Differential Fault Analysis of AES: Toward Reducing Number of Faults

Chong Hee KIM

Information Security Group, ICTEAM institute, Université catholique de Louvain,
Place Sainte Barbe, 2, 1348 Louvain-la-Neuve, Belgium.

Abstract

Differential Fault Analysis (DFA) finds the key of a block cipher using differential information between correct and faulty ciphertexts obtained by inducing faults during the computation of ciphertexts. Among many ciphers AES has been the main target of DFA due to its popularity. DFA of AES has also been diversified into several directions: reducing the required number of faults, applying it to multi-byte fault models, extending to AES-192 and AES-256, or exploiting faults induced at an earlier round.

This article deals with the first three directions together, especially giving weight to reducing the required number of faults. Many previous works show that the required numbers of faults are different although the same fault model is used. This comes from lack of a general method of constructing and solving *differential fault equations*. Therefore we first present how to generate *differential fault equations* systematically and reduce the number of candidates of the key with them, which leads us to find the minimum number of faults. Then we extend to multi-byte fault models and AES-192/256.

Key words: Cryptanalysis, Side channel attacks, Differential fault analysis, Block ciphers, AES

1. Introduction

Differential Fault Analysis (DFA) uses differential information between correct and faulty ciphertexts to retrieve the secret key. Normally an attacker gets faulty ciphertexts by giving external impact on a device with voltage variation, glitch, laser, etc. [4]. The first DFA presented by Biham and Shamir in 1997 [8] targeted DES [1]. Afterward many people tried to break several cryptosystems such as Triple-DES [19], RC4 [7, 20], CLEFIA [13, 37], RSA [27, 40, 15, 31], ElGamal [3], LUC and Demytko [10], ECC [6, 9, 14], AES [11, 12, 16, 18, 33, 28, 23, 39, 36, 38, 5, 29, 21], SMS4 and MacGuffin [24], and DSA [30]. Among them AES has been the main target of DFA due to its popularity.

DFA of AES has also been diversified into several directions: reducing the required number of faults, applying it to multi-byte fault models, extending to AES-192 and AES-256, or exploiting faults induced at an earlier round. The first direction has been most actively researched because more easily the attacker can perform the attack as the required number of faults decreases. Piret and Quisquater showed for the first time that two faults are enough to find the key of AES-128 (AES with 128-bit key) in a one-byte fault model [33]. In 2009, Fukunaga and Takahashi showed that the key of AES-128 could be deduced with one pair of correct and faulty ciphertexts and exhaustive search of 2^{32} candidates [17]. The computational time was 8 - 35 minutes with a Core2 Duo 3.0 GHz PC. Mukhopadhyay also reached the same result [29]. In 2010, Tunstall and Mukhopadhyay demonstrated that exhaustive search could be further reduced to 2^8 with one fault [41].

The extension to AES with 192 and 256-bit key (AES-192 and AES-256 respectively) becomes important in recent years because AES-192 and AES-256 have been deployed more and more. We have to find more subkeys to retrieve the master secret key of AES-192 and AES-256 than AES-128. Hence, generally the required number of faults increases in DFA of AES-192 and AES-256. The work by Piret and Quisquater can be extended to attack AES-192 and AES-256 that need four pairs of correct and faulty ciphertexts. In 2009 Li et al. [25] proposed two DFA on AES-192 and AES-256 based on the Moradi et al.'s DFA on AES-128 [28]. The first method retrieved the key with 16 pairs of correct and faulty ciphertexts and the second method needs 3000 pairs. Barenghi et al. [5] showed that the AES-192 key (and the AES-256 key) could be retrieved with 16 pairs of correct and faulty ciphertexts. These two works require faulty ciphertexts from the *same* plaintext. Takahashi and Fukunaga showed that they could retrieve the AES-192 key using three pairs of correct and faulty ciphertexts and the AES-256 key using two pairs of correct and faulty ciphertexts and two pairs of correct and faulty plaintexts [38]. They reduced the number of faults by analyzing AES key schedule and exhaustive search within practical computational time. However, they need faulty plaintexts (therefore access to a decryption oracle) in DFA of AES-256, which is undesirable. In the extended abstract of this article [21], we showed that the AES-192 key could be found with two pairs of correct and faulty ciphertexts and AES-256 key could be found with three pairs.

Moradi et al. proposed DFA in multi-byte fault models where several bytes are assumed to be corrupted together in 2006 [28]. In an 8-bit architecture one-byte fault model is desirable. However depending on implementation, i.e., 16-bit/32-bit architecture or software implementation, multi-byte fault model can be much useful. In 2009, Saha et al. categorized multi-byte fault models in detail and presented more efficient DFA [35], where two or four pairs of correct and faulty ciphertexts are required.

The last research direction is the utilization of the faults induced at an earlier round. The general countermeasure of DFA is to protect the last few rounds. As redundancy is costly, one should ascertain exactly which rounds need to be protected. Phan and Yen first showed that the faults induced one or two rounds earlier than other DFA could be used to find the key [32]. In DFA of DES Rivain showed that the faults in the middle rounds could be used [34].

We know that the numbers of faults to find the key are different even in the same (*or* similar) fault model in the previous works. Given the pairs of correct and faulty ciphertexts and a fault model, the attacker constructs so called *differential fault equations* (DFE's) that consist of ciphertexts, subkeys, and the characteristics of faulty values. Then she finds subkeys by solving them. However, the methods of constructing and solving DFE's are different for every attack, which leads to different numbers of required faults. Therefore we first present how to generate *differential fault equations* systematically and reduce the number of candidates of the key with them, which leads us to find the minimum number of faults in a given fault model. Then we extend our attacks to AES-192/256 and multi-byte fault models. Our methods have the following advantages compared to the previous works:

- A systematic approach to constructing and solving *differential fault equations* has been proposed. Based on this we can estimate how many wrong candidates of the subkey can be removed before solving the equations, which leads us to find the minimum number of faults in a given fault model.

- We propose a more efficient DFA of AES-128 in a multi-byte fault model where at most twelve bytes are corrupted together. We can find the key with three faults while the previous attack [35] needs four faults.

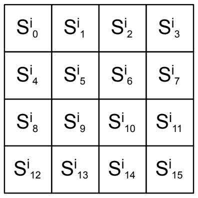
- We propose the first result on DFA of AES-192 and DFA of AES-256 in *multi-byte fault models*. We can find the AES-192 key with two faults when at most eight bytes are corrupted together and three faults when at most twelve bytes are corrupted together. We can find the AES-256 key with three faults when at most eight bytes are corrupted together and four faults when at most twelve bytes are corrupted together.
- In the extended abstract of this article [21], we showed that the AES-192 key could be found with two pairs of correct and faulty ciphertexts with 2^8 exhaustive search and AES-256 key could be found with three pairs with 2^{32} exhaustive search. We presented improved techniques that required no exhaustive search at the workshop [22]. We describe them in detail in this article.

The rest of this article is organized as follows. We briefly describe AES in Section 2. We explain our fault models in Section 3. Then Section 4 introduces how to generate differential fault equations in a byte-fault model. The next section describes differential fault analysis on AES-128, AES-192, and AES-256 using differential fault equations defined in the previous section. Then Section 6 and 7 deal with multi-byte fault models. Finally Section 8 compares proposed attacks with existing ones and concludes the article.

2. AES

AES [2] can encrypt and decrypt 128-bit blocks with 128, 192, or 256 bit-keys. The intermediate computation result of AES, called *state*, is usually represented by a 4×4 matrix, where each cell represents a byte. We denote the input of the i^{th} round by S^i, where $i \in \{1, \ldots, r\}$ and r is the number of rounds. S^0 is the plaintext and S^r is the input to the final round. As shown in Figure 1, S^i_j denotes the $(j+1)^{th}$ byte of the i^{th} state, where $j \in \{0, \ldots 15\}$. AES-128, AES-192, and AES-256 have 10, 12, and 14 rounds respectively. Each round function is composed of 4 transformations except the last round: *SubBytes*, *ShiftRows*, *MixColumns*, and *AddRoundKey*. The last round is lacking *MixColumns*.

- *SubBytes:* It is made up of the application of 16 identical 8×8 Sboxes. This is a non-linear byte substitution. We denote the function of *Sub-*



S^i_0	S^i_1	S^i_2	S^i_3
S^i_4	S^i_5	S^i_6	S^i_7
S^i_8	S^i_9	S^i_{10}	S^i_{11}
S^i_{12}	S^i_{13}	S^i_{14}	S^i_{15}

Figure 1: AES state is represented by a 4 × 4 matrix, where each cell represents a byte.

Bytes by **SB**. That is, $\mathbf{SB}(S^i) = \mathit{SubBytes}(S^i)$. We denote *Inverse SubBytes* by $\mathbf{SB^{-1}}$.

- *ShiftRows:* Each row of the *state* is cyclically shifted over different offsets. Row 0 is not shifted, row 1 is shifted by 1 byte, row 2 is shifted by 2 bytes, and row 3 by 3 bytes. We denote *ShiftRows* and its inverse, *InverseShiftRows*, by **SR** and $\mathbf{SR^{-1}}$ respectively.

- *MixColumns:* This is a linear transformation to each column of the *state*. Each column is considered as polynomial over $\mathbb{F}_{2^8}$ and multiplied modulo x^4+1 with a fixed polynomial $a(x) = 03{*}x^3{+}01{*}x^2{+}01{*}x{+}02$. We denote the function of *MixColumns* by **MC** and its inverse by $\mathbf{MC^{-1}}$.

- *AddRoundKey:* It is a bitwise XOR with a round key.

3. Fault model

We first assume that a byte of an AES state is corrupted by fault injection and the corrupted value is random and unknown to the attacker. The information on which is corrupted among 16 bytes may be known. For example, in [17] Fukunaga and Takahashi showed that they could control the location of a corrupted byte. Although the attacker does not know which byte is corrupted, she can conduct 16 independent and equivalent analysis. Therefore we assume that the attacker knows the location of the corrupted byte. Later we loosen this and assume that several bytes are corrupted. Like in [35], we assume that a random non-zero fault occurs across diagonals.

4. Differential fault equations in a one-byte fault model

As shown in figure 2 a byte fault induced between *MixColumns* of round $r-4$ and $r-3$ spreads to a column at the input of round $r-2$ and to four columns at the input of round $r-1$, where each column is affected by a one-byte error, and to four columns at the input of round r, where each column is affected by four faulty bytes. Depending on the methods of constructing and solving DFE's we have different results [33, 17, 41]. Hence, we propose a systematic method to construct DFE's as follows.

