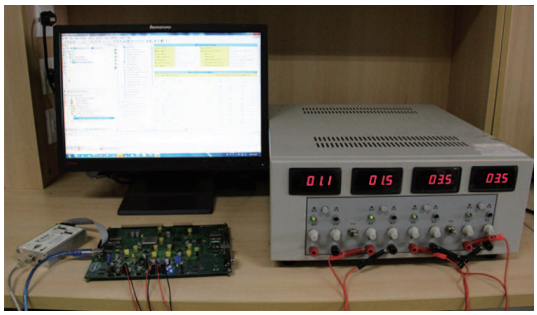

Figure 14 (Color online) The experiment platform.

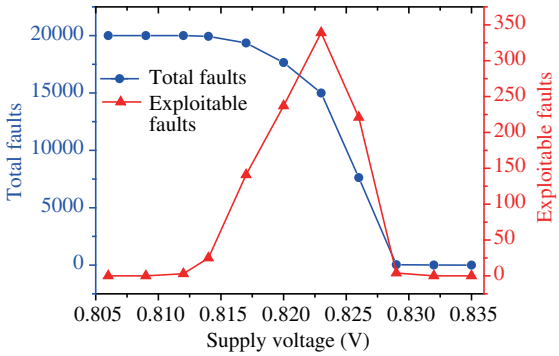

Figure 15 (Color online) The number of total faults and exploitable faults.

4 Experiment results

For better demonstration, attack experiment was conducted on an FPGA board. This attack targets a cryptographic FPGA Virtex-XC5VLX110. The standard 128-bit AES algorithm is implemented in the cryptographic FPGA. To realize random fault injection, low voltage attack based on setup time violation was carried out on the attack target. In the experiment, the cryptographic FPGA was fed by a power supply with appropriate precision and the voltage measurements were taken by a multimeter with a precision of 1 mV. Figure 14 presents this experiment platform.

In order to investigate the characteristics of faults, several intermediate voltages were chosen within the exploitable voltage range to implement the attack. For effectivity and consistency of the comparison, 20000 different plaintexts and one secret key (3141592653589793238462643383279 5) are prepared. For each voltage level, these plaintexts were encrypted by the same secret-key and the 20000 ciphertexts were recorded. By comparing the ciphertext with the corresponding correct one, it can be known whether the result is faulty, and the exploitable faults can be selected out based on the fault model. The number of the total faults and the exploitable ones at each voltage are depicted in Figure 15. As shown in Figure 15, the FPGA moves from an error-free state to a fully faulty behavior within about 25 mV. However, the number of exploitable faulty ciphertexts which have a diagonal-fault distribution does not show the same trend, which varies like a bell shape and reaches the maximum at about 823 mV. Thus 823 mV was chosen as the attack voltage since more exploitable faults can be gathered and the single-byte faults will be the main fault type.

To verify the efficiency and accuracy of our proposed fault model, we gathered 12000 faulty ciphertexts which have the same fault location ($S10_{0,0}$, $S10_{1,3}$, $S10_{2,2}$, $S10_{3,1}$) by repeating the random fault attack. Then the 12000 faulty ciphertexts were divided into 2000 groups randomly and each group has 6 ciphertexts. With previous multi-byte fault model, all the 6 ciphertexts of one group should be utilized to recover the 4-byte round-key. To validate our analysis above, attack methods based on the proposed fault model are applied to the 2000 groups' ciphertexts. Attack results are shown in Figure 16. 1710 out of 2000 groups recovered the round-key by analyzing 2 pairs of correct and faulty ciphertexts, which achieves quite a high proportion of 85.5%. As for the attacks requiring 3 ciphertexts to obtain the round-key, it makes up about 11.8% of the 2000 groups. Considering attacks that succeed within 3 faulty ciphertexts, 97.3% (1946/2000) of the attacks can achieve this goal, which coincide with our prediction above. Note that only a small part of attacks needed more than 4 faulty ciphertexts to recover the secret key, which is owing to the small probability of multi-byte faults in the appropriate attack conditions. Only 51 groups completed the attack with 4 faulty ciphertexts and 3 groups used 5 faulty ciphertexts to find out the secret key. It is worth noting that none groups needed 6 faulty ciphertexts to accomplish the analysis, while 6 is the average number of faulty ciphertexts to recover the round-key in the previous fault attacks with unknown and random faults.

