A method for fault recognition in the last three rounds of Advanced Encryption Standard

Huilong Jiang,1,2* Xiang Zhu,1,2,M† Jinfeng Pang,3 Zhipeng Liu,3 Jianwei Han,1,2† and Yue Li†
1Space Environment Effects Laboratory, National Space Science Center, Chinese Academy of Sciences, Beijing, China
2School of Astronomy and Space Science, University of Chinese Academy of Sciences, Beijing, China
3College of Internet of Things Engineering, Hohai University, Nanjing, China
*Email: zhuxiang@nssc.ac.cn

A large number of studies are there for Advanced Encryption Standard (AES) fault attack analysis, but less for fault recognition. This paper presents a recognition method for single-byte fault which is induced in the last three rounds of AES. Studying the differential characteristics of Sbox, the single byte fault induced in ninth round or tenth round will be identified respectively with 9.3 and 9.1 ciphertexts. For the fault induced in eighth round, the fault value can be obtained with 188.5 ciphertexts by analyzing the differential features of two Sboxes and MixColumns. As an auxiliary means for fault attacks, this method is used to realize the byte or bit level physical positioning of confidential data in the encryption chip, which is beneficial to reduce the blindness of the attacker’s experiments and obtain the sensitive area of fault attack.

Introduction: Fault Attack (FA) is an active non-invasive or semi-intrusive attack method. It’s generally divided into two steps, fault injection and fault analysis. There are many different methods of fault injection, such as a laser, voltage spike, clock glitch and focused ion-beam [1–3]. The analysis models for fault analysis include differential fault attack (DFA), collision fault attack (CFA), invalid fault attack (IFA) or algebraic fault attack (AFA). DFA is to obtain the security key by solving the differential equation of the fault differential, which is an efficient means of fault attacks. Generally, fault can be induced in the last three rounds of AES, and each round has its own fault analysis model respectively [4].

There are a lot of literatures on AES fault attacks, but few on fault recognition. For an embedded security chip, the real physical location of secret data is not clear before attacking, so an adversary would have great blindness in the process of fault injection. Therefore, it is necessary to study the features of fault to reduce blindness and time cost of an attack. The fault recognition method that we propose can accurately identify the single byte fault in last three rounds of AES with less ciphertext cost, and help attacker to check the wrong byte and the fault differential value. In fact, adversary can also use this method to further locate the physical position of each bit or byte of the secret data and reveal the sensitive areas of Laser Fault Injection (LFI) in the SRAM of security chip, which is of great significance to improve attack efficiency and the success rate of experimental repetitiveness. To apply this method, the following conditions must be considered:

• The fault must be induced in the operation from eighth, ninth or tenth round. Due to the strong diffusion ability of AES, those faults produced from seventh or earlier rounds cannot be inferred.

• The fault must be produced from a fixed-byte modification. Compared with voltage spike or clock glitch, laser can easily produce single byte fault.

For the tenth round, the correct input differential can be calculated by statistical analysis of the output differential of single Sbox. For the ninth round, 4 bytes of subkey needs to be calculated in advance. As for the eighth round, adversary needs to calculate the tenth subkey through the DFA model. For AES-128, adversary can obtain the fault differential by decryption in principle, but it is not appropriate for AES-192/256. For the sake of generality, this paper analyzes the statistical characteristics of two Sbox to calculate the fault differential of the eighth round.

Fig. 1 Fault propagation of the tenth round

This article is organized as following: Section 2 briefly introduces the AES algorithm. Section 3 discusses fault recognition of last three rounds. Then, Section 4 introduce the practical application of this method, which reveals the physical location of each bit or byte of secret data in SRAM. Finally, we get to make a summary of the article.

AES algorithm description: The AES algorithm is a block cipher algorithm released by the National Institute of Standards and Technology (NIST) in 2001 to replace the Data Encryption Standard (DES). The length of a data packet is 128 bits, and the length of the key can be 128, 192 or 256 bits as needed. The plain text is arranged as a 4 × 4 bytes array, arranged in a certain order, which is called the State Matrix. AES is an iterative block cipher based on the SPN structure of finite field operations. Each round includes non-linear layer SubBytes and linear layer ShiftRows, MixColumns and AddRoundKey, but the last round does not include MixColumns. The number of iterations of the three versions of AES-128/192/256 is 10, 12 and 14, respectively.

Fault recognition for the last three rounds: The idea of this part is that fault recognition is to give a certain number of fault ciphertext pairs \(X\) and output fault byte \(\alpha \in \{0, 1, 2, \ldots, 15\}\) and fault differential \(f \in \{1, 2, \ldots, 255\}\). First of all, it is necessary to determine the position of the faulty round. It is easy to know that the fault ciphertexts of the ninth and tenth rounds have only 4 or 1 wrong byte respectively, and all the bytes are wrong in the eighth round (further verification is required to distinguish the fault from eighth round or earlier); then, the fault byte \(\alpha\) and fault differential \(f\) are obtained according to the following ideas.