- Select a column of each round input that contains bytes affected by faulty values.
- The number of corrupted bytes in a column should be larger than the number of faulty bytes that affect them.
- Depending on the characteristics of the faulty values affecting bytes in a column, we construct a different type of DFE.

We can reduce the candidates of the key only when the number of corrupted bytes of a column is larger than the number of faulty bytes affecting those bytes (we will see in detail in Section 6). Therefore DFE's constructed from the columns at S^r are useless. To explain types of DFE's, we focus on input and output differences of a nonlinear part, i.e., *SubBytes*. We use the following notations:

X_i	a correct byte input of the Sbox,
X_i^*	a faulty byte input of the Sbox,
$Y_i = \mathit{Sbox}(X_i)$	a byte output of the Sbox for X_i,
$Y_i^* = \mathit{Sbox}(X_i^*)$	a byte output of the Sbox for X_i^*,
$\Delta X_i = X_i \oplus X_i^*$	input difference between X_i and X_i^*,
$\Delta Y_i = Y_i \oplus Y_i^*$	output difference between Y_i and Y_i^*,
$\mathcal{R}(\Delta X_1, \ldots, \Delta X_i)$	relation among $\Delta X_1, \ldots, \Delta X_i$,
$\#(\Delta X_i \mid \Delta Y_i = \beta)$	the number of possible values for ΔX_i when ΔY_i equals β,
$\#(Y_i \mid \Delta X_i = \alpha \wedge \Delta Y_i = \beta)$	the number of possible values for Y_i when ΔX_i equals α and ΔY_i equals β,
K_j^i	the $(j+1)^{th}$ byte of the i^{th} round key,
P_A	the success probability that a candidate satisfies Equation A.

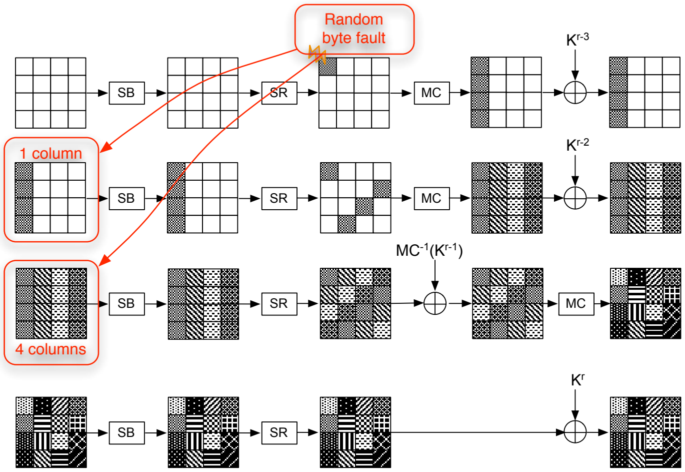


Figure 2: A random byte fault induced between *MixColumns* of round $r-4$ and $r-3$ spreads to one column at the input of round $r-2$ and four columns at the input of round $r-1$.

AES Sbox has 127 possible input differences for one output difference [2]. That is, $\#(\Delta X_i | \Delta Y_i = \beta) = 127$. And $\#(Y_i | \Delta X_i = \alpha \wedge \Delta Y_i = \beta) = 2$ for 126 cases and 4 for the other case.

We can make DFE's with $\Delta X_{i_1}, \ldots, \Delta X_{i_m}$ and $\Delta Y_{i_1}, \ldots, \Delta Y_{i_m}$ to find $K^r_{i_1}, \ldots, K^r_{i_m}$. We can find $\Delta Y_{i_1}, \ldots, \Delta Y_{i_m}$ from correct and faulty ciphertexts and $\mathcal{R}(\Delta X_{i_1}, \ldots, \Delta X_{i_m})$ from the fault model. We denote therefore a *differential fault equation* by:

$$\mathbf{DFE}(K^r_{i_1}, \ldots, K^r_{i_m} | \Delta Y_{i_1}, \ldots, \Delta Y_{i_m}, \mathcal{R}(\Delta X_{i_1}, \ldots, \Delta X_{i_m})),$$

where, $i_1, \ldots, i_m \in \{0, \ldots, 15\}$ and $m \in \{1, \ldots, 16\}$. We simply rewrite it as:

$$\mathbf{DFE}(K^r_{i_1}, \ldots, K^r_{i_m} | S^r_{s_1,\ldots,s_k}, Type),$$

where $S^r_{s_1,\ldots,s_k}$ shows that an equation is constructed with bytes at position $s_1, \ldots, s_k$ of S^r, $k \leq 4$. *MixColumns* propagate a one-byte error to four bytes. Therefore we consider at most four bytes of S^r. *Type* is the type of the equation. We can classify differential fault equations into several types by $\mathcal{R}(\Delta X_{i_1}, \ldots, \Delta X_{i_m})$.

4.1. Type A1

We can construct a type *A differential fault equation* when we know the exact value of the input difference of Sbox. We note that we always know the output difference as we know both correct and faulty ciphertexts. If we know input differences of m bytes, we denote by a type Am *differential fault equation*, where $1 \leq m \leq 4$. Normally we know one byte input difference, we explain type $A1$ *differential fault equation*. We assume that we know input and output differences of one Sbox of round r. That is, we know (α, β) such that $\Delta X_1 = \alpha$ and $\Delta Y_1 = \beta$. Then $\#(Y_1 | \Delta X_1 = \alpha \wedge \Delta Y_1 = \beta) = 2^1$. Y_1 can be represented as a linear combination of keys. That is,

$$Y_1 = f(K^r_{i_1}, \ldots, K^r_{i_m}), \tag{1}$$

where f is a linear function. For example, in the final round of AES, $Y_1 = C_1 \oplus K^r_1$, where C_1 is one byte of the ciphertext C and K^r_1 is the corresponding byte of the last round subkey. Similarly $Y^*_1 = C^*_1 \oplus K^r_1$, where C^*_1 is one byte of the faulty ciphertext C^*. As C_1 and C^*_1 are known, we can find the difference $C_1 \oplus C^*_1 = (Y_1 \oplus K^r_1) \oplus (Y^*_1 \oplus K^r_1) = Y_1 \oplus Y^*_1 = \Delta Y_1 = \beta$. We also know $\Delta X_1 = \alpha$. Hence, we have two candidates for Y_1. Therefore we have two possible values for K^r_1 from $K^r_1 = Y_1 \oplus C_1$. We can make a differential fault equation:

$$\begin{aligned}
& \mathbf{DFE}(K^r_1 | \Delta Y_1, \mathcal{R}(\Delta X_1)) \\
= \quad & \mathbf{DFE}(K^r_1 | \Delta Y_1, \Delta X_1) \\
= \quad & \mathbf{DFE}(K^r_1 | S^r_1, A1) \qquad (2) \\
\Leftrightarrow \quad & \Delta X_1 = \alpha = Sbox^{-1}(Y_1) \oplus Sbox^{-1}(Y^*_1) \\
\Leftrightarrow \quad & \Delta X_1 = \alpha = Sbox^{-1}(K^r_1 \oplus C_1) \oplus Sbox^{-1}(K^r_1 \oplus C^*_1).
\end{aligned}$$

Only two candidates for Y_1 (and therefore for K^r_1) satisfy this equation. We denote this equation by a "*Type A*1" differential fault equation. The success probability that a candidate for K^r_1 satisfies this equation, $P_{Type\ A1}$, is equal to $\frac{2}{256} = 2^{-7}$. If there are q candidates for K^r_1, the number of remaining candidates is $2^{-7} \cdot q$. In the above example, the number of candidates for K^r_1 is 2^8. Therefore the number of remaining candidates for K^r_1 is $2^{-7} \cdot 2^8 = 2$.

4.2. Type B2

We can construct a type B *differential fault equation* when we do not know the input differences of Sboxes but the relation among them. We first

[1] The percentage that it is 2 is 99.21%. With 0.79%, it is 4. In average it is 2.0157.

explain type $B2$ *differential fault equation* and then $B3$ and $B4$ equations in the subsequent sections. We note that we cannot construct a $B1$ equation. We consider two bytes at S^r. We denote the correct output bytes of two S-boxes by Y_1, Y_2 and the faulty output bytes by Y_1^*, Y_2^*. From the assumption we know ΔY_1 and ΔY_2. We also know $\mathcal{R}(\Delta X_1, \Delta X_2) \Leftrightarrow \Delta X_1 = a_1 \Delta X_2$ for a constant a_1. Hence, we can construct a differential fault equation:

$$
\begin{array}{ll}
 & \mathbf{DFE}(K_1^r, K_2^r | \Delta Y_1, \Delta Y_2, \mathcal{R}(\Delta X_1, \Delta X_2)) \\
= & \mathbf{DFE}(K_1^r, K_2^r | \Delta Y_1, \Delta Y_2, \Delta X_1 = a_1 \Delta X_2) \\
= & \mathbf{DFE}(K_1^r, K_2^r | S_{1,2}^r, B2) \qquad (3) \\
\Leftrightarrow & \left\{ \begin{array}{l} \Delta X_1 = Sbox^{-1}(Y_1) \oplus Sbox^{-1}(Y_1^*) \\ \Delta X_2 = Sbox^{-1}(Y_2) \oplus Sbox^{-1}(Y_2^*) \\ \Delta X_1 = a_1 \Delta X_2 \end{array} \right. \\
\Leftrightarrow & \left\{ \begin{array}{l} \Delta X_1 = Sbox^{-1}(K_1^r \oplus C_1) \oplus Sbox^{-1}(K_1^r \oplus C_1^*) \\ \Delta X_2 = Sbox^{-1}(K_2^r \oplus C_2) \oplus Sbox^{-1}(K_2^r \oplus C_2^*) \\ \Delta X_1 = a_1 \Delta X_2 \end{array} \right.
\end{array}
$$

We have that $\#(\Delta X_i | \Delta Y_i = \beta_i) = 127$ for $i = 1, 2$ and $\#((\Delta X_1, \Delta X_2) | (\Delta X_1 = a_1 \Delta X_2) \wedge (\Delta Y_1 = \beta_1) \wedge (\Delta Y_2 = \beta_2)) \simeq 64^2$. For each pair of $(\Delta X_1, \Delta X_2)$, $\#((Y_1, Y_2) | (\Delta X_1 = a_1 \Delta X_2) \wedge (\Delta Y_1 = \beta_1) \wedge (\Delta Y_2 = \beta_2)) = 4$. As we have 64 possible pairs of $(\Delta X_1, \Delta X_2)$, the number of possible candidates for (Y_1, Y_2) (and therefore for (K_1^r, K_2^r)) is $64 \times 4 = 2^8$. We denote this equation by a *Type* $B2$ differential fault equation. The success probability that a candidate for (K_1^r, K_2^r) satisfies this equation, $P_{Type\ B2}$, is equal to $\frac{2^8}{2^{16}} = 2^{-8}$. If there are q candidates for (K_1^r, K_2^r), the number of remaining candidates is $2^{-8} \cdot q$. If $q = 2^{16}$, the number of remaining candidates is $2^{-8} \cdot 2^{16} = 2^8$.