For every key candidate in the attack, the computation based on (1) will be carried out once. Each

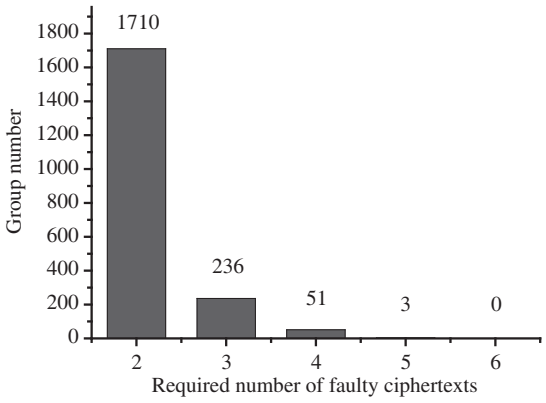

Figure 16 The attack result of the proposed method.

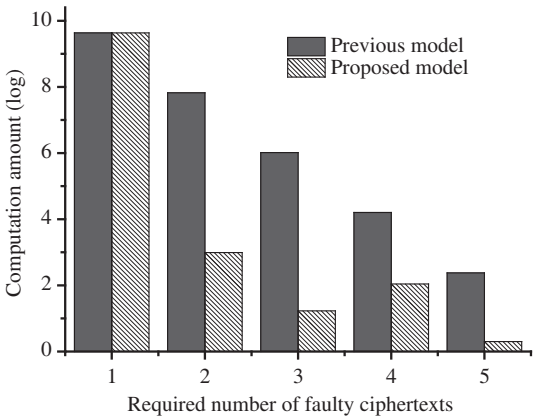

Figure 17 The computation amount of proposed model.

Table 2 The number of attacks that completed with the minimum amount of computation

Required number of faulty ciphertexts	2	3
Groups with minimum computation amount	1329	66
Total groups	1710	236

computation takes a certain period of time. Therefore the amount of computation influences the time efficiency directly. In our model, since the exploitable paths will be executed first, the computation can be completed in advance if the exploitable path corresponds to the actual situation. In this situation the amount of computation can be reduced effectively, which leads to an improvement of time efficiency. Moreover, in the practical attacks, exploitable paths that originate from the 1-byte situation will be processed in priority. If the analysis succeeds in finding out the correct round-key in these paths, the minimum amount of computation can be realized.

For better verification of the attack efficiency, a further comparison of the amount of computation in each analysis is presented in Figure 17. Based on the attack results of the 2000 groups' ciphertexts, the average minimum computation amount needed in each analysis is given, and the corresponding average amount of previous model is shown for comparison. As depicted in Figure 17, our model shows a dramatic reduction of computation amount compared with the previous model. These two models share the same amount of computation in the first analysis of attack since all the possible values of a word are regarded as key candidates at the beginning. Our model shows its advantage from the second analysis on. For instance, with the previous model, the standard amount of computation in the second analysis is about 6.65×10^7. But the minimum computation amount of our model is only about 1000, which achieves a remarkable reduction. For the third analysis of attack, significant reduction of computation amount is also realized; the numerical value has dropped from about 10^7 to only about 15. The same trend also exists in other analyses, but we predominantly focus on the first three analyses since the attack results show that most of our attacks can obtain the round-key within 3 faulty ciphertexts. On the other hand, because the attack voltage is adjusted at an appropriate point at which the single-byte faults will be the main fault type, the minimum amount of computation can be realized with a high probability. Table 2 shows the number of attacks which are completed with the minimum amount of computation. For the attacks which require two rounds of analysis, 1329 of the 2000 groups recover the key with the minimum computation amount, accounting for as high as 66.45 percent. Among 236 groups of attacks which demand three rounds of analysis, 66 groups can use the minimum amount of computation to find out the key, which also achieves a considerable percentage of 27.97%.

