AB-SIFA: SIFA with Adjacent-Byte Model

Chunya Hu[1] , Yongbo Hu[1], Wenfeng Zhu[1], Zinxin Tan[1], Qi Zhang[1] , Zichao Gong[1], Yanhao Gong[1], Luyao Jin[1] and Pengwei Feng[1]

[1] Vulnerability Analysis Laboratory, Goodix Technology, Shanghai, China
hcy_0323@163.com, huyongbo@goodix.com

Abstract. Statistical Ineffective Fault Attack (SIFA) has been a threat for implementations of symmetric cryptographic primitives. Unlike Differential Fault Attacks (DFA) which takes both correct and faulty ciphertexts, SIFA can recover the secret key with only correct ciphertexts. The classic SIFA is only effective on fault models with non-uniform distribution of intermediate value. In this paper, we present a new fault model named adjacent-byte model, which describes a non-uniform distribution of relationship between two bytes (i.e. exclusive-or). To the best of our knowledge, it is the first time that this fault model has been proposed. We also show that the adjacent-byte faults can be induced by different fault sources and easy to reproduce. Then a new SIFA attack method called AB-SIFA on symmetric cryptography is proposed. We demonstrate the effectiveness of this new attack by simulating the attack. Finally, our attacks are applied to a software implementations of AES-128 with redundant countermeasure and a hardware AES co-processor, utilizing voltage glitches and clock glitches.

Keywords: Fault Attack, Fault Model, Statistical Ineffective Fault Attack, AES.

1 Introduction

Fault attacks [6, 2], e.g. clock/voltage glitches [4, 8, 11], ElectroMagnetic (EM) perturbations[20] and laser injection [12,19], are proven to be practical attacks against implementations of cryptographic algorithms. Among these fault sources, laser injection is an invasive attack, which requires de-packaging the device to have direct access to its components. Also, laser injection can produce fine-grained errors. Clock/voltage glitches and EM perturbations are non-invasive attacks, which are practical and low-cost.

Typically, a fault is injected during the cryptographic algorithm's processing and then algebraic approaches or statistical distinguishers are used to derive the secret key. The first successful fault attacks were present by Boneh et al. in 1997 [2]. Later, in 2001, they injected a random fault in RSA based on Chinese Remainder Theorem (CRT) to reveal the secret key [13]. Their findings triggered further research on how to break cryptographic algorithms by injecting a single fault during its execution. Since the seminal work of Boneh et al., lot of papers have proposed fault attack on several

widely used cryptographic primitives, which includes symmetric cryptographic algorithms DES [3] and AES [9], as well as asymmetric cryptographic algorithms DSA [14] and Elliptic Curve Cryptograph [10].

1.1 Related work

Biham et al. first introduced a far-reaching attack method called Differential Fault Attacks (DFA) which is the most prominent fault analysis technique [3]. DFA exploited the difference between correct and faulty ciphertexts. This technique was first introduced for asymmetric ciphers, and quickly extended to symmetric ciphers.

Statistical Fault Attacks (SFA), which originally proposed by Fuhr et al. [7], as a method to recover the secret key by limited number of faulty ciphertexts. In contrast to DFA, SFA requires solely one AES-128 state byte following any (possibly unknown) non-uniform distribution. It tried to attack the last 4 rounds of AES-128, and concluded that the best result would be obtained if the target was one byte before the last MixColumns application.

In contrast to SFA, Ineffective Fault Attacks (IFA) by Clavier [5], relied on exploiting faulted encryptions where the induced faults were ineffective. It induced a stuck-at-0 fault in one byte during one execution of AES. If faults on targeted exclusive-or instruction are ineffective, then the normal exclusive-or outputs are simultaneously equal to zero.

Recently, Statistical Ineffective Fault Attack (SIFA), a combination of SFA and IFA, was first proposed by Dobraunig et al. [1]. While IFA requires precise knowledge of a fault, SIFA has much more relaxed requirements, thus it can utilize various different models. Many different countermeasures [15-17] are ineffective for SIFA, because only correct ciphertexts are required. Since then, many SIFA-related works have been published [21-22].