4.3. Type B3 and B4

Type $B3$ and *Type* $B4$ equations are the extension of *Type* $B2$ equation for three and four bytes respectively. We can make a *Type* $B3$ differential

[2] This is the case that we pick up 127 balls twice independently and find the match among 255 different balls. The average expected number is $\frac{127}{255} \cdot \frac{127}{255} \cdot 255 = 63.25$.

fault equation as follows:

$$\begin{array}{ll} & \mathbf{DFE}(K_1^r, K_2^r, K_3^r | \Delta Y_1, \Delta Y_2, \Delta Y_3, \mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3)) \\ = & \mathbf{DFE}(K_1^r, K_2^r, K_3^r | \Delta Y_1, \Delta Y_2, \Delta Y_3, \Delta X_1 = a_1 \Delta X_2 = a_2 \Delta X_3) \\ = & \mathbf{DFE}(K_1^r, K_2^r, K_3^r | S_{1,2,3}^r, B3) \qquad (4) \\ \Leftrightarrow & \begin{cases} \Delta X_1 = Sbox^{-1}(Y_1) \oplus Sbox^{-1}(Y_1^*) \\ \Delta X_2 = Sbox^{-1}(Y_2) \oplus Sbox^{-1}(Y_2^*) \\ \Delta X_3 = Sbox^{-1}(Y_3) \oplus Sbox^{-1}(Y_3^*) \\ \Delta X_1 = a_1 \Delta X_2 = a_2 \Delta X_3 \end{cases} \\ \Leftrightarrow & \begin{cases} \Delta X_1 = Sbox^{-1}(K_1^r \oplus C_1) \oplus Sbox^{-1}(K_1^r \oplus C_1^*) \\ \Delta X_2 = Sbox^{-1}(K_2^r \oplus C_2) \oplus Sbox^{-1}(K_2^r \oplus C_2^*) \\ \Delta X_3 = Sbox^{-1}(K_3^r \oplus C_3) \oplus Sbox^{-1}(K_3^r \oplus C_3^*) \\ \Delta X_1 = a_1 \Delta X_2 = a_2 \Delta X_3 \end{cases} \end{array}$$

We have that $\#((\Delta X_1, \Delta X_2, \Delta X_3) | (\Delta X_1 = a_1 \Delta X_2 = a_2 \Delta X_3) \wedge (\Delta Y_1 = \beta_1) \wedge (\Delta Y_2 = \beta_2) \wedge (\Delta Y_3 = \beta_3)) \simeq 32$ and (Y_1, Y_2, Y_3) is $\#((Y_1, Y_2, Y_3) | (\Delta X_1 = a_1 \Delta X_2 = a_2 \Delta X_3 = x_1) \wedge (\Delta Y_1 = y_1) \wedge (\Delta Y_2 = y_2) \wedge (\Delta Y_3 = y_3)) = 8$. Therefore the number of possible candidates for (Y_1, Y_2, Y_3) $(or\ K_1^r, K_2^r, K_3^r\ )$ is $32 \times 8 = 2^8$. So we have $P_{Type\ B3} = \frac{2^8}{2^{24}} = 2^{-16}$. Similarly we can construct a *Type* $B4$ differential fault equation and have $P_{Type\ B4} = \frac{2^8}{2^{32}} = 2^{-24}$.

Note. Until now we have considered differential fault equations at S^r. Hence, $Y_i = C_i \oplus K_i^r$. If we know Y_i, we can find K_i^r. However Y_i can be a linear combination of several K_i^r's if we make an equation at other rounds. That is, we may have $Y_i = \oplus_{j=1}^{k} [a_j (C_j \oplus K_j^r)]$, where a_j is a constant.

5. Differential fault analysis in a one-byte fault model

As we have $P_{Type\ B4} < P_{Type\ B3} < P_{Type\ B2} < P_{Type\ A1}$, it is better to construct as many $B4$ equations as possible. In a one-byte fault model we can construct at most five $B4$ equations. However, sometimes we need to transform $B4$ equations into $A1, B1, B2$, or $B3$ equations to reduce the time complexity. We show how to construct and solve DFE's in AES-128, AES-192, and AES-256. Furthermore we show how to estimate the number of wrong candidates that can be removed with DFE's, which leads us to find the minimal number of faults in a given fault model.

5.1. DFA on AES-128

If a fault is induced between *MixColumns* of round $r-3$ and $r-2$, we can make four $B4$ differential fault equations at S^r and one $B4$ differential fault equation at S^{r-1}. Without loss of generality, we assume that the first byte of the state is corrupted by the fault. Then we have

$$\begin{aligned}
&\mathbf{DFE}_1(K_0^r, K_7^r, K_{10}^r, K_{13}^r | S_{0,4,8,12}^r, B4),\\
&\mathbf{DFE}_2(K_1^r, K_4^r, K_{11}^r, K_{14}^r | S_{1,5,9,13}^r, B4),\\
&\mathbf{DFE}_3(K_2^r, K_5^r, K_8^r, K_{15}^r | S_{2,6,10,14}^r, B4),\\
&\mathbf{DFE}_4(K_3^r, K_6^r, K_9^r, K_{12}^r | S_{3,7,11,15}^r, B4),\\
&\mathbf{DFE}_5(K_0^{r-1}, \ldots, K_{15}^{r-1} | S_{0,4,8,12}^{r-1}, B4).
\end{aligned} \tag{5}$$

As K^{r-1} can be computed from K^r by AES key schedule, the size of the key space for the above equations is 2^{128}. The number of candidates satisfying the equations is $2^{128} \cdot (P_{Type\ B4})^5 = 2^{128} \cdot (2^{-24})^5 = 2^8$. Therefore one fault is enough to find the key. This explains why the authors in [41] found the key with one fault and exhaustive search of 2^8 candidates.

The time complexity to solve $\mathbf{DFE}_1$, $\mathbf{DFE}_2$, $\mathbf{DFE}_3$, or $\mathbf{DFE}_4$ is 2^{16} instead of 2^{32}, since we can find candidates for the first two bytes and then extend the list of candidates by one byte [33]. The complexity to solve $\mathbf{DFE}_5$ is 2^{32} since only 2^{32} candidates for $K_0^{r-1}, \ldots, K_{15}^{r-1}$ satisfying $\mathbf{DFE}_1$, $\mathbf{DFE}_2$, $\mathbf{DFE}_3$, and $\mathbf{DFE}_4$ simultaneously remain [41].

5.2. DFA on AES-192

To find the secret key of AES-192 we need K^r and the right half of K^{r-1}. The left half of K^{r-1} can be computed from K^r. Therefore the total key space for K^{r-1} and K^r is $2^{64} \cdot 2^{128} = 2^{192}$. With one pair of correct and faulty ciphertexts we can make five $B4$ differential fault equations. The number of candidates satisfying these equations is $2^{192} \cdot (P_{Type\ B4})^5 = 2^{192} \cdot (2^{-24})^5 = 2^{72}$. It means that we cannot find the secret key with one fault. With one fault we can remove 2^{120} wrong candidates. With two pairs of correct and faulty ciphertexts we can construct ten $B4$ equations and therefore remove 2^{240} wrong candidates. Therefore two faults are enough to find AES-192 secret key. However we have to be careful in exploiting these equations. We show two attacks that exploit the equations differently and therefore give different results.

5.2.1. Attack 1

We assume that we have two pairs of correct and faulty ciphertexts by inducing a fault between *MixColumns* of round $r-3$ and $r-2$. Then, we can make eight $B4$ differential fault equations at S^r and two $B4$ differential fault equations at S^{r-1}. That is, we have:

$$\begin{aligned}
&\mathbf{DFE}_1^{1st\,pair}(K_0^r, K_7^r, K_{10}^r, K_{13}^r | S_{0,4,8,12}^r, B4),\\
&\mathbf{DFE}_2^{1st\,pair}(K_1^r, K_4^r, K_{11}^r, K_{14}^r | S_{1,5,9,13}^r, B4),\\
&\mathbf{DFE}_3^{1st\,pair}(K_2^r, K_5^r, K_8^r, K_{15}^r | S_{2,6,10,14}^r, B4),\\
&\mathbf{DFE}_4^{1st\,pair}(K_3^r, K_6^r, K_9^r, K_{12}^r | S_{3,7,11,15}^r, B4),\\
&\mathbf{DFE}_5^{1st\,pair}(K_0^{r-1}, \ldots, K_{15}^{r-1} | S_{0,4,8,12}^{r-1}, B4),\\
&\mathbf{DFE}_6^{2nd\,pair}(K_0^r, K_7^r, K_{10}^r, K_{13}^r | S_{0,4,8,12}^r, B4),\\
&\mathbf{DFE}_7^{2nd\,pair}(K_1^r, K_4^r, K_{11}^r, K_{14}^r | S_{1,5,9,13}^r, B4),\\
&\mathbf{DFE}_8^{2nd\,pair}(K_2^r, K_5^r, K_8^r, K_{15}^r | S_{2,6,10,14}^r, B4),\\
&\mathbf{DFE}_9^{2nd\,pair}(K_3^r, K_6^r, K_9^r, K_{12}^r | S_{3,7,11,15}^r, B4),\\
&\mathbf{DFE}_{10}^{2nd\,pair}(K_0^{r-1}, \ldots, K_{15}^{r-1} | S_{0,4,8,12}^{r-1}, B4),
\end{aligned} \tag{6}$$

where, $\mathbf{DFE}_1^{1st\,pair}, \ldots, \mathbf{DFE}_5^{1st\,pair}$ are from the first pair and $\mathbf{DFE}_6^{2nd\,pair}$ $\ldots \mathbf{DFE}_{10}^{2nd\,pair}$ are from the second pair. With $\mathbf{DFE}_1^{1st\,pair}$ and $\mathbf{DFE}_6^{2nd\,pair}$ we can find $(K_0^r, K_7^r, K_{10}^r, K_{13}^r)$ as the number of remaining wrong candidates is $2^{32} \cdot (P_{Type\ B4})^2 = 2^{32} \cdot 2^{-48} \simeq 0$. With $\mathbf{DFE}_2^{1st\,pair}$ and $\mathbf{DFE}_7^{2nd\,pair}$ we can find $(K_1^r, K_4^r, K_{11}^r, K_{14}^r)$. Similarly all bytes of K^r can be found.

