Differential Fault Analysis of AES using a Single Multiple-Byte Fault

Subidh Ali[1], Debdeep Mukhopadhyay[1], and Michael Tunstall[2]

[1] Department of Computer Sc. and Engg, IIT Kharagpur, West Bengal, India.
{subidh,debdeep}@cse.iitkgp.ernet.in
[2] Department of Computer Science, University of Bristol,
Merchant Venturers Building, Woodland Road,
Bristol BS8 1UB, United Kingdom.
tunstall@cs.bris.ac.uk

Abstract. In this paper we present an improved fault attack on the Advanced Encryption Standard (AES). This paper presents an improvement on a recently published differential fault analysis of AES that requires one fault to recover the secret key being used. This attack requires that one byte entering into the eighth round is corrupted. We show that the attack is possible where more than one byte has been affected. Experimental results are described where a fault is injected using a glitch in the clock, demonstrating that this attack is practical.

1 Introduction

There are numerous methods for injecting a fault into a microprocessor, such as electromagnetic radiation, light, temperature variations, spikes in supply voltages and clock glitches [1]. Preventing all of these methods is becoming more and more complex for hardware designers. This problem is compounded by the ongoing scaling of devices to nano technologies where faults will be even harder to prevent. While these faults may have undesirable effects on normal applications, it can be totally disastrous for cryptographic systems. This was first noted by Boneh et al. [4], who observed that a single fault in one of the two exponentiations required to generate a RSA signature using the Chinese remainder theorem would allow an attacker to retrieve the private key. Subsequently, Biham and Shamir [2] proposed the idea of Differential Fault Analysis (DFA), based on differential cryptanalysis, to attack DES. This fault attack required around 50 to 1500 faulty ciphertexts to extract an entire secret key, assuming that a one bit fault was being introduced at a random point in the algorithm during each execution.

In 2001 NIST standardized a new block cipher named Advanced Encryption Standard (AES) [8]. Subsequently, it has become a popular target for cryptanalysis because it will, in many cases, replace DES. In 2004 Giraud described a differential fault analysis of AES that required a single byte fault to be introduced at the beginning of the ninth round and 250 fault ciphertexts [6]. Similarly, Blömer and Seifert proposed an attack that required between 128 and 256 faulty ciphertexts [3]. The attack on AES was further improved by Dusart et al. who prosed an attack that required 50 faulty ciphertexts [5].

Piret and Quisquater showed that a DFA on AES is possible with only two faulty ciphertexts and a key search of 48 and 40 bits [10]. This was subsequently improved upon by Mukhopadhyay who pointed out that the fault attack against AES can be performed with a single byte fault and an key search among 2^{32} key hypotheses [7]. This analysis was extended by Tunstall and Mukhopadhyay where it was shown that the number of key hypotheses can be reduced to 2^8 from a single fault [12].

Recent work by Saha et al. has shown that multiple byte faults can be analyzed in the same manner described by Mukhopadhyay [11]. That is, if multiple byte faults are induced,

while leaving a significant number of bytes untouched, a secret key can still be recovered using differential fault analysis. This paper revisits this work and applies the method described by Tunstall and Mukhopadhyay [12] to improve the attack. We also present experimental results to support the claims made by demonstrating the fault attack on an iterated AES core prototyped on a Xilinx Spartan-3E platform using glitches in the clock line.

Notation

In this paper, multiplications are considered to be polynomial multiplications over $\mathbb{F}_{2^8}$ modulo the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. It should be clear from the context when a mathematical expression contains integer multiplication.

Organization

The paper is organized as follows: In Section 2 we describe the background to this paper. In Section 3 we describe a previously published attack based on one of the fault models given in Section 2. In Section 4 we extend the work to multi byte fault models. In Section 5 we describe some experimental results, and we conclude in Section 6.

2 Background

2.1 The Advanced Encryption Standard

Algorithm 1: The AES-128 encryption function.

Input: The 128-bit plaintext block P and key K.
Output: The 128-bit ciphertext block C.

$X \leftarrow$ `AddRoundKey`(P, K) ;
for $i \leftarrow 1$ **to** 10 **do**
  $X \leftarrow$ `SubBytes`(X) ;
  $X \leftarrow$ `ShiftRows`(X) ;
  if $i \neq 10$ **then**
    $X \leftarrow$ `MixColumns`(X) ;
  end
  $K \leftarrow$ `KeySchedule`(K) ;
  $X \leftarrow$ `AddRoundKey`(X, K) ;
end
$C \leftarrow X$;
return C

The structure of the Advanced Encryption Standard (AES), as used to perform encryption, is illustrated in Algorithm 1. Note that we restrict ourselves to considering AES-128 and that the description above omits a permutation typically used to convert the plaintext $P = (p_1, p_2, \ldots, p_{16})_{(256)}$ and key $K = (k_1, k_2, \ldots, k_{16})_{(256)}$ into a 4×4 array of bytes, known as the state matrix. For example, the 128-bit plaintext input block to AES is arranged in the following fashion

$$\begin{pmatrix} p_1 & p_5 & p_9 & p_{13} \\ p_2 & p_6 & p_{10} & p_{14} \\ p_3 & p_7 & p_{11} & p_{15} \\ p_4 & p_8 & p_{12} & p_{16} \end{pmatrix}$$

The corresponding fault free (CT) and faulty ciphertexts (CT') are respectively:

$$\mathbf{CT} = \begin{pmatrix} x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \\ x_4 & x_8 & x_{12} & x_{16} \end{pmatrix} \quad \mathbf{CT'} = \begin{pmatrix} x'_1 & x'_5 & x'_9 & x'_{13} \\ x'_2 & x'_6 & x'_{10} & x'_{14} \\ x'_3 & x'_7 & x'_{11} & x'_{15} \\ x'_4 & x'_8 & x'_{12} & x'_{16} \end{pmatrix}$$

where $x_i \in \{0, \ldots, 255\}$. $\forall i \in \{1, \ldots, 16\}$.