Besides efficiency, accuracy is also an important factor of the fault model. To evaluate the accuracy of our model, four analysis processes selected from the attack experiments are presented in Figure 18. The corresponding faulty ciphertexts and the correct ones in each analysis are given in Table 3. The figures in the oval frames are the actual number of key candidates in the real attacks. Besides, the

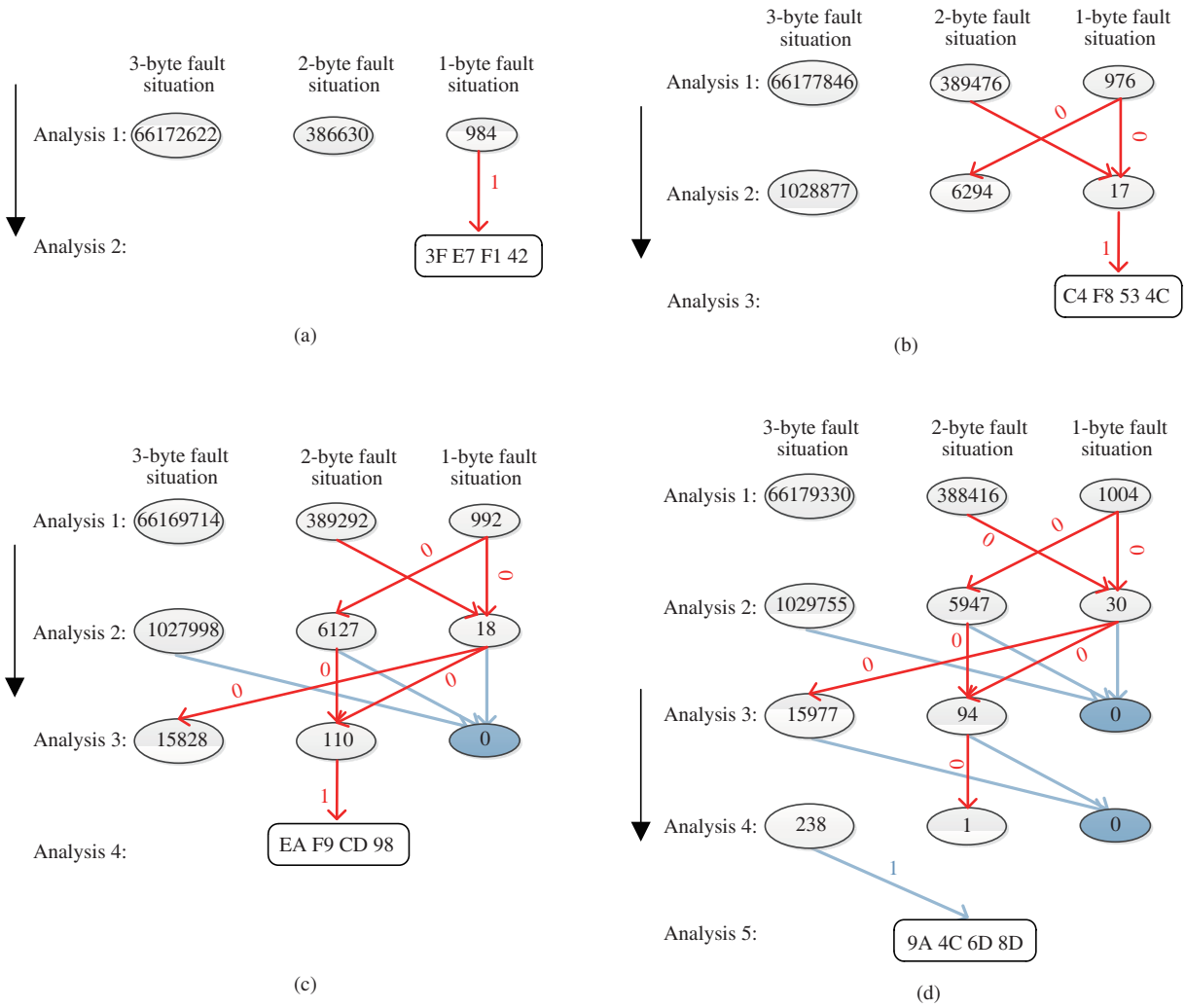

Figure 18 (Color online) Four attack processes based on the proposed fault model. The attack for recovering (a) $(S10_{0,0}, S10_{1,3}, S10_{2,2}, S10_{3,1},)$; (b) $(S10_{0,1}, S10_{1,0}, S10_{2,3}, S10_{3,2},)$; (c) $(S10_{0,3}, S10_{1,2}, S10_{2,1}, S10_{3,0},)$; (d) $(S10_{0,2}, S10_{1,1}, S10_{2,0}, S10_{3,3},)$.

four recovered results just constitute the whole round-key. For better demonstration of our model, the presented analyses cover attacks that succeed with different number of faulty ciphertexts. It can be seen that due to two continuous single-byte fault infections in the first two faulty ciphertexts, the first attack recovered the four-byte round-key with only two faulty ciphertexts, which is the most common scenario in our attack. After analyzing three pairs of correct and faulty ciphertexts, the round-key was obtained in the second attack. More faulty ciphertexts were needed in the other two attacks, i.e. four for the third attack and five for the fourth attack. It is attributed to the fact that the faulty ciphertexts in the early analyses are influenced by multi-byte faults. According to the recovered results, the whole round-key is composed as '3ff86d98c44ccd429af9f14ceae7538d'. By applying