1.2 Contribution

In this paper, a new fault model named adjacent-byte model is introduced for SIFA. And we demonstrate that it can be induced by voltage glitches. In addition, adjacent-byte model also can be used on other fault attacks. We also proposed AB-SIFA which is a new SIFA attack method with adjacent-byte model on symmetric cryptography. And the effectiveness of AB-SIFA is analyzed through simulation. Then, we have successfully retrieved the secret key of an AES-128 software implementation on ATMEGA2560 and a hardware implementation on ATXMEGA256A3. The voltage glitches and clock glitches are used in our practical attacks.

Organization. The rest of this paper is organized as follows. The background was introduced in Section 2, and the adjacent-byte model is described in Section 3. In Section 4, we present AB-SIFA and analyze it with simulation. And Section 5 contains the results of practical fault attacks for both a software implementation and a hardware implementation of AES-128. Finally we conclude in Section 6.

2 Background

2.1 Notations

The intermediate States obtained before and after round *r* are denoted by *Sin* and *Sout*. The key used in round *r* is denoted by *Kr*, and the ciphertext is denoted by *C* in AES. The operations of AES, i.e. *SubBytes*, *ShiftRows*, *MixColumns*, *AddRoundKey*, are denoted by *SB*, *SR*, *MC*, *AK*, and SB^{-1}, SR^{-1}, MC^{-1}, AK^{-1} for their inverse operations.

2.2 Fault Model

A fault model defines one kind of fault during cryptographic algorithm calculation caused by fault attack on the chips or FPGA. From the fault model, the designer or user can then efficiently and effortlessly predict the consequences of this particular fault. Two kinds of fault models are widely used which are bit fault model and byte fault model. Since it is sophisticated and high-cost to introduce the bit fault, only byte fault model was considered here. The generic fault models include stuck-at model, bit-flip model and random-and model.

- Stuck-at model: The stuck-at model describes a class of faults where the byte is always 0 or 0xFF.
- Bit-flip model: The bit-flip model describes a class of faults where some bits in a byte are flipped.
- Random-and model: The random-and model describes a class of faults where a byte logically AND with a random number.

2.3 Squared Euclidean Imbalance

The Squared Euclidean Imbalance (SEI) [1, 7] is a tool to measure imbalance. The formula for SEI is as follows:

$$\mathrm{SEI}(p) \coloneqq \sum_{x \in X} (p(x) - \theta(x))^2 \tag{1}$$

where X is p is the real distribution got after practical attacks, and θ is a nearly uniform distribution. In this paper, it also be used to measure the bias.

2.4 Statistical Ineffective Fault Attacks

SIFA [1] relies on different fault models including stuck-at-0 model, random-and model and “imperfect” bit-flip model. “Imperfect” bit-flip model means the probability of 1 flipping 0 is different from that 0 flipping 1. Therefore, it is very inclusive of different fault types.

The authors supposed that a byte of the intermediate values before last MixColumns is unevenly distributed caused by fault attack. By the last MixColumns, one attacked byte will influence 4 bytes in the last round. Therefore, they guessed the 4 bytes of the last round key firstly and partially decrypts the last operation of correct ciphertexts to

obtain a partial state. Incorrect key hypotheses resulted in a uniformly distributed state, while for the correct hypothesis the state distribution was non-uniform. The key candidate that gives the highest SEI is the guessed one. And their practical attacks rely on fault induction by means of clock glitches.

3 Adjacent-Byte Model

3.1 Definition

Here we define a new fault model at byte level and name it adjacent-byte model. It describes that one intermediate value is equal to another by fault attack and usually these two intermediates are adjacent during the computation.