We also define the key matrix for the subkeys used in the ninth and tenth round as:

$$\mathbf{K_{10}} = \begin{pmatrix} k_1 & k_5 & k_9 & k_{13} \\ k_2 & k_6 & k_{10} & k_{14} \\ k_3 & k_7 & k_{11} & k_{15} \\ k_4 & k_8 & k_{12} & k_{16} \end{pmatrix} \quad \mathbf{K_9} = \begin{pmatrix} k'_1 & k'_5 & k'_9 & k'_{13} \\ k'_2 & k'_6 & k'_{10} & k'_{14} \\ k'_3 & k'_7 & k'_{11} & k'_{15} \\ k'_4 & k'_8 & k'_{12} & k'_{16} \end{pmatrix}$$

The encryption itself is conducted by the repeated use of a number of round functions:

- The `SubBytes` function is the only non-linear step of the block cipher. It is a bricklayer permutation consisting of an S-box applied to the bytes of the state. Each byte of the state matrix is replaced by its multiplicative inverse, followed by an affine mapping. Thus the input byte x is related to the output y of the S-Box by the relation, $y = A\,x^{-1} + B$, where A and B are constant matrices. In the remainder of this paper we will refer to the function S as the SubBytes function and S^{-1} as the inverse of the SubBytes function.
- The `ShiftRows` function is a byte-wise permutation of the state.
- The `KeySchedule` function generates the next round key from the previous one. The first round key is the input key with no changes, subsequent round keys are generated using the `SubBytes` function and XOR operations. This is shown in Algorithm 2, that shows how the r^{th} round key is computed from the $(r-1)^{th}$ round key. The value h_r is a constant defined for the r^{th} round, and $<<$ is used to denote a bitwise left shift.
- The `MixColumn` is a bricklayer permutation operating on the state column by column. Each column of the state matrix is considered as a 4-dimensional vector where each element belongs to $\mathbb{F}(2^8)$. A 4×4 matrix M whose elements are also in $\mathbb{F}(2^8)$ is used to map this column into a new vector. This operation is applied on all the 4 columns of the state matrix. Here M and its inverse M^{-1} are defined as:

$$M = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \quad M^{-1} = \begin{pmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{pmatrix}$$

  All the elements in M and M^{-1} are elements of $\mathbb{F}(2^8)$ expressed as a decimal digit.
- `AddRoundKey`: Each byte of the array is XORed with a byte from a corresponding array of round subkeys.

2.2 Fault Model of the Attack

The fault model is central to the description of a fault based cryptanalysis. In our attack we consider two types of faults: one byte faults and faults that affect multiple bytes lying in different diagonals of the state matrix of AES. We formally define the diagonal of the AES state matrix as follows:

Definition 1. Diagonal: *A diagonal is a set of four bytes of the state matrix, where the i^{th} diagonal is defined as follows.* $D_i = \{b_{j,(j+i)mod4}$ *; where* $1 \leq j \leq 4\}$

Formally the fault models are classified as follows:

1. **Model $\mathbf{M^{(1)}}$:** Faults under this class are single byte faults, that only affect one of the four diagonals.

Algorithm 2: The AES-128 `KeySchedule` function.

Input: $(r-1)^{th}$ round key ($X = x_i$ for $i \in \{1, \ldots, 16\}$).
Output: r^{th} round key X.

for $i \leftarrow 0$ **to** 3 **do**
    $x_{i+1} \leftarrow x_{i+1} \oplus S(x_{((i+1)\wedge 3)+13})$;
end
$x_1 \leftarrow x_1 \oplus h_r$;
for $i \leftarrow 5$ **to** 16 **do**
    $x_i \leftarrow x_i \oplus x_{i-4}$;
end
return X

2. **Model $\mathbf{M_d}$:** Faults under this class are faults that affect multiple bytes. They affect d diagonals of the state matrix, where $1 \leq d \leq 4$. This model is further classified into four different submodels:
   (a) **Model $\mathbf{M_1^{(i)}}$:** The faults in this class affect i byte locations of one diagonal; where $2 \leq i \leq 4$.
   (b) **Model $\mathbf{M_2^{(i,j)}}$:** The faults in this class affect two of the four diagonals; One with i modified bytes and the other with j modified bytes; where $1 \leq i, j \leq 4$.
   (c) **Model $\mathbf{M_3^{(i,j,k)}}$:** The faults in this class affect three of the four diagonals, where the faulty diagonals have i, j and k modified bytes respectively; where $1 \leq i, j, k \leq 4$.
   (d) **Model $\mathbf{M_4}$:** Faults in this class are multiple byte faults, that affect all of the four diagonals .

The rationale of the above fault model comes from observations described in [11] where faults are injected into an iterated implementation of AES using a glitch in the clock frequency. As the clock frequency of the glitches was increased, the number of bytes affected spread along the different diagonals.