the key schedule of AES on the round-key, the original key is recovered as '3141592653589793238462643383279 5', which is just the correct key of the crypto-system. In the real attack, the attacker can use the recovered key to encrypt the same plaintexts of the target, and verify the correctness of the attack by comparing the ciphertexts with the encrypted results of the crypto-system.

To sum up, analysis and comparison above demonstrate that on the premise of accuracy, the proposed fault model shows significant advantage of efficiency compared with the previous fault model towards unknown and random faults. Almost all the attacks can be completed within 3 faulty ciphertexts and on average only 2.17 faulty ciphertexts are needed to find the four-byte round-key in the experiment.

Table 3 The corresponding faulty ciphertexts and the correct ones in each analysis

		Ciphertexts 1	Ciphertexts 2	Ciphertexts 3	Ciphertexts 4	Ciphertexts 5
$S10_{0,0}, S10_{1,3}, S10_{2,2}, S10_{3,1}$	Correct	DF CE D4 BE	26 08 CE 03	–	–	–
	Faulty	AE EE 45 93	D3 EC FF 25	–	–	–
$S10_{0,1}, S10_{1,0}, S10_{2,3}, S10_{3,2}$	Correct	4D 8E 50 3F	3D FB D1 EF	E1 1D E3 EC	–	–
	Faulty	8E 20 9E 56	E0 8D 5C 6C	60 C3 5A 43	–	–
$S10_{0,3}, S10_{1,2}, S10_{2,1}, S10_{3,0}$	Correct	78 32 CD 9D	9F 10 5D E9	96 20 D0 E9	61 56 3F 6A	–
	Faulty	3E FC 9E B3	FD C1 F3 FD	67 32 9E D8	45 91 92 3A	–
$S10_{0,2}, S10_{1,1}, S10_{2,0}, S10_{3,3}$	Correct	6C F0 00 D7	66 55 E9 D6	EF A0 F6 E2	80 C2 48 39	36 2C BE BE
	Faulty	21 7C 4D E8	22 B6 8B 27	55 B0 87 C2	38 90 A8 20	FC 70 23 13

Moreover, less amount of computation in each analysis brings about significant improvement of time efficiency in our attack.