This fault model can be obtained by fault attack. Because whether it is attackers use glitch attack, EM perturbation, or laser injection, there is a probability that to disturb the program execution and skip an instruction during a cryptographic algorithm calculation. If there are two adjacent assignment operations or calculation operations in the program, skipping the later one by fault injection may lead to the results that these two operations become the same one. Hence, this fault model can be reproduced in practical attacks.

3.2 Experimental Study of Model

A software AES-128 was implemented on ATMEGA2560 for evaluation. For simplicity, we implemented a AES-128 with look-up table for SubBytes which was the most popular way nowadays. And the secret key of AES was known for reverse analysis. The following Listing is the code of SubBytes.

Listing 1 SubBytes for adjacent-byte model analysis

```
static uint8_t s_box[256] PROGMEM = {0x63,0x7c,0x77,…,0x54,0xbb,0x16};
void sub_bytes(uint8_t *state) {
    uint8_t i, j;
    uint8_t row, col;
    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            row = (state[4*i+j] & 0xf0) >> 4;
            col = state[4*i+j] & 0x0f;
            state[4*i+j] = s_box[16*row+col];
        }
    }
```

To get the faults, voltage glitches was used in this experiment, of which the voltage was manipulated to be lower or higher during the cryptographic algorithm calculation. The device still worked, but faults occurred during the calculation.

First, the approximate location of the SubBytes in 9th round was determined through the side channel information on power trace. Then, with this information, voltage glitches attack was conducted on 10k runs of AES with the same plaintext. After fault attacks, the faulty ciphertexts were collected, and then the values of SubBytes in 9th round using known key was also computed to analyze the results of the attack. Finally, there were 2559 faulty ciphertexts with either bit-flip faults, random-and faults or adjacent-byte faults. In detail, here 1429 ciphertexts (about 55.8%) had two same bytes and 807 (about 31.5%) ciphertexts had one bytes in SubBytes of 9th round. And also some of the remaining ciphertexts had more faulty bytes, which may result by accidental attack on the other operation of AES.

It proved that the adjacent-byte fault model could be caused by the voltage glitches. And sometimes it is more likely to occurred than the other fault model in the practical attack.

4 AB-SIFA

This section described a new fault model and the proposed attack. This attack was presented on implementations of AES-128 as example. It can be easily applied for other symmetric cryptographic algorithm as well.

4.1 Motivation

Faults induced by fault attack (e.g., voltage glitches, clock glitches) can be distinguished into two classes: faults on program flow and faults on data flow. Faults on program flow means one instruction to be replaced by another, which may result in an algorithm modification. On the contrary, faults on data flow only affect data, without modifying any instructions. Typically, these two classes lead to very similar visible outputs. Either kind fault can cause the faulty intermediate value v' from v.

In classic SIFA [1], authors considered two cases, one where the value of v' was independent of the value v, like byte-stuck-at faults, random faults, and the other was that v' depends in some sense on v, like bit-stuck-at faults, random-and faults. We think they only considered the faults on data flow, but ignored the faults on program flow which may result in v' having a vault independent of v but related to anther intermediate value w.

Take AES as an example, the theoretical probability that two intermediate values are equal in a specific round is 1/256. Assume that a fault can always injected successfully on the 9th round SubBytes which causes the 9th byte to be equal to the 10th byte. Then the probability that these two bytes are equal in every correct ciphertext will be 100%. Then a non-uniform distribution can be obtained after fault injection. This fault model is mentioned in section 3 and named adjacent-byte model. The fault distribution table is shown in Table 1. Assume that the value of w is 0x63. It shows that whatever the value of v is, v' is 0x63.

Table 1. Fault distribution table for adjacent-byte model.

v / v′	0x00	0x01	…	0x63	…	0xEF	0xFF
0x00	0	0	…	1	…	0	0
0x01	0	0	…	1	…	0	0
…	…	…	…	…	…	…	…
0x63	0	0	…	1	…	0	0
…	…	…	…	…	…	…	…
0xEF	0	0	…	1	…	0	0
0xFF	0	0	…	1	…	0	0

Moreover, in practical attack, adjacent-byte faults occurred frequently by different fault sources. To solve this situation, we propose a new SIFA attack in this paper.