3 Previous Work

3.1 Analyzing the Final Round

An attack that requires one ciphertext where a fault has been injected into the beginning of the eighth round of an instantiation of AES is described in [7]. This attack reduces the number of possible keys from 2^{128} to 2^{32}. A one byte fault injected into the beginning of the eighth round will propagate as shown in Figure 1. This corresponds to the model $\mathbf{M^{(1)}}$ defined above.

Let us consider CT, CT', and K_{10} as defined in Section 2.1 where all elements are in $\mathbb{F}_{2^8}$. If we use the interrelation between the faulty bytes of the first column c_1 at the end of the ninth round `MixColumn` as in Figure 1, we have following four equations:

$$\begin{aligned} 2\,F &= S^{-1}(x_1 \oplus k_1) \oplus S^{-1}(x'_1 \oplus k_1) \\ F &= S^{-1}(x_{14} \oplus k_{14}) \oplus S^{-1}(x'_{14} \oplus k_{14}) \\ F &= S^{-1}(x_{11} \oplus k_{11}) \oplus S^{-1}(x'_{11} \oplus k_{11}) \\ 3\,F &= S^{-1}(x_8 \oplus k_8) \oplus S^{-1}(x'_8 \oplus k_8) \end{aligned},$$

where $F \in \mathbb{F}_{2^8}$. These four equations can be solved for four key bytes k_1, k_8, k_{11} and k_{14}. The key space of this quadruplet of key bytes is reduced to an expected value of 2^8. This can be repeated for the each of the columns using similar formulae to produce 2^{32} possible key hypotheses, as described in [7].

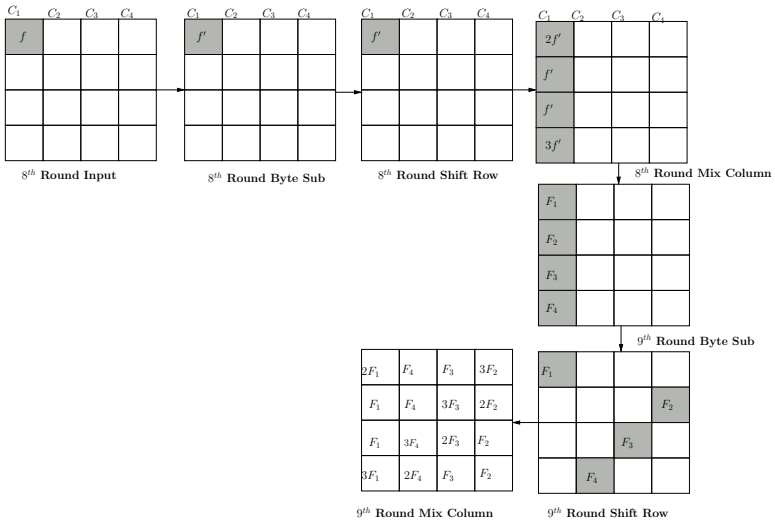

Fig. 1. Propagation of byte fault induced at the input of eighth round

3.2 Extending to Multiple Rounds

If we consider the difference at the end of the eighth round, as defined in Section 3.1, a similar analysis can be conducted using Equations 1–4 defined below. We further define $e_1 = 2\,f'$, $e_2 = f'$, $e_3 = f'$ and $e_4 = 3\,f'$ where $f', e_i \forall i \in \{1, \ldots, 4\}$ are $\in \mathbb{F}_{2^8}$. The number of key hypotheses remaining after evaluating these equations is expected to be 2^8 [12].

$$\begin{aligned} e_1 = S^{-1}\Big(14\left(S^{-1}(x_1 \oplus k_1) \oplus ((k_1 \oplus S(k_{14} \oplus k_{10}) \oplus h_{10}))\right) \oplus 11\left(S^{-1}(x_8 \oplus k_8) \oplus \right. \\ \left.(k_2 \oplus S(k_{15} \oplus k_{11}))\right) \oplus 13\left(S^{-1}(x_{11} \oplus k_{11}) \oplus (k_3 \oplus S(k_{16} \oplus k_{12}))\right) \oplus \\ 9\left(S^{-1}(x_8 \oplus k_8) \oplus (k_4 \oplus S(k_{13} \oplus k_9))\right)\Big) \oplus S^{-1}\Big(14\left(S^{-1}(x'_1 \oplus k_1)\right. \\ \left. \oplus ((k_1 \oplus S(k_8 \oplus k_{10}) \oplus h_{10}))\right) \oplus 11\left(S^{-1}(x'_8 \oplus k_8) \oplus (k_2 \oplus S(k_{15} \oplus k_{11})\right) \oplus \\ 13\left(S^{-1}(x'_{11} \oplus k_{11}) \oplus (k_3 \oplus S(k_{16} \oplus k_{12}))\right) \oplus 9\left(S^{-1}(x'_8 \oplus k_8) \oplus \right. \\ \left.(k_4 \oplus S(k_{13} \oplus k_9))\right)\Big) \end{aligned} \tag{1}$$