Now we have two equations, $\mathbf{DFE}_5^{1st\,pair}$ and $\mathbf{DFE}_{10}^{2nd\,pair}$, that include K^{r-1}. As we know the left half of K^{r-1}, we transform these two $B4$ equations into the following equations[3] that include only the right half of K^{r-1}:

$$\begin{aligned}
&\mathbf{DFE}_{11}^{1st\,pair}(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^r | S_8^{r-1}, A1),\\
&\mathbf{DFE}_{12}^{1st\,pair}(K_3^{r-1}, K_7^{r-1}, K_{11}^{r-1}, K_{15}^r | S_4^{r-1}, A1),\\
&\mathbf{DFE}_{13}^{2nd\,pair}(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^r | S_8^{r-1}, A1),\\
&\mathbf{DFE}_{14}^{2nd\,pair}(K_3^{r-1}, K_7^{r-1}, K_{11}^{r-1}, K_{15}^r | S_4^{r-1}, A1).
\end{aligned}$$

[3] We note that with K^r and the left half of K^{r-1} we can find the input difference of one faulty byte of S^{r-1} [21]. Therefore we can construct $A1$ equations that have only the right half of K^{r-1}.

Algorithm 1: DFA on AES-192 with 2 faults: attack 1

```
Input: Two pairs of correct and faulty ciphertexts by inducing a fault
       between *MixColumns* of round $r-3$ and $r-2$.
Output: $2^{36}$ candidates for AES-192 secret key.
begin
    Construct $\mathbf{DFE}_1^{\mathit{1st\ pair}}(K_0^r, K_7^r, K_{10}^r, K_{13}^r | S_{0,4,8,12}^r, B4)$ and
    $\mathbf{DFE}_6^{\mathit{2nd\ pair}}(K_0^r, K_7^r, K_{10}^r, K_{13}^r | S_{0,4,8,12}^r, B4)$ and find
    $(K_0^r, K_7^r, K_{10}^r, K_{13}^r)$.
    For the other three columns of $K^r$ construct similar **DFE**'s and
    solve them. (Finally we have $K^r$.)
    Compute the left half of $K^{r-1}$ from $K^r$ by AES key schedule.
    Find the input difference of one faulty byte of $S^{r-1}$.
    Construct $\mathbf{DFE}_{11}^{\mathit{1st\ pair}}(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^r | S_8^{r-1}, A1)$ and
    $\mathbf{DFE}_{13}^{\mathit{2nd\ pair}}(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^r | S_8^{r-1}, A1)$ and find $2^{18}$
    candidates for $(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^r)$.
    Similarly find $2^{18}$ candidates for $(K_3^{r-1}, K_7^{r-1}, K_{11}^{r-1}, K_{15}^r)$. (Finally
    we have one candidate for $K^r$ and $2^{36}$ candidates for right half of
    $K^{r-1}$.)
    Output $2^{36}$ candidates for AES-192 secret key.
end
```

With $\mathbf{DFE}_{11}^{1st\ pair}$ and $\mathbf{DFE}_{13}^{2nd\ pair}$ we can reduce the number of candidates for $(K_2^{r-1}, K_6^{r-1},\ K_{10}^{r-1},\ K_{14}^r)$ to $2^{32} \cdot (P_{Type\ A1})^2 = 2^{32} \cdot 2^{-14} = 2^{18}$. With $\mathbf{DFE}_{12}^{1st\ pair}$ and $\mathbf{DFE}_{14}^{2nd\ pair}$ we can reduce the number of candidates for $(K_3^{r-1}, K_7^{r-1}, K_{11}^{r-1}, K_{15}^r)$ to 2^{18}. Finally we have 2^{36} candidates for AES-192 secret key (see Algorithm 1).

5.2.2. Attack 2

In the previous attack eight $B4$ equations are used to find K^r. They could have reduced $(2^{24})^8 = 2^{192}$ candidates but were used to reduce 2^{128} candidates for K^r. Therefore Attack 1 is inefficient. By changing the location of a fault we can further increase the efficiency of the attack. Now we assume that one pair is obtained by inducing a fault between *MixColumns* of round $r-3$ and $r-2$ and the other is obtained by inducing a fault one round earlier. Then we can construct four $B4$ differential fault equations at S^r and one $B4$ differential fault equation at S^{r-1} with the first pair as follows:

$$\begin{aligned}
&\mathbf{DFE}_1^{1st\ pair}(K_0^r, K_7^r, K_{10}^r, K_{13}^r | S_{0,4,8,12}^r, B4)\\
&\mathbf{DFE}_2^{1st\ pair}(K_1^r, K_4^r, K_{11}^r, K_{14}^r | S_{1,4,11,14}^r, B4),\\
&\mathbf{DFE}_3^{1st\ pair}(K_2^r, K_5^r, K_8^r, K_{15}^r | S_{2,5,8,15}^r, B4),\\
&\mathbf{DFE}_4^{1st\ pair}(K_3^r, K_6^r, K_9^r, K_{12}^r | S_{3,6,9,12}^r, B4),\\
&\mathbf{DFE}_5^{1st\ pair}(K_0^{r-1}, \ldots, K_{15}^{r-1} | S_{0,4,8,12}^{r-1}, B4).
\end{aligned} \tag{7}$$

With the second pair we have four $B4$ differential fault equations at S^{r-1} and one $B4$ differential fault equation at S^{r-2} as follows:

$$\begin{aligned}
&\mathbf{DFE}_6^{2nd\ pair}(K_0^{r-1}, \ldots, K_{15}^{r-1} | S_{0,4,8,12}^{r-1}, B4),\\
&\mathbf{DFE}_7^{2nd\ pair}(K_0^{r-1}, \ldots, K_{15}^{r-1} | S_{1,5,9,13}^{r-1}, B4),\\
&\mathbf{DFE}_8^{2nd\ pair}(K_0^{r-1}, \ldots, K_{15}^{r-1} | S_{2,6,10,14}^{r-1}, B4),\\
&\mathbf{DFE}_9^{2nd\ pair}(K_0^{r-1}, \ldots, K_{15}^{r-1} | S_{3,7,11,15}^{r-1}, B4),\\
&\mathbf{DFE}_{10}^{2nd\ pair}(K_0^{r-2}, \ldots, K_{15}^{r-2} | S_{0,4,8,12}^{r-2}, B4).
\end{aligned} \tag{8}$$

With $\mathbf{DFE}_1^{1st\ pair}$ we reduce the number of candidates for $(K_0^r, K_7^r, K_{10}^r, K_{13}^r)$ from 2^{32} to 2^8. Similar analysis can be done for $\mathbf{DFE}_2^{1st\ pair}$, $\mathbf{DFE}_3^{1st\ pair}$, and $\mathbf{DFE}_4^{1st\ pair}$. Therefore we have 2^{32} candidates for K^r. As the left half of K^{r-1} can be computed from K^r by AES key schedule, we transform

$\mathbf{DFE}_{5}^{1st\ pair}$, $\mathbf{DFE}_{6}^{2nd\ pair}$, $\mathbf{DFE}_{7}^{2nd\ pair}$, $\mathbf{DFE}_{8}^{2nd\ pair}$, and $\mathbf{DFE}_{9}^{2nd\ pair}$ into the following equations that have only the left half of K^{r-1}:

$$\begin{array}{l}
\mathbf{DFE}_{11}^{2nd\ pair}(K_0^{r-1}, K_1^{r-1}, K_4^{r-1}, K_5^{r-1}, K_8^{r-1}, K_9^{r-1}, K_{12}^{r-1}, K_{13}^{r-1} | S_{0,12}^{r-1}, B2),\\
\mathbf{DFE}_{12}^{2nd\ pair}(K_0^{r-1}, K_1^{r-1}, K_4^{r-1}, K_5^{r-1}, K_8^{r-1}, K_9^{r-1}, K_{12}^{r-1}, K_{13}^{r-1} | S_{0,12}^{r-1}, B2),\\
\mathbf{DFE}_{13}^{2nd\ pair}(K_0^{r-1}, K_1^{r-1}, K_4^{r-1}, K_5^{r-1}, K_8^{r-1}, K_9^{r-1}, K_{12}^{r-1}, K_{13}^{r-1} | S_{1,5}^{r-1}, B2),\\
\mathbf{DFE}_{14}^{2nd\ pair}(K_0^{r-1}, K_1^{r-1}, K_4^{r-1}, K_5^{r-1}, K_8^{r-1}, K_9^{r-1}, K_{12}^{r-1}, K_{13}^{r-1} | S_{6,10}^{r-1}, B2),\\
\mathbf{DFE}_{15}^{2nd\ pair}(K_0^{r-1}, K_1^{r-1}, K_4^{r-1}, K_5^{r-1}, K_8^{r-1}, K_9^{r-1}, K_{12}^{r-1}, K_{13}^{r-1} | S_{11,15}^{r-1}, B2).
\end{array}$$

With these five $B2$ equations we reduce the number of wrong candidates for the left half of K^{r-1} to $2^{32} \cdot (P_{Type\ B2})^5 = 2^{32} \cdot (2^{-8})^5 \simeq 0$. Hence, we have one correct value for the left half of K^{r-1} (and K^r). To find the right half of K^{r-1} we construct the following equations from $\mathbf{DFE}_{5}^{1st\ pair}$, $\mathbf{DFE}_{6}^{2nd\ pair}$, $\mathbf{DFE}_{7}^{2nd\ pair}$, $\mathbf{DFE}_{8}^{2nd\ pair}$, and $\mathbf{DFE}_{9}^{2nd\ pair}$:

$$\begin{array}{l}
\mathbf{DFE}_{16}^{1st\ pair}(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^{r} | S_8^{r-1}, A1),\\
\mathbf{DFE}_{17}^{1st\ pair}(K_3^{r-1}, K_7^{r-1}, K_{11}^{r-1}, K_{15}^{r} | S_4^{r-1}, A1),\\
\mathbf{DFE}_{18}^{2nd\ pair}(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^{r} | S_8^{r-1}, A1),\\
\mathbf{DFE}_{19}^{2nd\ pair}(K_3^{r-1}, K_7^{r-1}, K_{11}^{r-1}, K_{15}^{r} | S_4^{r-1}, A1),
\end{array}$$

$$\begin{array}{l}
\mathbf{DFE}_{20}^{2nd\ pair}(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^{r} | S_{13}^{r-1}, A1),\\
\mathbf{DFE}_{21}^{2nd\ pair}(K_3^{r-1}, K_7^{r-1}, K_{11}^{r-1}, K_{15}^{r} | S_9^{r-1}, A1),\\
\mathbf{DFE}_{22}^{2nd\ pair}(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^{r} | S_2^{r-1}, A1),\\
\mathbf{DFE}_{23}^{2nd\ pair}(K_3^{r-1}, K_7^{r-1}, K_{11}^{r-1}, K_{15}^{r} | S_{14}^{r-1}, A1),\\
\mathbf{DFE}_{24}^{2nd\ pair}(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^{r} | S_7^{r-1}, A1),\\
\mathbf{DFE}_{25}^{2nd\ pair}(K_3^{r-1}, K_7^{r-1}, K_{11}^{r-1}, K_{15}^{r} | S_3^{r-1}, A1).
\end{array}$$

We reduce the number of wrong candidates for $(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^{r})$ to $2^{32} \cdot (P_{Type\ A1})^5 = 2^{32} \cdot (2^{-7})^5 \simeq 0$. Therefore we have one correct value for $(K_2^{r-1}, K_6^{r-1}, K_{10}^{r-1}, K_{14}^{r})$. Similarly we find $K_3^{r-1}, K_7^{r-1}, K_{11}^{r-1}, K_{15}^{r-1}$. Finally we can find the AES-192 secret key with AES key schedule as we know K^r and the right half of K^{r-1}. We summarized the procedure in Algorithm 2.