4.2 Description of the Attack

To describe AB-SIFA method, we also take AES-128 as an example.

Side Channel Analysis. Before the fault injection, the attacker should know the approximate location of calculation for 9th round. Thus, the side channel analysis is used to obtain the power consumption and runtime information of AES-128.

Fault Injection. In classical block ciphers, non-linear operation offers a meaningful target for attack. Hence, a fault can be injected during the 9th round of SubBytes shown in Figure 1. It is assumed that one intermediate value of the 9th round SubBytes output is faulty, and there is a probability that this byte is same with another byte. For example, consider $s_{0,3}$ and $s_{1,0}$ are equal after fault attack, and the exclusive-or value of them is 0. When $s_{0,3}$ and $s_{1,0}$ are equal before attack, the correct ciphertexts will be obtained. These ciphertexts are *Successful but Ineffective Ciphertexts*. When $s_{0,3}$ and $s_{1,0}$ are not equal before attack, the outputs will be faulty ciphertexts if the fault injection is succeed. These ciphertexts are called *Successful and Effective Ciphertexts*. And usually the attackers cannot get them, because of redundant countermeasure. If the fault injection is failed, the *Unsuccessful Ciphertexts* will be obtained. Therefore, the exclusive-or value of $s_{0,3}$ and $s_{1,0}$ is non-uniform after attack.

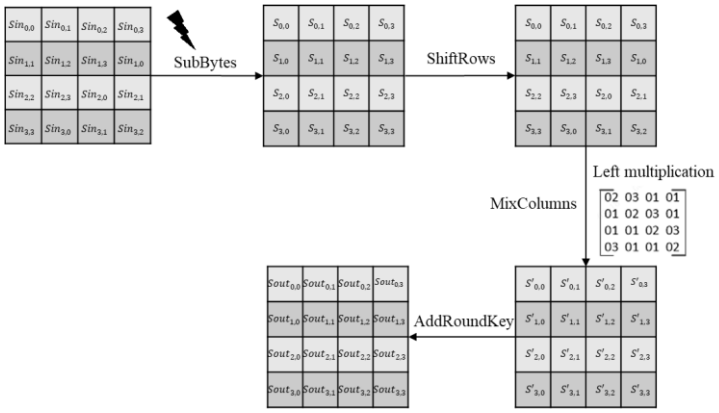


Fig. 1. Algorithm flow of the 9th round

Key Determining. Plenty of ciphertexts were collected after fault injected. Since $s_{0,3}$ and $s_{1,0}$ are in the same column, the last 4-byte round key ($K10_{0,3}$, $K10_{1,3}$, $K10_{2,3}$, $K10_{3,3}$) are guessed. And the last operation of each correct ciphertext is partially decrypted to obtain a 4-byte $SB9'$ which is different from the 9th round SubBytes $SB9$.

$$SB9' = SR^{-1}(MC^{-1}(SB^{-1}(SR^{-1}(K10 \oplus C)))) \tag{2}$$

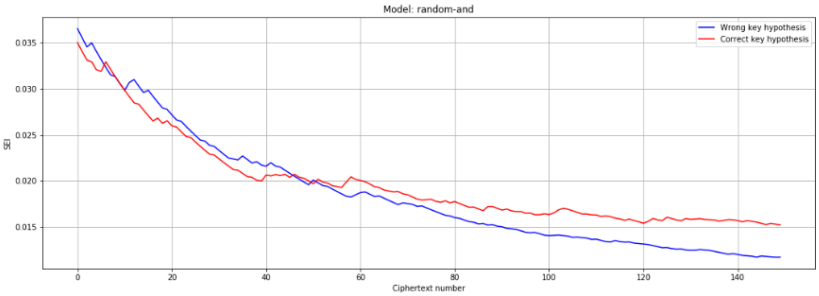
$$SB9 = SR^{-1}(MC^{-1}(K9 \oplus (SB^{-1}(SR^{-1}(K10 \oplus C))))) \tag{3}$$

In calculation of $SB9'$, the 9th round AddRoundKey is ignored. When the secret key is fixed, it does not affect the distribution of the bytes. Thus, it also has no influence on the bias of $s_{0,3} \oplus s_{1,0}$.