$$\begin{aligned} e_2 = S^{-1}\Big(9\left(S^{-1}(x_{13} \oplus k_{13}) \oplus (k_{13} \oplus k_9)\right) \oplus 14\left(S^{-1}(x_{10} \oplus k_{10}) \oplus (k_{10} \oplus k_{14})\right)\Big) \oplus \\ 11\left(S^{-1}(x_7 \oplus k_7) \oplus (k_{15} \oplus k_{11})\right) \oplus 13\Big(S^{-1}(x_4 \oplus k_4) \oplus (k_{16} \oplus k_{12})\Big)\Big) \oplus \\ S^{-1}\Big(9\left(S^{-1}(x'_{13} \oplus k_{13}) \oplus (k_{13} \oplus k_9)\right) \oplus 14\left(S^{-1}(x'_{10} \oplus k_{10}) \oplus (k_{10} \oplus k_{14})\right)\Big) \oplus \\ 11\left(S^{-1}(x'_7 \oplus k_7) \oplus (k_{15} \oplus k_{11})\right) \oplus 13\left(S^{-1}(x'_4 \oplus k_4) \oplus (k_{16} \oplus k_{12})\right)\Big) \end{aligned} \tag{2}$$

$$\begin{aligned} e_3 = S^{-1}\Big(13\left(S^{-1}(x_9 \oplus k_9) \oplus (k_9 \oplus k_5)\right) \oplus 9\left(S^{-1}(x_6 \oplus k_6) \oplus (k_{10} \oplus k_6)\right)\Big) \oplus \\ 14\left(S^{-1}(x_3 \oplus k_3) \oplus (k_{11} \oplus k_7)\right) \oplus 11\left(S^{-1}(x_{16} \oplus k_{16}) \oplus (k_{12} \oplus k_8)\right)\Big) \oplus \\ S^{-1}\Big(13\left(S^{-1}(x'_9 \oplus k_9) \oplus (k_9 \oplus k_5)\right) \oplus 9\left(S^{-1}(x'_6 \oplus k_6) \oplus (k_{10} \oplus k_6)\right)\Big) \oplus \\ 14\left(S^{-1}(x'_3 \oplus k_3) \oplus (k_{11} \oplus k_7)\right) \oplus 11\left(S^{-1}(x'_{16} \oplus k_{16}) \oplus (k_{12} \oplus k_8)\right)\Big) \end{aligned} \tag{3}$$

$$
\begin{aligned}
e_4 = S^{-1}\Big(11\left(S^{-1}(x_2 \oplus k_2) \oplus (k_2 \oplus k_1)\right) \oplus 13\left(S^{-1}(x_5 \oplus k_5) \oplus (k_6 \oplus k_5)\right)\Big) \oplus \\
9\left(S^{-1}(x_{12} \oplus k_{12}) \oplus (k_{10} \oplus k_9)\right) \oplus 14\left(S^{-1}(x_{15} \oplus k_{15}) \oplus (k_{14} \oplus k_{13})\right)\Big) \oplus \\
S^{-1}\Big(11\left(S^{-1}(x'_2 \oplus k_2) \oplus (k_2 \oplus k_1)\right) \oplus 13\left(S^{-1}(x'_5 \oplus k_5) \oplus (k_6 \oplus k_5)\right)\Big) \oplus \\
9\left(S^{-1}(x'_{12} \oplus k_{12}) \oplus (k_{10} \oplus k_9)\right) \oplus 14\left(S^{-1}(x'_{15} \oplus k_{15}) \oplus (k_{14} \oplus k_{13})\right)\Big)
\end{aligned} \tag{4}
$$

4 Proposed Multi Byte Attack Based on Model M_d

In this section we perform an analysis similar to that described in the section where the faults injected correspond to the multiple byte fault model M_d, where d is the number of diagonals affected by a fault and $1 \leq d \leq 4$. In this section we describe attacks based on previous work by Saha et al. where only the last round is analyzed [11], in a similar manner to that described in Section 3.2. For each model we extend the attack by Saha et al. by adding a second phase to the attack.

4.1 Multi Byte Attack Based on Model $M_1^{(i)}$

In this section we consider model $M_1^{(i)}$, where i bytes of one diagonal are affected by a fault injected at the beginning of the eighth round of an instantiation of AES. In all cases the first phase of the attack is as described in [11], since all the affected bytes are in the same column after the computation of the `MixColumn` at the end of the eighth round. This can be seen in Figure 2 that shows the propagation of fault; where a fault corrupts only one diagonal of the state matrix.

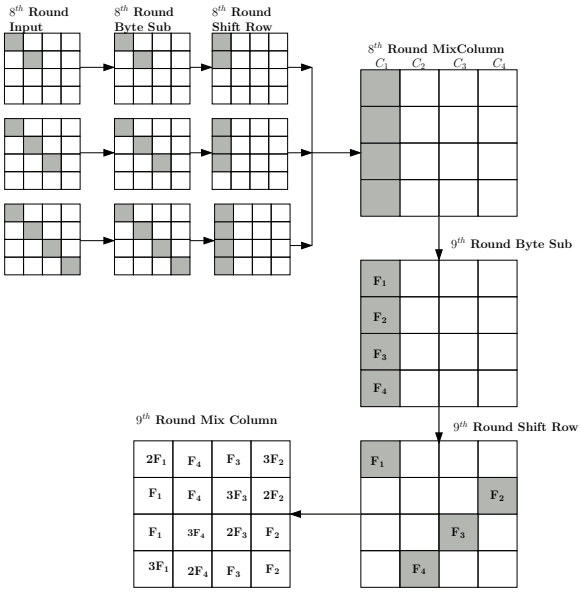

Fig. 2. Propagation of one diagonal fault induced at the input of eighth round.