Algorithm 2: DFA on AES-192 with 2 faults: attack 2

```
Input: Two pairs of correct and faulty ciphertexts. The 1st pair is
       obtained by inducing a fault between MixColumns of round
       $r-3$ and $r-2$ and the 2nd pair is obtained by inducing a
       fault one round earlier.
Output: AES-192 secret key.
begin
    Construct $\mathbf{DFE}_1^{1st\,pair}(K_0^r, K_7^r, K_{10}^r, K_{13}^r | S_{0,4,8,12}^r, B4)$ and find $2^8$
    candidates for $(K_0^r, K_7^r, K_{10}^r, K_{13}^r)$.
    For the other three columns of $K^r$ construct DFE's and find $2^8$
    candidates. (Finally we have $2^{32}$ candidates for $K^r$.)
    Construct $\mathbf{DFE}_{11}^{1st\,pair}$ $\mathbf{DFE}_{12}^{2nd\,pair}$, $\mathbf{DFE}_{13}^{2nd\,pair}$, $\mathbf{DFE}_{14}^{2nd\,pair}$, and
    $\mathbf{DFE}_{15}^{2nd\,pair}$.
    for $2^{32}$ candidates of $K^r$ do
        Compute the left half of $K^{r-1}$ from $K^r$ by AES key schedule.
        if (a candidate satisfies $\boldsymbol{DFE}_{11}^{1st\,pair}$, $\boldsymbol{DFE}_{12}^{2nd\,pair}$, $\boldsymbol{DFE}_{13}^{2nd\,pair}$,
        $\boldsymbol{DFE}_{14}^{2nd\,pair}$, and $\boldsymbol{DFE}_{15}^{2nd\,pair}$) then
            Stop the loop and the candidate is $K^r$.
        end
    end
    Construct five $A1$ equations for the $3^{rd}$ column of $K^{r-1}$ and find it.
    Construct five $A1$ equations for the $4^{th}$ column of $K^{r-1}$ and find it.
    Find the AES-192 secret key by AES key schedule with $K^r$ and
    the right half of $K^{r-1}$.
    Output AES-192 secret key.
end
```

5.3. DFA on AES-256

To find the secret key of AES-256 we need K^r and K^{r-1}. With one pair of correct and faulty ciphertexts we can make five $B4$ equations and remove 2^{120} wrong candidates. With two pairs we can make ten $B4$ equations and remove 2^{240} wrong candidates. Theoretically with two pairs we can reduce the key space for AES-256 from 2^{256} to 2^{16}. We describe two methods to find the AES-256 key.

5.3.1. Attack with two faults

Our attack is similar to that of Li et al. in [26] to find the AES-256 key with two pairs of correct and faulty ciphertexts. However, our attack uses different DFE's systematically made according to Section 4 and the fact that *MixColumns* and *AddRoundKey* can be switched, i.e., $\mathbf{MC}(A) \oplus K = \mathbf{MC}(A \oplus \mathbf{MC^{-1}}(K))$. We assume that one pair of correct and faulty ciphertexts is obtained by inducing a fault between *MixColumns* of round $r-3$ and $r-2$ and the other is obtained by inducing a fault one round earlier. Then we have the following equations:

$$\begin{aligned}
&\mathbf{DFE}_1^{1st\ pair}(K_0^r, K_7^r, K_{10}^r, K_{13}^r | S_{0,4,8,12}^r, B4),\\
&\mathbf{DFE}_2^{1st\ pair}(K_1^r, K_4^r, K_{11}^r, K_{14}^r | S_{1,5,9,13}^r, B4),\\
&\mathbf{DFE}_3^{1st\ pair}(K_2^r, K_5^r, K_8^r, K_{15}^r | S_{2,6,10,14}^r, B4),\\
&\mathbf{DFE}_4^{1st\ pair}(K_3^r, K_6^r, K_9^r, K_{12}^r | S_{3,7,11,15}^r, B4),\\
&\mathbf{DFE}_5^{1st\ pair}(K_0^{r-1}, \ldots, K_{15}^{r-1} | S_{0,4,8,12}^{r-1}, B4),
\end{aligned} \tag{9}$$

and

$$\begin{aligned}
&\mathbf{DFE}_6^{2nd\ pair}(\widehat{K}_0^{r-1}, \widehat{K}_7^{r-1}, \widehat{K}_{10}^{r-1}, \widehat{K}_{13}^{r-1} | S_{0,4,8,12}^{r-1}, B4),\\
&\mathbf{DFE}_7^{2nd\ pair}(\widehat{K}_1^{r-1}, \widehat{K}_4^{r-1}, \widehat{K}_{11}^{r-1}, \widehat{K}_{14}^{r-1} | S_{1,5,9,13}^{r-1}, B4),\\
&\mathbf{DFE}_8^{2nd\ pair}(\widehat{K}_2^{r-1}, \widehat{K}_5^{r-1}, \widehat{K}_8^{r-1}, \widehat{K}_{15}^{r-1} | S_{2,6,10,14}^{r-1}, B4),\\
&\mathbf{DFE}_9^{2nd\ pair}(\widehat{K}_3^{r-1}, \widehat{K}_6^{r-1}, \widehat{K}_9^{r-1}, \widehat{K}_{12}^{r-1} | S_{3,7,11,15}^{r-1}, B4),\\
&\mathbf{DFE}_{10}^{2nd\ pair}(K_0^{r-2}, \ldots, K_{15}^{r-2} | S_{0,4,8,12}^{r-2}, B4),
\end{aligned} \tag{10}$$

where, $\widehat{K}_i^{r-1}$ is $\mathbf{MC^{-1}}(K_i^{r-1})$. With $\mathbf{DFE}_1^{1st\ pair}$ we reduce the number of candidates for $(K_0^r, K_7^r, K_{10}^r, K_{13}^r)$ from 2^{32} to 2^8. Similar analysis can be done for $\mathbf{DFE}_2^{1st\ pair}$, $\mathbf{DFE}_3^{1st\ pair}$, and $\mathbf{DFE}_4^{1st\ pair}$. Therefore we have 2^{32} candidates for K^r. For each candidate for K^r we solve $\mathbf{DFE}_6^{2nd\ pair}$, $\mathbf{DFE}_7^{2nd\ pair}$,

$\mathbf{DFE}_8^{2nd\,pair}$, and $\mathbf{DFE}_9^{2nd\,pair}$. Hence, we have 2^{32} candidates for $\widehat{K}^{r-1}$ (also for K^{r-1}) for each candidate for K^r. We consider the rightmost three columns of K^{r-2} that can be computed from K^r by AES key schedule. We construct a differential fault equation: $\mathbf{DFE}_{11}^{2nd\,pair}$ $(K_1^{r-1}, K_2^{r-1}, K_3^{r-1}, K_4^{r-1}, K_5^{r-1}, K_6^{r-1}$, K_8^{r-1}, K_9^{r-1}, K_{11}^{r-1}, K_{12}^{r-1}, $K_{14}^{r-1}, K_{15}^{r-1}|S_{4,8,12}^{r-2}, B3)$ for three bytes at S^{r-2}. By solving the equation, we have 2^8 candidates for 12 bytes of K^{r-1}. Finally we have 2^{16} candidates for K^{r-1} (2^8 for 12 bytes and 2^8 for remaining 4 bytes). Therefore we have 2^{48} candidates for (K^{r-1}, K^r). By solving $\mathbf{DFE}_5^{1st\,pair}$, we reduce the number of candidates for (K^{r-1}, K^r) from 2^{48} to 2^{24}. Finally for two bytes at S^{r-2} we construct a differential fault equation: $\mathbf{DFE}_{12}^{2nd\,pair}$ $(K_0^{r-2}, \ldots, K_{15}^{r-2}|S_{0,4}^{r-2}, B2)$ and reduce the number of the candidates to 2^{16} (See Algorithm 3).

5.3.2. Attack with three faults

The overall complexity of the previous attack with two pairs of correct and faulty ciphertexts is 2^{48}, which is too high to be implemented. Therefore we try to find a practical solution with one more pair of correct and faulty ciphertexts. With three pairs we can construct fifteen $B4$ equations and remove $(2^{120})^3 = 2^{360}$ wrong candidates. Therefore three faults are enough to find 256 bits of the secret key of AES-256. We assume that we can obtain two pairs of correct and faulty ciphertexts by inducing a fault between *MixColumns* of round $r-3$ and $r-2$ and a pair by inducing a fault one round earlier. With the first and the second pairs we construct eight $B4$ equations at S^r and therefore find K^r. With the third pair we construct four $B4$ equations at S^{r-1} and therefore reduce the number of candidates for K^{r-1} from 2^{128} to 2^{32}. Finally we find K^{r-1} with two $B4$ equations at S^{r-1} from the first and the second pairs.