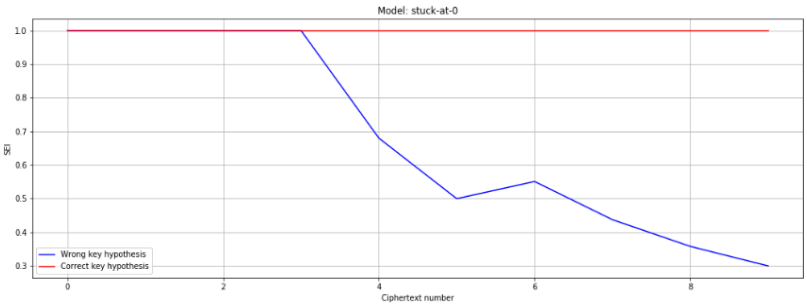
For each key guess, the distance between the distribution of $s_{0,3} \oplus s_{1,0}$ and a uniform distribution can now be measured. Finally, the correct key will lead to the distribution with highest SEI. Notice that if two bytes are equal in different columns in the 9th round MixColumns, the last 8-byte round key should be guessed.

4.3 Simulated Analysis

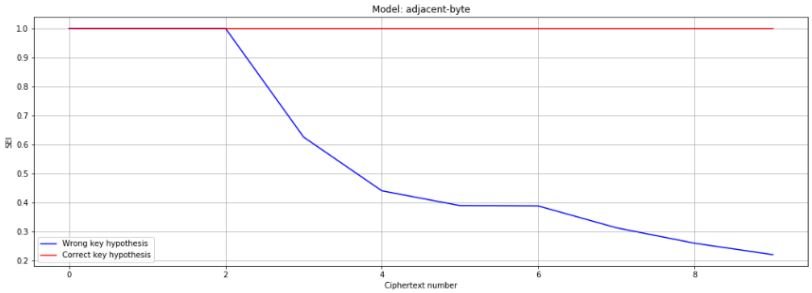
In this section, we illustrate AB-SIFA on adjacent-byte model more detail by simulated analysis. To compare with the classic SIFA, the theoretical analysis was conducted on different fault models including stuck-at-0 model and random and model.



(a) The numbers of ciphertexts needed for SIFA with random-and model



(b) The numbers of ciphertexts needed for SIFA with stuck-at-0 model



(c) The numbers of ciphertexts needed for AB-SIFA with adjacent-byte model

Fig. 2. The analysis results of SIFA or AB-SIFA on three different model

Assume that the attackers succeed for every fault attack on 9th round SubBytes of AES-128. Thus, all the bytes mentioned below were the bytes in 9th round SubBytes. There are 1000 correct ciphertexts with random plaintexts and a fixed key. Among them, we used 100 Successful and Effective Ciphertexts for each of three models. The results are shown in Figure 2. In the simulated analysis, the results shew that the adjacent-byte model was very similar to the stuck-at-0 model. And it needs minimum number of successfully attacked ciphertexts. Compare to stuck-at-0 model and adjacent-byte model, random-and model needs more ciphertexts to recover the key.

5 Practical Attack and Result Analysis

In this section, we successfully break AES-128 using SIFA with adjacent-byte model. A software implementation and a hardware implementation of AES-128 are attacked by voltage glitches and clock glitches in our experiments.

5.1 Target and Experiment Setup

Target. The targets of evaluation (TOE) are two Atmel AVR based micro-controllers. The target on ATMEGA2560 is a publicly available software AES-128 implementation from AVRCryptoLib [18] protected by redundant countermeasure. Hence, in every fault attack, AES-128 will be calculated twice. If and only if the results are the same, the ciphertext is output. The second target of evaluation is a hardware AES-128 implementation on ATXMEGA256A3.