Hence the first phase analysis of all the instances of fault model $M_1^{(i)}$ will produce 2^{32} key hypotheses, as described in Section 3.1. These 2^{32} key hypotheses can further be reduced by a second phase of the attacks. The key reduction in the second phase is dependent on different

instances of model $M_1^{(i)}$. For example, model $M_1^{(1)}$ will correspond to the fault model required for the attack described in Section 3 and the number of key hypotheses can be reduced using the analysis described in Section 3.2. In the following sections we describe the second phase for the remaining models $M_1^{(i)}$ for $2 \leq i \leq 4$.

Proposed Second Phase of the Attack Based on Model $M_1^{(2)}$. In this model an attacker is expected to have injected a two bytes fault in any one of the four diagonals. Figure 3 shows the propagation of fault based on model $M_1^{(2)}$. Given that the number of key hypotheses has been reduced to 2^{32} by the first phase of the attack.

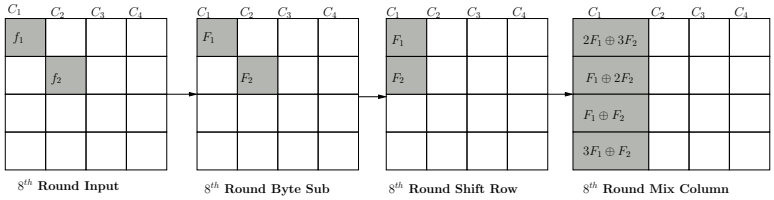

Fig. 3. Propagation of two byte faults induced at the input of eighth round

If we denote the fault values of the first column at the end of eighth round `MixColumn` operation by e_1, e_2, e_3, and e_4 we get

$$e_1 = 2\,F_1 \oplus 3\,F_2, \quad e_2 = F_1 \oplus 2\,F_2, \quad e_3 = F_1 \oplus F_2, \quad \text{and} \quad e_4 = 3\,F_1 \oplus F_2$$

where e_i for $e_i \in \{1, \ldots, 4\}$ is defined as described in Section 3.2. If we eliminate F_1, F_2 from the above system of equations we will have the following relationship between e_1, e_2, e_3 and e_4,

$$e_2 \oplus e_4 = e_1 \quad \text{and} \quad 2\,e_2 \oplus 3\,e_4 = 7\,e_3 \ .$$

Here, it is clear that if we fix any two of the four variables $\{e_1, e_2, e_3, e_4\}$ then the remaining two variables can only take one value. This gives 2^{16} possible values for the quadruplet $\{e_1, e_2, e_3, e_4\}$ each of which will produce 0, 2 or 4 possible key hypotheses, but is expected to return one key hypotheses [9]. We can, therefore, state that an arbitrary key value will produce a valid quadruplet $\{e_1, e_2, e_3, e_4\}$ with a probability of $2^{16}/2^{32} = 2^{-16}$. This will, therefore, reduce the number of possible key hypotheses to 2^{16}.

Proposed Second Phase of the Attack Based on Model $M_1^{(3)}$. Similarly, if three bytes in one diagonal are affect by a fault then we can define

$$e_1 = 2\,F_1 \oplus 3\,F_2 \oplus F_3, \quad e_2 = F_1 \oplus 2\,F_2 \oplus 3\,F_3,$$
$$e_3 = F_1 \oplus F_2 \oplus 2\,F_3, \quad \text{and} \quad e_4 = 3\,F_1 \oplus F_2 \oplus F_3\,,$$

where we assume that the effect of the fault is as shown in Figure 4.

If we eliminate F_1, F_2 and F_3 from the above system of equations we will have the following relationship between e_1, e_2, e_3 and e_4,

$$11\,e_1 \oplus 13\,e_2 = 9\,e_3 \oplus 14\,e_4 \ .$$

It is interesting to note that this can be written as $11\,e_1 \oplus 13\,e_2 \oplus 9\,e_3 \oplus 14\,e_4 = 0$, where the coefficients correspond to the values in the Inverse MixColumn matrix M^{-1}. Following the same reasoning as above this will reduce the number of key hypotheses from 2^{32} to 2^{24}.

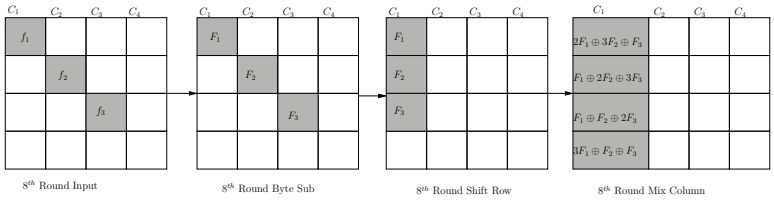

Fig. 4. Propagation of four byte faults induced at the input of eighth round

Proposed Multi Bytes Attack Based on Mode $M_1^{(4)}$. If all four bytes in a diagonal are affected then there is no information to be exploited in a second analysis phase. This is because there will be 2^{32} possible valid combinations for $\{e_1, e_2, e_3, e_4\}$. The number of key hypotheses will not go below the 2^{32} found using the attack defined by Saha et al. [11].

4.2 Proposed Multi Byte Attack Based one Model $M_2^{(i,j)}$

In this section we apply our previous analysis on two diagonal fault model $M_2^{(i,j)}$; where the fault affects two of the four diagonal of the state matrix and i and j bytes are affected in these diagonals. As in Section 4.1, the attack is broken into two phases where the first phase has been defined previously by Saha et al. [11].