6. Differential fault equations in a multi-byte fault model

Up to now we assume that one byte is disturbed by fault injection. We loosen this and assume that several bytes are corrupted together. Following the models in [35], we assume four models where a non-zero fault is injected at the input of the i^{th} round:

- Model 0 ($M0$): A random non-zero fault is induced in one of the diag-

Algorithm 3: DFA on AES-256 with 2 faults

```
Input: Two pairs of correct and faulty ciphertexts. The 1st pair is obtained by inducing a fault between MixColumns of round $r-3$ and $r-2$ and the 2nd pair is obtained by inducing a fault one round earlier.
Output: $2^{16}$ candidates for AES-256 secret key.
begin
    Construct $\mathbf{DFE}_1^{\mathit{1st\ pair}}$ and find $2^8$ candidates for $(K_0^r, K_7^r, K_{10}^r, K_{13}^r)$.
    For each of the other three columns construct a DFE and find $2^8$ candidates.
    Construct $\mathbf{DFE}_6^{\mathit{2nd\ pair}}, \ldots, \mathbf{DFE}_9^{\mathit{2nd\ pair}}$.
    for $2^{32}$ candidates for $K^r$ do
        Find $2^8$ candidates for $(\widehat{K}_0^{r-1}, \widehat{K}_7^{r-1}, \widehat{K}_{10}^{r-1}, \widehat{K}_{13}^{r-1})$ by solving $\mathbf{DFE}_6^{\mathit{2nd\ pair}}$.
        For each of the other three columns find $2^8$ candidates.
        Find $K^{r-1} = \mathbf{MC}(\widehat{K}^{r-1})$.
        Solve $\mathbf{DFE}_{11}^{\mathit{2nd\ pair}}$ and reduce the number of candidates for $K^{r-1}$ from $2^{32}$ to $2^{16}$.
    end
    for $2^{48}$ candidates of $(K^{r-1}, K^r)$ do
        Solve $\mathbf{DFE}_5^{\mathit{1st\ pair}}$ and $\mathbf{DFE}_{12}^{\mathit{2nd\ pair}}$.
        if (a candidate satisfies the equations) then
            Put it as a valid candidate.
        end
    end
    Find the AES-256 secret key by the AES key schedule.
    Return $2^{16}$ candidates for the AES-256 secret key.
end
```

onals, D_0, D_1, D_2 and D_3[4],

- Model 1 ($M1$): A random non-zero fault occurs across two diagonals,
- Model 2 ($M2$): A random non-zero fault occurs across three diagonals,
- Model 3 ($M3$): A random non-zero fault occurs across four diagonals,

where a *diagonal* is a set of four bytes of the state. We have the following four diagonals:

$$\begin{aligned} D_0 &= \{S_0, S_5, S_{10}, S_{15}\}, \\ D_1 &= \{S_1, S_6, S_{11}, S_{12}\}, \\ D_2 &= \{S_2, S_7, S_8, S_{13}\}, \\ D_3 &= \{S_3, S_4, S_9, S_{14}\}. \end{aligned}$$

6.1. Differential fault equations in $M1$

The differential fault equations in $M0$ are the same as those in Section 4. Hence, we start with $M1$. We do not know the input differences of Sboxes but the relation among them. We denote new differential fault equations for two, three, and four bytes in $M1$ by *Type* $C2, C3$, and $C4$ equations respectively to distinguish them from $B2, B3$, and $B4$ equations. At most two bytes are corrupted in each column of the input of the i^{th} *MixColumns*. We denote errors in two bytes by α_1 and α_2. Then we have $\mathcal{R}(\Delta X_1, \Delta X_2) \Leftrightarrow (\Delta X_1 = a_{1,1}\alpha_1 + a_{1,2}\alpha_2) \wedge (\Delta X_2 = a_{2,1}\alpha_1 + a_{2,2}\alpha_2)$ in a $C2$ equation, where $a_{i,j}$ is a constant. We have $\#(\Delta X_i | \Delta Y_i = \beta_i) = 127$ for $i = 1, 2$. Furthermore $\#((\Delta X_1, \Delta X_2) | \mathcal{R}(\Delta X_1, \Delta X_2) \wedge (\Delta Y_1 = \beta_1) \wedge (\Delta Y_2 = \beta_2)) = 127 \cdot 127 \simeq 2^{14}$ since both ΔX_1 and ΔX_2 can take any value. For each pair of $(\Delta X_1, \Delta X_2)$, the number of possible candidates for (Y_1, Y_2) is 4. As we have 2^{14} possible pairs of $(\Delta X_1, \Delta X_2)$, the number of possible candidates for (Y_1, Y_2) (and therefore for (K_1^r, K_2^r)) is $2^{14} \times 4 = 2^{16}$. The success probability that a candidate for (K_1^r, K_2^r) satisfies the equation, $P_{Type\,C2}$, is equal to $\frac{2^{16}}{2^{16}} = 1$. Therefore we cannot remove any wrong candidate.

In a $C3$ equation, we have $\mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3) \Leftrightarrow (\Delta X_1 = a_{1,1}\alpha_1 + a_{1,2}\alpha_2) \wedge (\Delta X_2 = a_{2,1}\alpha_1 + a_{2,2}\alpha_2) \wedge (\Delta X_3 = a_{3,1}\alpha_1 + a_{3,2}\alpha_2)$, where $a_{i,j}$ is a

[4] This is the same fault model described in Section 4. We consider a one-byte fault before *MixColumns* while models in [35] consider faults after *MixColumns*.

constant. We can rewrite $\mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3)$ as:

$$\mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3) \Leftrightarrow a\Delta X_1 + b\Delta X_2 = c\Delta X_3,$$

where a, b, and c are constants. Then $\#((\Delta X_1, \Delta X_2, \Delta X_3)|\mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3) \wedge (\Delta Y_1 = \beta_1) \wedge (\Delta Y_2 = \beta_2) \wedge (\Delta Y_3 = \beta_3)) = 127 \cdot 2^6 \simeq 2^{13}$. Because ΔX_3 can take any value but both ΔX_1 and ΔX_2 should satisfy $a\Delta X_1 + b\Delta X_2 = c\Delta X_3$. The possible values for such $(\Delta X_1, \Delta X_2, \Delta X_3)$ is $255 \cdot \frac{127}{255} \cdot \frac{127}{255} \simeq 2^6$. For each pair of $(\Delta X_1, \Delta X_2, \Delta X_3)$, the number of possible candidates for (Y_1, Y_2, Y_3) is 8. Hence, the number of possible candidates for (Y_1, Y_2, Y_3) is $2^{13} \times 8 = 2^{16}$. The success probability that a candidate satisfies the equation, $P_{Type\ C3}$, is equal to $\frac{2^{16}}{2^{24}} = 2^{-8}$.

In a $C4$ equation, we have $\mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4) \Leftrightarrow (\Delta X_1 = a_{1,1}\alpha_1 + a_{1,2}\alpha_2) \wedge (\Delta X_2 = a_{2,1}\alpha_1 + a_{2,2}\alpha_2) \wedge (\Delta X_3 = a_{3,1}\alpha_1 + a_{3,2}\alpha_2 \wedge (\Delta X_4 = a_{4,1}\alpha_1 + a_{4,2}\alpha_2)$, where $a_{i,j}$ is a constant. We can rewrite $\mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4)$ as:

$$\begin{aligned} &\mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4) \\ \Leftrightarrow \quad &\begin{cases} a\Delta X_1 + b\Delta X_2 = c\Delta X_3, \\ d\Delta X_1 + e\Delta X_2 = f\Delta X_4, \end{cases} \end{aligned}$$

where a, b, c, d, e, and f are constants. ΔX_3 can take any value but both ΔX_1 and ΔX_2 should satisfy $a\Delta X_1 + b\Delta X_2 = c\Delta X_3$. The number of possible values for $(\Delta X_1, \Delta X_2, \Delta X_3)$ is $2^7 \cdot 2^6$. Similarly the number of possible values for $(\Delta X_1, \Delta X_2, \Delta X_4)$ satisfying $d\Delta X_1 + e\Delta X_2 = f\Delta X_4$ is $2^7 \cdot 2^6$. As both equations should be satisfied together, we find collision between two sets for $(\Delta X_1, \Delta X_2)$. Hence, the number of possible values for $(\Delta X_1, \Delta X_2)$ satisfying both equations is 2^4 and the number of possible values for $(\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4)$ is $(2^4 \cdot 2^7) \times 2 = 2^{12}$. Finally $\#((\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4)| \mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4) \wedge (\Delta Y_i = \beta_i), i = 1, \cdots, 4)$ is 2^{12} and the number of possible candidates for (Y_1, Y_2, Y_3, Y_4) is $2^{12} \times 16 = 2^{16}$. The success probability that a candidate satisfies this equation, $P_{Type\ C4}$, is equal to $\frac{2^{16}}{2^{32}} = 2^{-16}$.

6.2. Differential fault equations in $M2$ and $M3$

We denote new differential fault equations for two, three, and four bytes in $M2$ by *Type* $D2, D3$, and $D4$ equations respectively. Similarly we denote them by *Type* $E2, E3$, and $E4$ equations respectively in $M3$. As we have seen in a $C2$ equation, if the number of bytes in a differential fault equation is not larger than that of faulty bytes, we cannot remove any wrong candidates. That is, $D2, D3, E2, E3$ and $E4$ equations are useless. Therefore we only consider $D4$ equation. Three bytes are corrupted in each column of the input of the i^{th} *MixColumns* in $M2$. Hence we denote an error of each byte by α_i, where $i = 1, 2, 3$. In a $D4$ equation, we have

$$\begin{aligned} & \mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4) \\ \Leftrightarrow \quad & \begin{cases} (\Delta X_1 = a_{1,1}\alpha_1 + a_{1,2}\alpha_2 + a_{1,3}\alpha_3) \\ (\Delta X_2 = a_{2,1}\alpha_1 + a_{2,2}\alpha_2 + a_{2,3}\alpha_3) \\ (\Delta X_3 = a_{3,1}\alpha_1 + a_{3,2}\alpha_2 + a_{3,3}\alpha_3) \\ (\Delta X_4 = a_{4,1}\alpha_1 + a_{4,2}\alpha_2 + a_{4,3}\alpha_3) \end{cases} \\ \Leftrightarrow \quad & a\Delta X_1 + b\Delta X_2 + c\Delta X_3 = d\Delta X_4, \end{aligned}$$

where $a_{i,j}$, a, b, c and d are constants. The number of possible values for $(\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4)$ satisfying $a\Delta X_1 + b\Delta X_2 + c\Delta X_3 = d\Delta X_4$ is 2^{20}. Hence, $\#((\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4)|\mathcal{R}(\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4) \wedge (\Delta Y_i = \beta_i), i = 1, \cdots, 4) = 2^{20}$. The number of possible candidates for (Y_1, Y_2, Y_3, Y_4) is $2^{20} \times 16 = 2^{24}$. The success probability that a candidate satisfies this equation, $P_{Type\ D4}$, is equal to $\frac{2^{24}}{2^{32}} = 2^{-8}$.