Fault recognition of the tenth round: If the fault occurs after the MixColumns of ninth round, the ciphertext is just one byte error. As shown in Figure 1, suppose \(f\) represents the single-byte differential caused by fault injection. Since SubBytes is a non-linear operation, it has to be discussed in the following two situations: In the first case, the error \(f\) is injected before the tenth round of byte replacement. Due to the non-linear transformation of SubBytes, the fault byte differential \(f\) of the ciphertext is an indeterminate value; the second type, the error is introduced after SubBytes, the ciphertext byte fault differential brought is a fixed value \(f\). The value is calculated by Xoring the correct and wrong bytes of the ciphertext. Two ciphertext pairs can be used to distinguish those two kinds of faults: if the two differentials are equal, we can directly output the fault differential, because the probability that the differentials of the two ciphertext bytes after 1 Sbox are equal is only 1/127.

Suppose that the fault differential was introduced before SubBytes of the ninth round, that is, the input differential \(f\) of Sbox is a fixed value and \(f\) is the output differential of the Sbox. Due to the incomplete differential of the Sbox, \(f\) has only 127 values instead of 256. Therefore, this question can be transformed into using the output differential of the Sbox for each ciphertext to obtain a set of input differentials, and through the intersection of these sets, the unique input differential is finally determined. For the input differential of 0x11, the input of the Sbox is randomly generated and meets this differential. The relationship between the number of ciphertext pairs required and the number of times of 255 input differentials are shown in Figure 2 in a single experiment. The red circle in Figure 2 represents the minimum number of ciphertexts that can distinguish the differential between the correct and the incorrect input differentials. It can be noted that, since the correct differential will appear every time, the curve is approximately linearly increasing, while the incorrect differential does not in line with the trend. After 10,000 simulation of all 255 possible values, it is found that the average number is about 9.3. When more than 10 ciphertexts are used, the probability of getting the correct differential is.
corresponds to 127 input differentials, and all the 255 input differentials distinguish the correct differential. The output differentials of each Sbox evenly distributed in the range of 1–255, then the Sbox differential determines that the only correct input differential will appear every time, after the remaining 254 differentials are filtered by N output differentials, the average number of differentials that occur N times is 254 × (255/254)^N. If the number of differentials that occur N times is less than 1, we get N > log 255/254 ≈ 7.90. This means that if one select about 8 output differentials, except for the correct differential, the average number of differentials that appear eight times is only 1; then, if the number is less than 1, at least nine differentials are required, which is quite close to the simulation result.

Fault recognition of the ninth round: If the fault occurs between two MixColumns of eighth round and ninth round, there will be 4 faulty bytes in the ciphertext. If the fault occurs before the ninth round of SubBytes, the fault differential pass through two SubBytes operations and a single MixColumns operation; if the fault occurs after the SubBytes, the fault differential goes through 1 SubBytes operation and 1 MixColumns operation. In either case, after mixed operation of the ninth round, the fault differential satisfies a certain proportion relationship. For faults introduced after SubBytes operation, the fault recognition is completed by obtaining the fault differential before the MixColumns of ninth round with the DFA model. Let Δ represents the differential of the second byte caused by fault injection. \( x_i \) (0 ≤ i ≤ 16) represents the intermediate values of each byte after MixColumns, and \( (C_{12}, C_6, C_3, C_7), (C_{12}, C_6, C_3, C_7) \) represent the ciphertext before and after the change, respectively, then

\[
\begin{align*}
\text{Sub}(X_{12} \oplus \Delta) \oplus \text{Sub}(X_{12}) &= C_{12} \\
\text{Sub}(X_6 \oplus \Delta) \oplus \text{Sub}(X_6) &= C_6 \\
\text{Sub}(X_3 \oplus \Delta) \oplus \text{Sub}(X_3) &= C_3 \\
\text{Sub}(X_7 \oplus \Delta) \oplus \text{Sub}(X_7) &= C_7
\end{align*}
\]

Solving Equation (1) can obtain the corresponding four subkey bytes and the fault differential \( \Delta \) [5] of the ninth round of MixColumns input. 10,000 random simulation of 16 ciphertexts shows that the number of ciphertext pairs required to obtain a four-byte complete key is about 4.1. For the second type of fault, the SubBytes operation of the ninth round results in an approximately random differential value, so the recognition process is a little more complicated than the first type. Four key-bytes should be obtained by solving Equation (1) firstly. Then input differentials of the ninth MixColumns are calculated for each ciphertext to obtain the set of output differentials of the ninth SubBytes. Finally, using the similar method as before to calculate the fault differential value. As shown in Figure 3, the simulation found that the average number of ciphertexts required to identify the second type of fault by using this method is 9.1. Using 10 ciphertexts can obtain the input differential with a probability of about 75%, while 13 ciphertexts exceed 95%, basically the same as the tenth round.