The First Phase of the Attack Based on Model $M_2^{(i,j)}$. In the first phase of the attack the difference in the last round is produced by two columns of the state matrix at the end of the eighth round being corrupted by a fault. This analysis will be identical for all values of i and j assuming the same two diagonals are affected. If, for example, the two left-most columns are affected then the fault will propagate through the ninth round as shown in Figure 5.

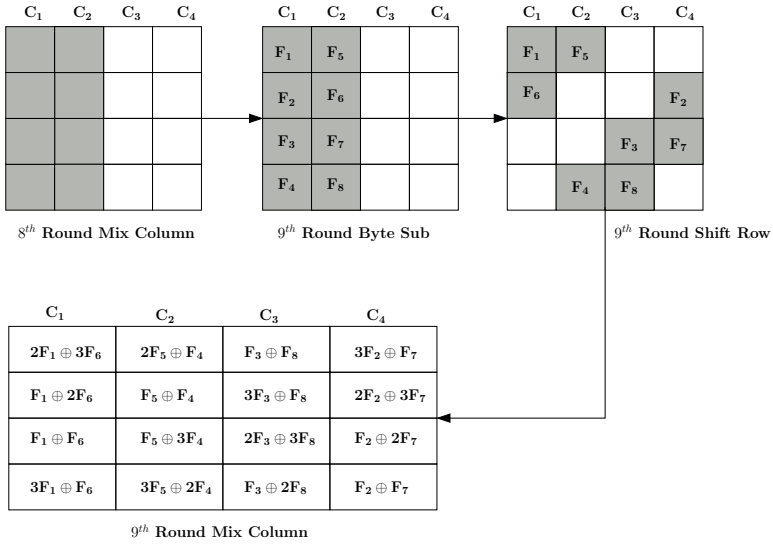

Fig. 5. Propagation of two byte faults induced at the input of the eighth round.

For each column can be analyzed independently. If we consider the first column c_1 and define the difference

$$a_1 = 2\,F_1 \oplus 3\,F_6, \quad a_2 = F_1 \oplus 2\,F_6, \quad a_3 = F_1 \oplus F_6, \quad \text{and} \quad a_4 = 3\,F_1 \oplus F_6\,,$$

as defined in Figure 5. If we eliminate F_1, F_6 from the above system of equations we will have a relation between a_1, a_2, a_3 and a_4, we have

$$a_2 \oplus a_4 = a_1 \quad \text{and} \quad 2\,a_2 \oplus 3\,a_4 = 7\,a_3$$

where a_1, a_2, a_3, and a_4 can be expressed as,

$$a_1 = S^{-1}(x_1 \oplus k_1) \oplus S^{-1}(x'_1 \oplus k_1), \quad a_2 = S^{-1}(x_{14} \oplus k_{14}) \oplus S^{-1}(x'_{14} \oplus k_{14}),$$
$$a_3 = S^{-1}(x_{11} \oplus k_{11}) \oplus S^{-1}(x'_{11} \oplus k_{11}), \quad \text{and} \quad a_4 = S^{-1}(x_8 \oplus k_8) \oplus S^{-1}(x'_8 \oplus k_8)\ .$$

As described above this gives 2^{16} possible values for the quadruplet $\{a_1, a_2, a_3, a_4\}$ each of which will produce 0, 2 or 4 possible key hypotheses, but is expected to return one key hypotheses [9]. This will, therefore, produce 2^{16} hypotheses for the quadruplet $\{k_1, k_8, k_{11}, k_{14}\}$ and, therefore, 2^{64} key hypotheses for the secret key from analyzing all the four columns.

Proposed Second Phase of the Attack Based on Model $M_2^{(i,j)}$. In this section we present the proposed second phase of the attack on model $M_2^{(i,j)}$. There are 10 different instances of model $M_2^{i,j}$ based on the number of faulty bytes in two faulty diagonals. If we consider the two faulty diagonals as D_x and D_y and corresponding faulty columns as c_x and c_y at the end of eighth round `MixColumn` then, depending on the values of i and j, each of the two faulty columns c_x and c_y can produce three different systems of equations in the manner described above. It is clear from Section 4.1 that a faulty diagonal where all four bytes are affected does not help in reducing key hypotheses in the second phase.

Both sets of equations can be evaluated independently, and can both be used to reduce the number of valid key hypotheses. For example, if we consider $M_2^{(1,1)}$, there two faulty diagonals in this model are D_1 and D_2 and the corresponding infected columns are c_1 and c_2. Figure 6 shows the propagation of a fault corresponding to model $M_2^{(1,1)}$.

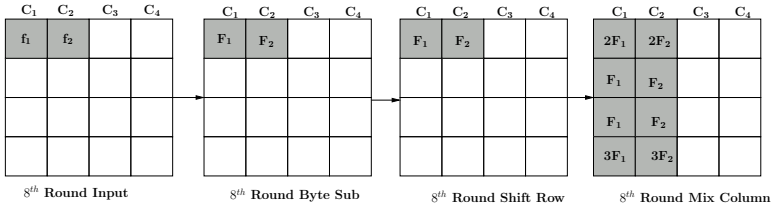

Fig. 6. Propagation of faults based on model $M_2^{(1,1)}$

In the first stage of this attack the 2^{64} key hypotheses generated from the first phase of the attack is tested by the system of equations generated for columns c_1 and c_2. We note that the first stage of the attack will produce four sets of 2^{16} hypotheses each of which corresponds to 32 bits of the secret key. An attacker is not required to search through the entire 2^{64} hypotheses.