Experiment Setup. The voltage/clock glitch fault injection system shown in Figure 3 is composed of a control desktop, TOE, a voltage generator and a clock generator. The desktop is the main controller of this system which controls the generators and TOE.

In voltage glitch attack, the voltage of TOE is generated by voltage generator with its internal clock. While clock glitch attack needs an external clock generated by the clock generator with a stable operating voltage.

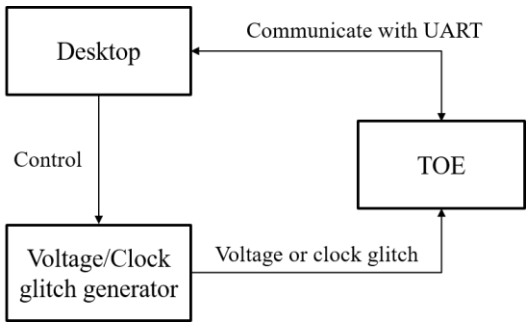


Fig. 3. Glitch fault injection platform connection and setup

5.2 Attack on Software Implementation of AES

We demonstrated the applicability of adjacent-byte model for an AES-128 from the AVRCryptoLib [18] implementation on ATMEGA2560. The code of each round of AES was shown in Listing 2. SubBytes was implemented by simple one-byte assignment. According to the attack strategy we proposed before, the SubBytes could be attacked targets.

In this experiment, we attacked the calculation of SubBytes in 9th round. In total, 2303 correct ciphertexts corresponding to the ineffective fault after 10k fault injections were obtained. *Successful but Ineffective Ciphertexts* and *Unsuccessful Ciphertexts* consist the correct ciphertexts set.

Listing 2 Each round of software implementation AES

```
static uint8_t aes_sbox[256] PROGMEM = {0x63,0x7c,0x77,…,0x54,0xbb,0x16};
static void aes_enc_round(aes_cipher_state_t *state, const aes_roundkey_t *k)
{
    uint8_t tmp[16], t;
    uint8_t i;
    /* SubBytes */
    for (i = 0; i < 16; ++i) {
        tmp[i] = pgm_read_byte(aes_sbox + state->s[i]);
    }
    /* ShiftRows */
        ……
     /* MixColumns */
        ……
     /* AddRoundKey */
        ……}
```

Using this ciphertext set, 4-byte *K10* was successfully recovered by AB-SIFA. The result of attack presented in Figure 4 showed the number of correct ciphertexts needed until the correct key candidate could be reliably distinguished. Due to the noise of *Unsuccessful Ciphertexts*, about 700 correct ciphertexts were required to recover the 32-bit key used in 9th round.

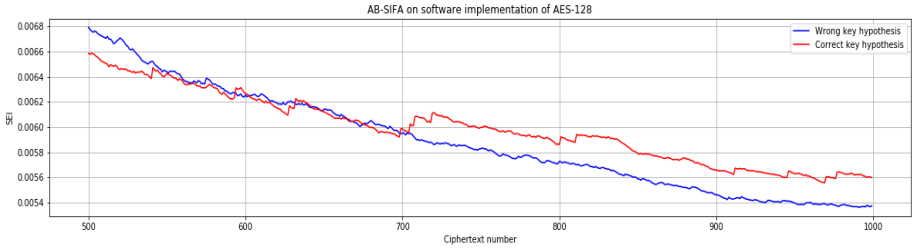


Fig. 4. The result of AB-SIFA for AES-128 on ATMEGA2560

However, the classic SIFA failed for this ciphertext set. We thought it was because in this set the number of faults required by SIFA was not enough. The detail results were in Figure 5.