7. Differential fault analysis in a multi-byte fault model

We show differential fault analysis on AES-128, AES-192, and AES-256 in multi-byte fault models.

7.1. DFA on AES-128

7.1.1. Attack in $M1$

We assume that a fault is induced between *MixColumns* of round $r-3$ and $r-2$ and corrupts two diagonals D_0 and D_1. Then we can construct four $C4$ differential fault equations at S^r and two $B4$ differential fault equation at S^{r-1}. That is, we have

$$\begin{aligned} & \mathbf{DFE}_1(K_0^r, K_7^r, K_{10}^r, K_{13}^r | S_{0,4,8,12}^r, C4), \\ & \mathbf{DFE}_2(K_1^r, K_4^r, K_{11}^r, K_{14}^r | S_{1,5,9,13}^r, C4), \\ & \mathbf{DFE}_3(K_2^r, K_5^r, K_8^r, K_{15}^r | S_{2,6,10,14}^r, C4), \\ & \mathbf{DFE}_4(K_3^r, K_6^r, K_9^r, K_{12}^r | S_{3,7,11,15}^r, C4), \\ & \mathbf{DFE}_5(\widehat{K}_0^{r-1}, \widehat{K}_7^{r-1}, \widehat{K}_{10}^{r-1}, \widehat{K}_{13}^{r-1} | S_{0,4,8,12}^{r-1}, B4), \\ & \mathbf{DFE}_6(\widehat{K}_1^{r-1}, \widehat{K}_4^{r-1}, \widehat{K}_{11}^{r-1}, \widehat{K}_{14}^{r-1} | S_{1,5,9,13}^{r-1}, B4). \end{aligned} \tag{11}$$

From four $C4$ equations we reduce the number of candidates for K^r from 2^{128} to $2^{128} \cdot (P_{Type\ C4})^4 = 2^{128} \cdot (2^{-16})^4 = 2^{64}$. We further reduce the number of candidates with two $B4$ equations to $2^{64} \cdot (P_{Type\ B4})^4 = 2^{64} \cdot (2^{-24})^2 = 2^{16}$. However the complexity is 2^{64}. Therefore it is better to use one more pair of correct and faulty ciphertexts.

7.1.2. Attack in M2

We assume that a fault is induced between *MixColumns* of round $r-3$ and $r-2$ and corrupts three diagonals D_0, D_1, and D_2. Then we can construct four $D4$ differential fault equations at S^r and three $B4$ differential fault equations at S^{r-1}. From four $D4$ equations we reduce the number of candidates for K^r from 2^{128} to $2^{128} \cdot (P_{Type\ D4})^4 = 2^{128} \cdot (2^{-8})^4 = 2^{96}$. We further reduce the number of candidates with three $B4$ equations to $2^{96} \cdot (P_{Type\ B4})^3 = 2^{96} \cdot (2^{-24})^3 = 2^{24}$. However the complexity is 2^{96}. Therefore we need more pairs of correct and faulty ciphertexts to reduce the complexity. If we have three pairs, we have twelve $D4$ differential fault equations at S^r and nine $B4$ differential fault equations at S^{r-1}. Hence, we can reduce the number of candidates for K^r to $2^{128} \cdot (P_{Type\ D4})^1 2 = 2^{128} \cdot (2^{-8})^1 2 = 2^{32}$. Then using at least two $B4$ equations, we can find the key. We note that our attack needs one less pair compared to that of [35] where four pairs were required.

7.2. DFA on AES-192

7.2.1. Attack in M1

We assume that a fault is induced between *MixColumns* of round $r-3$ and $r-2$ and corrupts two diagonals D_0 and D_1. Then we can construct four $C4$ differential fault equations at S^r and two $B4$ differential fault equations at S^{r-1}. The number of candidates we can reduce with these six equations is $(2^{16})^4 \cdot (2^{24})^2 = 2^{112}$. Hence, we need at least two pairs of correct and faulty ciphertexts.

We assume that two pairs are obtained by giving faults between *MixColumns* of round $r-3$ and $r-2$. We can reduce the number of candidates for K^r to $2^{128} \cdot (P_{Type\ C4})^8 = 2^{128} \cdot (2^{-16})^8 = 1$ with eight $C4$ equations. The left half of K^{r-1} can be computed by AES key schedule. Therefore we have to find the right half of K^{r-1} with four $B4$ equations at S^{r-1}. As seen in Section 5.2, we can construct eight $A1$ equations from these four $B4$ equations. By solving $A1$ equations, we can reduce the number of candidates for the right half of K^{r-1} from 2^{64} to $2^{64} \cdot (P_{Type\ A1})^8 = 2^{64} \cdot (2^{-7})^8 = 2^8$. Finally we have 2^8 candidates for the AES-192 key.

7.2.2. Attack in $M2$

We assume that a fault is induced between *MixColumns* of round $r-3$ and $r-2$ and corrupts three diagonals D_0, D_1, and D_2. Then we can construct four $D4$ differential fault equations at S^r and three $B4$ differential fault equations at S^{r-1}. The number of candidates we can reduce with these seven equations is $(2^8)^4 \cdot (2^{24})^3 = 2^{104}$. Hence, we need at least two pairs of correct and faulty ciphertexts. However, the complexity is 2^{64} as we can reduce the number of candidates for K^r to 2^{64} with two pairs. Therefore we need three pairs. With three pairs, we can reduce the number of candidates for K^r to 2^{32}. The right half of K^{r-1} can be computed from twelve $A1$ equations at S^{r-1}.

7.3. DFA on AES-256

7.3.1. Attack in $M1$

We assume that a fault is injected between *MixColumns* of round $r-3$ and $r-2$ and corrupts two diagonals D_0 and D_1. Then we can construct four $C4$ differential fault equations at S^r and two $B4$ differential fault equations at S^{r-1}. The number of candidates we can reduce with these six equations is $(2^{16})^4 \cdot (2^{24})^2 = 2^{112}$. Hence, we need at least two pairs of correct and faulty ciphertexts. We assume that two pairs are obtained by giving a fault between *MixColumns* of round $r-3$ and $r-2$. We can reduce the number of candidates for K^r to $2^{128} \cdot (P_{Type\,C4})^8 = 2^{128} \cdot (2^{-16})^8 = 1$ with eight $C4$ equations.

We have four $B4$ equations at S^{r-1}. Two are from the 1^{st} pair and the other two are from 2^{nd} pair. If all four equations are constructed at different position of S^{r-1}, we can find 2^{32} candidates for K^{r-1}. Because the number of candidates for each of the four bytes of K^{r-1} is reduced from 2^{32} to 2^8. If two equations are constructed at the same position (we call it a *match*), we can find the corresponding four bytes of K^{r-1}. However, we cannot reduce the number of candidates for one set of four bytes. Therefore the number of candidates for K^{r-1} is $1 \cdot 2^8 \cdot 2^8 \cdot 2^{32} = 2^{48}$. If there are two matches, we have $1 \cdot 1 \cdot 2^{32} \cdot 2^{32} = 2^{64}$ candidates. Therefore we need one more pair of correct and faulty ciphertexts.

We assume that the third pair is obtained by giving a fault one round earlier. Then after finding K^r with the first two pairs, we find 2^{64} candidates for K^{r-1} with the third pair. After solving four $B4$ equations at S^{r-1} from the first and second pairs, the number of candidates for K^{r-1} is 2^{32} at the worst case and 1 at the best case. Because the number of candidates for each set of four bytes of K^{r-1} is 2^{16} before applying four $B4$ equations.

7.3.2. Attack in $M2$

We assume that a fault is induced between *MixColumns* of round $r-3$ and $r-2$ and corrupts three diagonals D_0, D_1, and D_2. Then we can construct four $D4$ differential fault equations at S^r and three $B4$ differential fault equations at S^{r-1}. The number of candidates we can reduce with these seven equations is $(2^8)^4 \cdot (2^{24})^3 = 2^{104}$. Hence, we need at least two pairs of correct and faulty ciphertexts. However, the complexity is 2^{64} as we can reduce the number of candidates for K^r to 2^{64} with two pairs. Therefore we need three pairs.

With three pairs, we can reduce the number of candidates for K^r to 2^{32}. As seen in the previous section, the number of candidates for K^{r-1} is 2^{32} at the worst case and 1 at the best case. Therefore the total number of candidates for the AES-256 key is 2^{32} to 2^{64}. If we have four pairs, we can find K^r and K^{r-1}.

8. Comparison and Conclusions

We summarize differential fault analysis on AES-128, AES-192, and AES-256 in Table 1, 2, and 3.

For one-byte fault model, our attacks on AES-192 and AES-256 show the best performance. In our previous work [21], attacks on AES-192 output 2^{32} or 2^8 candidates with two pairs of correct and faulty ciphertexts. Therefore we have to perform exhaustive search. However the new attack outputs one correct value for the AES-192 key with two pairs. Considering AES-256, our previous attack in [21] outputs 2^{32} candidates but the new attack finds one correct key with the same three pairs of correct and faulty ciphertexts. Our attack on AES-128 in a multi-byte model needs less faulty ciphertexts. In $M2$, we need three pairs of correct and faulty ciphertexts. However the previous work in [35] needs four pairs.

There are four directions in the research of differential fault analysis. Many works have been proposed to improve DFA into one or two directions. In this article we consider the main three directions together and show how to improve differential fault analysis in all three directions. Although the same fault model is used, the required numbers of faults are different in the previous works. This comes from lack of detail examination on constructing and

Table 1: Comparison with existing DFA on AES-128

Reference	Fault model	No. of faults	Exhaustive search
[33]	1 byte random fault	2	1
[41]	1 byte random fault	1	2^{8}
[35]	$M1$ (at most 8 bytes)	2	1
	$M2$ (at most 12 bytes)	4	1
This article	$M1$ (at most 8 bytes)	2	1
	M2 (at most 12 bytes)	**3**	**1**

Table 2: Comparison with existing DFA on AES-192

Reference	Fault model	No. of faults	Exhaustive search
[33]	1 byte random fault	4	1
[25] method 1	1-4 bytes random fault	12	1
[25] method 2	4 bytes random fault	3000	1
[5]	1 byte random fault	16	1
[38]	1 byte random fault	3	2^{8}
[21] attack 1	1 byte random fault	2	2^{32}
[21] attack 2	1 byte random fault	2	2^{8}
This article	**1 byte random fault**	**2**	**1**
	M1 (at most 8 bytes)	**2**	$\mathbf{2^{8}}$
	M2 (at most 12 bytes)	**3**	$\mathbf{2^{32}}$

solving differential fault equations. Hence, we propose a general method of constructing differential fault equations systematically. Based on this we can estimate how many wrong candidates of the subkey can be removed before solving the equations, which leads us to find the minimum number of faults in a given fault model. Following the systematic method of constructing differential fault equations, we propose new differential fault analysis on AES-192 and AES-256 in a one-byte fault model. We further increase our attacks to multi-byte fault models and propose a better attack on AES-128. Finally we propose the first results on differential fault analysis of AES-192 and AES-256 in multi-byte fault models.