Each of the sets of equations will validate a given key hypothesis with a probability of $p_{i,j} = 2^{-24}$, and, therefore, the probability both sets of equations validate a given key hypothesis will be 2^{-48}. The number of key hypotheses returned is $2^{64} \cdot 2^{-48} = 2^{16}$.

This method can be applied to all the instances of model $M_2^{i,j}$. The result of these analysis is shown in Table 1. As the Table 1 shows the proposed second phase of the attack gives best result on the model $M_2^{(1,1)}$ whereas it does not work for the model $M_2^{(4,4)}$. Therefore, we conclude in this section that the attack is most effective when the least number of bytes are affected.

Table 1. Results of The Proposed Second Phase of the Attack on Model $M_2^{i,j}$.

Model ($M_2^{i,j}$)	Probability ($p_{i,j}$)	Key Hypotheses
$M_2^{1,1}$	2^{-48}	2^{16}
$M_2^{1,2}$	2^{-40}	2^{24}
$M_2^{1,3}$	2^{-32}	2^{32}
$M_2^{1,4}$	2^{-24}	2^{40}
$M_2^{2,2}$	2^{-32}	2^{32}

Model ($M_2^{i,j}$)	Probability ($p_{i,j}$)	Key Hypotheses
$M_2^{2,3}$	2^{-24}	2^{40}
$M_2^{2,4}$	2^{-16}	2^{48}
$M_2^{3,3}$	2^{-16}	2^{48}
$M_2^{3,4}$	2^{-8}	2^{56}
$M_2^{4,4}$	–	2^{64}

4.3 Proposed Attack Based on Model $M_3^{(i,j,k)}$

According to this model the induced fault affects three of the four diagonals of the state matrix. These three faulty diagonals have i, j, and k bytes modified respectively; where $1 \leq i, j, k \leq 4$. As with the previous attacks, we divide this attack into two phases. Where, as previously, the first phase is defined by Saha et al. [11].

The First Phase of the Attack Based on Model $M_3^{(i,j,k)}$. This attack is similar to the first phase of the attack based on model $M_2^{(i,j)}$. Figure 7 shows the propagation of such a fault.

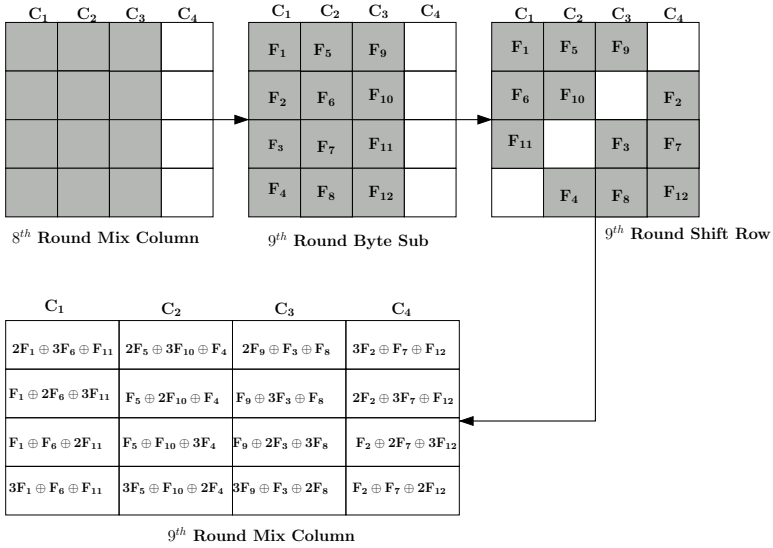

Fig. 7. Propagation of faults based on mode $M_2^{(1,1)}$

The difference in the result of the ninth round `MixColumn` can therefore be defined. For example, the left most column c_1 produces the quadruplet

$$a_1 = 2\,F_1 \oplus 3\,F_6 \oplus F_{11}, \quad a_2 = F_1 \oplus 2\,F_6 \oplus 3\,F_{11},$$
$$a_3 = F_1 \oplus 2\,F_6 \oplus 2\,F_{11}, \quad \text{and} \quad a_4 = 3\,F_1 \oplus F_6 \oplus F_{11}\ .$$

If we eliminate F_1, F_6, and F_{11} from above equations we get,

$$11\,a_1 \oplus 13\,a_2 = 9\,a_3 \oplus 14\,a_4$$

where a_1, a_2, a_3, and a_4 can be expressed as,

$$a_1 = S^{-1}(x_1 \oplus k_1) \oplus S^{-1}(x'_1 \oplus k_1), \quad a_2 = S^{-1}(x_{14} \oplus k_{14}) \oplus S^{-1}(x'_{14} \oplus k_{14})$$
$$a_3 = S^{-1}(x_{11} \oplus k_{11}) \oplus S^{-1}(x'_{11} \oplus k_{11}), \quad \text{and} \quad a_4 = S^{-1}(x_8 \oplus k_8) \oplus S^{-1}(x'_8 \oplus k_8)$$

The above equations will produce 2^{24} possible values for the quadruplet $\{k_1, k_8, k_{11}, k_{14}\}$. Similarly, solving the equivalent equations for the other columns will each return 2^{24} hypotheses, producing 2^{96} key hypotheses for the first phase of the attack.