Fault recognition of the eighth round: If the fault occurs between the MixColumns of seventh round and eighth round, all 16 bytes of ciphertext will be wrong. In order to calculate the differential, this paper adopts DFA method of the eighth round. [4] pointed out that using 1 pair of ciphertext can make the space of key candidate reduced to 2^32, and using two pairs of ciphertexts will recover the complete subkey of the tenth round. For the AES-128, it is enough to get the complete key relying on the tenth subkey, but AES-192/256 cannot. Therefore, this paper adopts a more general approach to judge the fault differential: using the statistical method similar to the former to calculate the input fault differential. If a fault is induced in the input of the eighth round, let \( f \) represent the fault differential, then the fault propagation mode is shown in Figure 4. Figure 5 shows the simulation of the fault differential 0x02 with four kinds of ratio caused by MixColumns of the eighth round. It can be seen that only in the first ratio there is a significant separation, which reveals the correct ratio and differential. 10,000 fault injections and fault recognition simulations were performed on all 16 bytes. Then the results show that 188.5 faulty ciphertext pairs are enough to complete the fault recognition of the eighth round.

Attack practice: This paper selects a commercial microcontroller unit (MCU) ATMega163L for laser fault attack. It has 1K bytes of internal SRAM data memory, 16K bytes of Flash program memory and 4 MHz maximum frequency. A standard AES-128 algorithm is implemented in the MCU. When MCU performing encryption, SRAM often saves secret data of the cryptographic algorithm, such as AES state matrix and subkey data, so we use the SRAM module as the target of attack.
The experimental platform relies on the single event effect (SEE) pulsed laser experimental system, which mainly includes 1064 nm laser optical systems, 3D mobile station, synchronous control systems and control computers. The experimental system can control the injection laser with high temporal and spatial accuracy. Figure 6 shows the layout taken from the backside and the target area of SRAM. By adjusting the time of laser injection, the fault occurs at a certain moment in the last three rounds. Then ciphertexts will be collected and the fault differential will be calculated.

**Attack result:** We locate the secret data in the SRAM with the proposed method. First, we set the trigger time of laser in a suitable range (in the last three rounds) which can be identified by power consumption. Then we scan the entire SRAM area with 3D mobile station and laser equipment to record the coordinate where fault occurs, and we use the proposed method to identify the fault bytes and flipped bits. Finally, we get the distribution of secret data in the SRAM. As shown in the Figure 6(a), a 900μm × 900μm area covering the SRAM is targeted as the attack area. After adjusting the time of laser injection, we finally injected the fault into the input of ninth round. Using the fault recognition method we proposed, recording the fault byte and location information, the distribution of AES state matrix is obtained, as shown in Figure 7.

Analyzing the distribution characteristics of these bits or bytes, we find that:

1. The type of fault in the experiment is almost single bit flip.
2. Each column stores 8 bytes of data, and those bytes are arranged in the negative direction of the y-axis.
3. The same bits of different bytes in each column are placed adjacent to each other, and different bits of the same byte are separated.

**Conclusion:** The proposed fault recognition method can accurately identify the single byte faults in last three rounds of AES with less ciphertexts cost. It helps attackers to recognize the details of the fault and reduce the blindness of attack. With the laser attacks technology, we located confidential data at the bit level. This method is of great significance in improving attack efficiency and experimental reproducibility.

**Acknowledgements:** This work was supported by The Key Deployment Projects of Chinese Academy of Sciences (KGFZD-135-16-005) and The Space Science Advance Research Projects of Chinese Academy of Sciences (XDA15014600).

© 2021 The Authors. *Electronics Letters* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Received: 30 November 2020   Accepted: 1 April 2021   doi: 10.1049/ell2.12188

**References**

1. Joye, M., Tunstall, M.: Fault Analysis in Cryptography. Information Security and Cryptography Series, pp. 295–311. Springer, Heidelberg (2012)
2. Agoyan, M., et al.: When clocks fail: On critical paths and clock faults. In: International Conference on Smart Card Research and Advanced Applications, Passau, pp. 182–193 (2010)
3. Torrance, R., James, D.: The state-of-the-art in IC reverse engineering. In: International Workshop on Cryptographic Hardware and Embedded Systems, Lausanne, pp. 363–381 (2009),
4. Mukhopadhyay, D.: An improved fault based attack of the Advanced Encryption Standard. In: International Conference on Cryptology in Africa, Gammarth, pp. 421–434 (2009).
5. Dusart, P., Letoumeaux, G., Vivolo, O.: Differential fault analysis on AES. In: International Conference on Applied Cryptography and Network Security, Kunming, pp. 293–306 (2003)