Table 3: Comparison with existing DFA on AES-256

Reference	Fault model	No. of faults	Exhaustive search
[33]	1 byte random fault	4	1
[25] method 1	1-4 bytes random fault	12	1
[25] method 2	4 bytes random fault	3000	1
[5]	1 byte random fault	16	1
[38]	1 byte random fault	4	2^{13}
[21]	1 byte random fault	3	2^{32}
This article	**1 byte random fault**	**3**	**1**
	M1 (at most 8 bytes)	**3**	$\mathbf{2^{32}}$
	M2 (at most 12 bytes)	**4**	**1**

References

[1] National Institute of Standard and Technology, *Data Encryption Standard*, NIST FIPS PUB 46-2, 1993.

[2] National Institute of Standard and Technology, *Advanced Encryption Standard*, NIST FIPS PUB 197, 2001.

[3] F. Bao, R. H. Deng, Y. Han, A. B. Jeng, A. D. Narasimhalu, and T.-H. Ngair. Breaking public key cryptosystems on tamper resistant devices in the presence of transient faults. In *Security Protocols, 5th International Workshop*, volume 1361 of *Lecture Notes in Computer Science*, pages 115–124. Springer, 1998.

[4] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The sorcerer's apprentice guide to fault attacks. In *Fault Diagnosis and Tolerance in Cryptography in association with DSN 2004 – The International Conference on Dependable Systems and Networks*, pages 330–342, 2004.

[5] A. Barenghi, G. Bertoni, L. Breveglieri, M. Pellicioli, and G. Pelosi. Low voltage fault attacks to AES and RSA on general purpose processors. IACR eprint archive, 2010-130, 2010.

[6] I. Biehl, B. Meyer, and V. Müller. Differential fault attacks on elliptic curve cryptosystems. In *Advances in Cryptology - CRYPTO 2000, 20th*

Annual International Cryptology Conference, volume 1880 of *Lecture Notes in Computer Science*, pages 131–146. Springer, 2000.

[7] E. Biham, L. Granboulan, and P. Q. Nguyen. Impossible fault analysis of RC4 and differential fault analysis of RC4. In *Fast Software Encryption: 12th International Workshop, FSE 2005*, volume 3557 of *Lecture Notes in Computer Science*, pages 359–367. Springer, 2005.

[8] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 1997.

[9] J. Blömer, M. Otto, and J.-P. Seifert. Sign change fault attacks on elliptic curve cryptosystems. In *2nd International Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2005*, pages 25–40, 2005.

[10] D. Bleichenbacher, M. Joye, and J.-J. Quisquater. A new and optimal chosen-message attack on RSA-type cryptosystems. In *Information and Communication Security, First International Conference, ICICS'97*, volume 1334 of *Lecture Notes in Computer Science*, pages 302–313. Springer, 1997.

[11] J. Blömer and J.-P. Seifert. Fault based cryptanalysis of the advanced encryption standard (AES). In *Financial Cryptography, 7th International Conference, FC 2003*, volume 2742 of *Lecture Notes in Computer Science*, pages 162–181. Springer, 2003.

[12] C.-N. Chen and S.-M. Yen. Differential fault analysis on AES key schedule and some coutnermeasures. In *Information Security and Privacy, 8th Australasian Conference, ACISP 2003*, volume 2727 of *Lecture Notes in Computer Science*, pages 118–129. Springer, 2003.

[13] H. Chen, W. Wu, and D. Feng. Differential fault analysis on CLEFIA. In *Information and Communications Security, 9th International Conference, ICICS 2007*, volume 4861 of *Lecture Notes in Computer Science*, pages 284–295. Springer, 2007.

[14] M. Ciet and M. Joye. Elliptic curve cryptosystems in the presence of permanent and transient faults. *Des. Codes Cryptography*, 36(1):33–43, 2005.

[15] J.-S. Coron, C. Giraud, N. Morin, G. Piret, and D. Vigilant. Fault attacks and countermeasures on vigilant's RSA-CRT algorithm. In *Fault Diagnosis and Tolerance in Cryptography, 7th International Workshop, FDTC 2010*, pages 89–96. IEEE Computer Society, 2010.

[16] P. Dusart, G. Letourneux, and O. Vivolo. Differential fault analysis on AES. In *Applied Cryptography and Network Security, First International Conference, ACNS 2003*, volume 2846 of *Lecture Notes in Computer Science*, pages 293–306. Springer, 2003.

[17] T. Fukunaga and J. Takahashi. Practical fault attack on a cryptographic LSI with ISO/IEC 18033-3 block ciphers. In *6th International Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2009*, pages 84–92. IEEE Computer Society, 2009.

[18] C. Giraud. DFA on AES. In *Advanced Encryption Standard - AES, 4th International Conference, AES 2004*, volume 3373 of *Lecture Notes in Computer Science*, pages 27–41. Springer, 2005.

[19] L. Hemme. A differential fault attack against early rounds of (Triple-)DES. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop*, volume 3156 of *Lecture Notes in Computer Science*, pages 254–267. Springer, 2004.

[20] J. J. Hoch and A. Shamir. Fault analysis of stream ciphers. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop*, volume 3156 of *Lecture Notes in Computer Science*, pages 240–253. Springer, 2004.

[21] C. H. Kim. Differential fault analysis against AES-192 and AES-256 with minimal faults. In *Fault Diagnosis and Tolerance in Cryptography, 7th International Workshop, FDTC 2010*, pages 3–9. IEEE Computer Society, 2010.

[22] C. H. Kim. Differential fault analysis against AES-192 and AES-256 with minimal faults. Slides of the presentation at FDTC 2010, avail-

able at http://sites.uclouvain.be/security/download/slides/Kim-2010-fdtc-slides.pdf, 2010.

[23] C. H. Kim and J.-J. Quisquater. New differential fault analysis on AES key schedule: Two faults are enough. In *Smart Card Research and Advanced Applications, 8th IFIP WG 8.8/11.2 International Conference, CARDIS 2008*, volume 5189 of *Lecture Notes in Computer Science*, pages 48–60. Springer, 2008.

[24] W. Li, D. Gu, and Y. Wang. Differential fault analysis on the contracting UFN structure, with application to SMS4 and Macguffin. *Journal of Systems and Software*, 82(2):346–354, 2009.

[25] W. Li, D. Gu, Y. Wang, J. Li, and Z. Liu. An extension of differential fault analysis on AES. In *International Conference on Network and System Security*, pages 443–446, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[26] Y. Li, S. Gomisawa, K. Sakiyama, and K. Ohta. An information theoretic perspective on the differential fault analysis against AES. IACR eprint archive, 2010-032, 2010.

[27] I.-C. Lin and C.-C. Chang. Security enhancement for digital signature schemes with fault tolerance in rsa. *Inf. Sci.*, 177(19):4031–4039, 2007.

[28] A. Moradi, M. T. M. Shalmani, and M. Salmasizadeh. A generalized method of differential fault attack against AES cryptosystem. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop*, volume 4249 of *Lecture Notes in Computer Science*, pages 91–100. Springer, 2006.

[29] D. Mukhopadhyay. An improved fault based attack of the advanced encryption standard. In *AFRICACRYPT 2009*, volume 5580 of *Lecture Notes in Computer Science*, pages 421–434. Springer-Verlag, 2009.

[30] D. Naccache, P. Q. Nguyen, M. Tunstall, and C. Whelan. Experimenting with faults, lattices and the DSA. In *Public Key Cryptography - PKC 2005, 8th International Workshop on Theory and Practice in Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 16–28. Springer, 2005.

[31] A. Pellegrini, V. Bertacco, and T. Austin. Fault-based attack of RSA authentication. In *2010 Design, Automation and Test in Europe*, pages 855–860, 2010. Available at www.date-conference.com/proceedings/PAPERS/2010/YEAR.HTM.

[32] R. C.-W. Phan and S.-M. Yen. Amplifying side-channel attacks with techniques from block cipher cryptanalysis. In *Smart Card Research and Advanced Applications, 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006*, volume 3928 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 2006.

[33] G. Piret and J.-J. Quisquater. A differential fault attack technique against SPN structures, with application to the AES and KHAZAD. In *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop*, volume 2779 of *Lecture Notes in Computer Science*, pages 77–88. Springer, 2003.

[34] M. Rivain. Differential fault analysis on DES middle rounds. In *Cryptographic Hardware and Embedded Systems - CHES 2009: 11th International Workshop*, volume 5747 of *Lecture Notes in Computer Science*, pages 457–469. Springer, 2009.

[35] D. Saha, D. Mukhopadhyay, and D. RoyChowdhury. A diagonal fault attack on the advanced encryption standard. IACR eprint archive, 2009-581, 2009.

[36] J. Takahashi and T. Fukunaga. Differential fault analysis on the AES key schedule. IACR eprint archive, 2007-480.

[37] J. Takahashi and T. Fukunaga. Improved differential fault analysis on CLEFIA. In *Fifth International Workshop on Fault Diagnosis and Tolerance in Cryptography, 2008, FDTC 2008*, pages 25–34. IEEE Computer Society, 2008.

[38] J. Takahashi and T. Fukunaga. Differential fault analysis on AES with 192 and 256-bit keys. IACR eprint archive, 2010-023, 2010.

[39] J. Takahashi, T. Fukunaga, and K. Yamakoshi. DFA mechanism on the AES key schedule. In *4th International Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2007*, pages 62–74. IEEE Computer Society, 2007.

[40] E. Trichina and R. Korkikyan. Multi fault laser attacks on protected CRT-RSA. In *Fault Diagnosis and Tolerance in Cryptography, 7th International Workshop, FDTC 2010*, pages 75–86. IEEE Computer Society, 2010.

[41] M. Tunstall and D. Mukhopadhyay. Differential fault analysis of the advanced encryption standard using a single fault. IACR eprint archive, 2009-575.