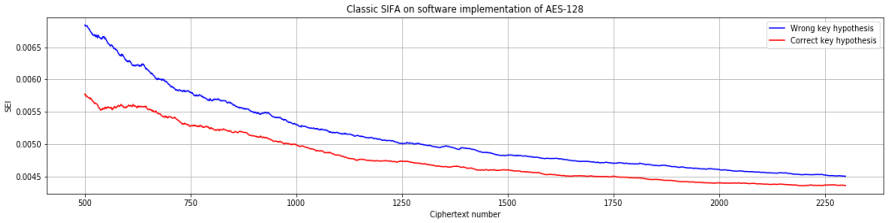


Fig. 5. Classic SIFA attack result for AES-128 on ATMEGA2560

5.3 Attack on Hardware Implementation of AES

The AES hardware crypto engine on ATXmega256A3 with 2MHz CPU frequency was evaluated as well. To illustrate the variety of fault attacks that could produce adjacent-byte model, the hardware AES was attacked by simple clock glitches in this experiment.

The attack method also injected fault in the 9th round of AES, so the external clock just needed to provide a faster clock at some point of the 9th round and kept normal for other rounds. The output clock from the clock-glitch generator was observed through Picoscope and presented in Figure 6 where the second clock cycle was split into two faster cycles.

Fig. 6. Shape of clocks

The clock glitches injected to the AES-128 calculation. As result, 1479 correct ciphertexts after 10k times fault injections were obtained. The Figure 7 showed the number of correct ciphertexts required for key recovering.

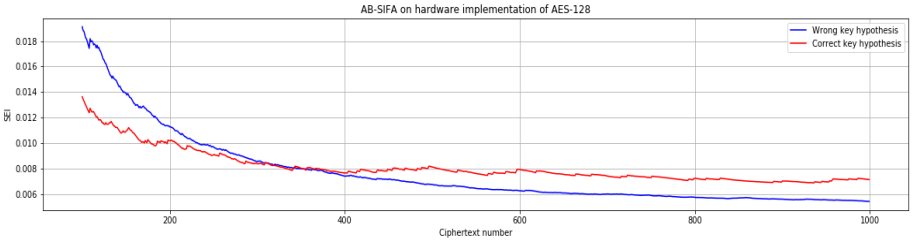


Fig. 7. The result of AB-SIFA for AES-128 on ATXmega256A3

6 Conclusion

In this paper, we presented a new fault model named adjacent-byte model which complemented the practical fault models in classic SIFA. Meanwhile, a SIFA-like method for this fault model was presented to recover the secret key. This attack can be easily extended to many other symmetric ciphers.

To demonstrate the feasibility of this fault model and attack in a practical attack, we successfully attack a software implementation of AES-128 on ATMEGA2560 and a hardware implementation of AES-128 on ATXMEGA256A3 with this model. The result shows that this attack can exploit many practical implementations protected with redundancy-based countermeasures against faults.

References

1. Dobraunig, C., Eichlseder, M., Korak, T., Mangard, S., Mendel, F., Primas, R.: SIFA: exploiting ineffective fault inductions on symmetric cryptography. IACR Transactions on Cryptographic Hardware and Embedded Systems, 547-572 (2018).
2. Boneh, D., DeMillo, R. A., Lipton R. J.: On the importance of checking cryptographic protocols for faults. In: International conference on the theory and applications of cryptographic techniques, vol. 1233, pp. 37-51. Springer, Berlin, Heidelberg (1997).
3. Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: Annual international cryptology conference, pp. 513-525. Springer, Berlin, Heidelberg (1997).
4. Moro, N., Dehbaoui, A., Heydemann, K., Robisson, B., Encrenaz, E.: Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller. In: 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, pp. 77-88. IEEE (2013).
5. Clavier, C.: Secret external encodings do not prevent transient fault analysis. In: International Workshop on Cryptographic Hardware and Embedded Systems. Springer, pp. 181-194. Berlin, Heidelberg (2007).
6. Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: Annual international cryptology conference, pp. 513-525. Springer, Berlin, Heidelberg (1997).
7. Fuhr, T., Jaulmes, É., Lomné ,V., Thillard, A.: Fault attacks on AES with faulty ciphertexts only. In: Fischer, W. and Schmidt, J. (eds) Fault Diagnosis and Tolerance in Cryptography –FDTC 2013, pp. 108-118. IEEE (2013).
8. Kömmerling, O., Kuhn, M. G.: Design Principles for Tamper-Resistant Smartcard Processors (99), 9-20 (1999).