5 Conclusion

In this paper, an efficient and accurate DFA model aiming at unknown and random faults is proposed. The concept of theoretical candidate number is introduced, which provides the possibility to select out the correct round-key with less than 6 faulty ciphertexts. Based on this principle, we present a new method to run the attack on AES. Analysis shows that this method can always recover the round-key with the least faulty ciphertexts no matter what the fault types are. If the attack environment is adjusted to an appropriate range, calculation results show that most of the attacks can be completed after analyzing 2 pairs of correct and faulty ciphertexts and about 99% can be done within 3 pairs. For better demonstration, attack experiment was established on an FPGA board. We realized the random fault attacks by lowering the supply voltage and adjusted the voltage to an appropriate point. Very successful results of the proposed attack methods are presented. 1710 out of 2000 groups recovered the secret round-key with 2 faulty ciphertexts and 97.3% of the all groups complete the analysis with less than 3 faulty ciphertexts. On the premise of accuracy, on average only 2.17 faulty ciphertexts are needed to find the round-key in the experiment. Moreover, less amount of computation in each round achieves significant improvement of time efficiency in our attack. It can be concluded that our model exhibits significant efficiency in the fault attacks with unknown and random faults.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 61306040), National Basic Research Program of China (973) (Grant No. 2015CB057201), Natural Science Foundation of Beijing (Grant No. 4152020), Natural Science Foundation of Guangdong Province (Grant No. 2015A030313147), and R&D Project of Guangdong Government (Grant No. 2014B090913001).

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

References

1 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-Verlag, 2006. 192–207
2 Tiri K, Verbauwhede I. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In: Proceedings of the Conference on Design, Automation and Test in Europe, Washington, 2004. 246–251
3 Boneh D, DeMillo R A, Lipton R J. On the importance of checking cryptographic protocols for faults. In: Advances in Cryptology-EUROCRYPT'97. Berlin: Springer-Verlag, 1997. 37–51
4 Biham E, Shamir A. Differential fault analysis of secret key cryptosystems. In: Advances in Cryptology-CRYPTO'97. Berlin: Springer-Verlag, 1997. 513–525
5 Biehl I, Meyer B, Müller V. Differential fault attacks on elliptic curve cryptosystems. In: Advances in Cryptology-CRYPTO 2000. Berlin: Springer-Verlag, 2000. 131–146
6 Daemen J, Rijmen V. The Design of Rijndael: AES - The Advanced Encryption Standard. New York: Springer Science & Business Media, 2013
7 Giraud C. DFA on AES. In: Proceedings of the 4th International Conference on Advanced Encryption Standard. Berlin: Springer-Verlag, 2004. 27–41

8 Blömer J, Seifert J P. Fault based cryptanalysis of the advanced encryption standard (AES). In: Financial Cryptography. Berlin: Springer-Verlag, 2003. 162–181
9 Dusart P, Letourneux G, Vivolo O. Differential fault analysis on A.E.S. In: Applied Cryptography and Network Security. Berlin: Springer-Verlag, 2003. 293–306
10 Piret G, Quisquater J J. A differential fault attack technique against SPN structures, with application to the AES and Khazad. In: Cryptographic Hardware and Embedded Systems-CHES 2003. Berlin: Springer-Verlag, 2003. 77–88
11 Moradi A, Shalmani M T M, Salmasizadeh M. A generalized method of differential fault attack against AES cryptosystem. In: Cryptographic Hardware and Embedded Systems-CHES 2006. Berlin: Springer-Verlag, 2006. 91–100
12 Agoyan M, Dutertre J M, Mirbaha A P, et al. Single-bit DFA using multiple-byte laser fault injection. In: Proceedings of IEEE International Conference on Technologies for Homeland Security, Waltham, 2010. 113–119
13 Selmane N, Guilley S, Danger J L. Practical setup time violation attacks on AES. In: Proceedings of the 7th European Dependable Computing Conference, Kaunas, 2008. 91–96
14 Barenghi A, Bertoni G, Breveglieri L, et al. Low voltage fault attacks to AES and RSA on general purpose processors. International Association for Cryptologic Research (IACR) ePrint Archive, 2010. 130