9. PiretY, G., Quisquater, J. J.: A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD. In: Walter, C.D., Koc, C.K., and Paar, C. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2003, vol. 2779, pp. 77–88. Springer (2003).
10. Biehl, I., Meyer, B., and Muller, V., Differential: Fault Analysis on Elliptic Curve Cryptosystems. In: Bellare, M. (eds.) Advances in Cryptology – CRYPTO 2000, vol. 1880, pp. 131–146. Springer (2000).
11. Skorobogatov, S.: Fault attacks on secure chips. Design and Security of Cryptographic Algorithms and Devices (2011).
12. Van Woudenberg, J. G., Witteman, M. F., Menarini, F.: Practical optical fault injection on secure microcontrollers. In: Breveglieri, L., Guilley, S., Koren, I., Naccache, D., and Takahashi, J., (eds.) 2011 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2011. pp. 91-99. IEEE Computer Society (2011).
13. Boneh, D., DeMillo, R. A., Lipton, R. J.: On the importance of eliminating errors in cryptographic computations. Journal of Cryptology (14), 101–119 (2001).
14. Bao, F., DengY, R. H., Han, Y., Jeng, A., Narasimhalu, A. D., and Ngair T.: Breaking public key cryptosystems on tamper resistant devices in the presence of transient faults. In: pp. 115–124. Springer, Berlin, Heidelberg (1997).
15. Tupsamudre, H., Bisht, S., Mukhopadhyay, D.: Destroying fault invariant with randomization - A countermeasure for AES against differential fault attacks. In: Batina, L. and Robshaw, M. (eds) Cryptographic Hardware and Embedded Systems – CHES 2014, LNCS, vol.8731, pp. 93–111. Springer (2014).
16. Bar-El, H., Choukri, H., Naccache, D., Tunstall, M., Whelan, C.: The sorcerer's apprentice guide to fault attacks. Proceedings of the IEEE 94(2), pp.370–382 (2006).
17. Patranabis, S., Chakraborty, A., Nguyen, P. H., Mukhopadhyay, D.: A biased fault attack on the time redundancy countermeasure for AES. In: Mangard, S. and Axel, Y. P. (eds.) Constructive Side-Channel Analysis and Secure Design – COSADE 2015, LNCS, vol.9064, pp. 189–203. Springer, Cham (2015).
18. AVR crypto lib, https://git.cryptolib.org/?p=avr-crypto-lib.git.
19. Selmke, B., Brummer, S., Heyszl, J., Sigl, G.: Precise laser fault injections into 90 nm and 45 nm SRAM-cells. In: Smart Card Research and Advanced Applications – CARDIS 2015, LNCS, vol.9514, pp. 193–205. Springer, Cham (2015).
20. De Mulder, E., Buysschaert, P., Ors, S. B., Delmotte, P., Preneel, B., Vandenbosch, G., Verbauwhede, I.: Electromagnetic analysis attack on an FPGA implementation of an elliptic curve cryptosystem. In: EUROCON 2005-The International Conference on" Computer as a Tool", pp. 1879-1882. IEEE (2005).
21. Azouaoui, M., Papagiannopoulos, K., Zürner, D.: Blind Side-Channel SIFA. In: 2021 Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 555-560. IEEE (2021).
22. Dobraunig, C., Eichlseder, M., Gross, H., Mangard, S., Mendel, F., Primas, R.: Statistical ineffective fault attacks on masked AES with fault countermeasures. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 315-342. Springer, Cham (2018).