Proposed Second Phase of the Attack Based on Model $M_3^{i,j,k}$ The model $M_3^{i,j,k}$ has three faulty diagonals and three corresponding faulty columns at the end of eighth round `MixColumn`. Let us assume that the faulty diagonals are D_x, D_y, D_z and the corresponding faulty columns are c_x, c_y, c_z. Each of these three columns will produce a system of equations similar to the second phase of the attack in Section 4.2. Therefore the key reduction process is affected by the equations formed to represent the difference produced by the affected columns c_x, c_y and c_z, in the same manner as previously described in Section 4.2. There are 20 possible instances of model $M_3^{i,j,k}$, and their effect on reducing the 2^{96} possible key hypotheses produced by the first phase of the attack is summarized in Table 2.

Table 2. Results of the proposed second phase of the attack on Model $M_3^{i,j,k}$.

Model ($M_3^{i,j,k}$)	Probability ($p_{i,j,k}$)	Key Hypotheses
$M_3^{1,1,1}$	2^{-72}	2^{24}
$M_3^{1,1,2}$	2^{-64}	2^{32}
$M_3^{1,1,3}$	2^{-56}	2^{40}
$M_3^{1,1,4}$	2^{-48}	2^{48}
$M_3^{1,2,2}$	2^{-56}	2^{40}
$M_3^{1,2,3}$	2^{-48}	2^{48}
$M_3^{1,2,4}$	2^{-40}	2^{56}
$M_3^{1,3,3}$	2^{-40}	2^{56}
$M_3^{1,3,4}$	2^{-32}	2^{64}
$M_3^{1,4,4}$	2^{-24}	2^{72}

Model ($M_3^{i,j,k}$)	Probability ($p_{i,j,k}$)	Key Hypotheses
$M_3^{2,2,2}$	2^{-48}	2^{48}
$M_3^{2,2,3}$	2^{-40}	2^{56}
$M_3^{2,2,4}$	2^{-32}	2^{64}
$M_3^{2,3,3}$	2^{-32}	2^{64}
$M_3^{2,3,4}$	2^{-24}	2^{72}
$M_3^{2,4,4}$	2^{-16}	2^{80}
$M_3^{3,3,3}$	2^{-24}	2^{72}
$M_3^{3,3,4}$	2^{-16}	2^{80}
$M_3^{3,4,4}$	2^{-8}	2^{88}
$M_3^{4,4,4}$	$--$	2^{96}

5 Experimental Result

To validate the proposed attack, and the models used, a series of experiments were conducted, which were based around an iterative AES-128 implementation using Verilog HDL in a Xilinx Spartan-3E FPGA xc3s500E device with an operating frequency of 36 MHz. Faults were injected into the eighth round input of AES by inducing clock glitches. The setup is subjected to two different input clocks;

one clock having a higher frequency than the other. The clocks are multiplexed in such a way that when the eighth round of the AES starts the device switches to a faster clock before returning to a normal clock speed. We used ChipScope Pro 7.1 analyzer to observe the faulty bytes in the state matrix of AES hardware running on the FPGA. The experiment started with a fast clock frequency set to 72 MHz. This frequency was gradually increased at the rate 0.2 MHz per step. At each step we perform 512 attempts to inject faults.

Appendix A summarizes the experimental results. The first column represents the fast clock frequency with which we attempted to inject faults. The second column gives the number of fault free samples out of 512 attempts and the subsequent columns depicts the number of faulty samples as per the corresponding fault model. The columns M_1, M_2, M_3, M_4 represent the number of faulty sample corresponding to faults affecting one, two, three, and four diagonals respectively. The rest of the columns correspond to the number of faulty samples of the 6 instances of fault models $M_1^{(i)}, M_2^{(i,j)}$, and $M_3^{(i,j,k)}$. Only those models for which we acquired at least one faulty sample were included in Appendix A.

The first fault appeared when the glitch clock frequency was set to 72.6 MHz, although only faults corresponding to model $M_1^{(1)}$ were observed with a clock speed less than 73.8 MHz. The experiment was continued with up to 80 MHz of clock frequency and we can observe that the probability of acquiring a sample belonging to a particular fault model is not uniform. Furthermore, it may be noted that the experimental findings show that the induced faults indeed belong to the desirable models (for which the proposed attacks reduce the key space to sizes which can be easily brute force searched). For example, the worst observed model is $M_3^{(1,1,3)}$, in case of which the proposed attack reduces the key space to 2^{40}. In most cases, however the faults belong to the faults $M_1^{(1)}, M_2^{(1,1)}$ and $M_2^{(1,2)}$ for which the attacks reduce the key space to expected values of $2^8, 2^{16}$ and 2^{24} respectively. Repeated experiments show that the nature and distribution of faults can be reproduced, thus showing that the attacker can suitably control the most probable faults to belong to models for which the attacks reduce the key space to practical limits. As the ratios of faults corresponding to a particular model and clock frequency can be reproduced, following the results tabulated in Appendix A an attacker can monitor the clock frequency and produce faults corresponding to a desired model with high probability.

6 Conclusion

The paper presents an differential fault analysis of AES where multiple bytes can be affected by a fault and an attacker will only require one acquisition to recover the secret key used. Experimental results have been provided to show that the practicality of the attack is improved by the techniques proposed in the paper for faults that affect multiple bytes. Not all of the attacks are practical as described since the number of key hypotheses generated is too large for a practical exhaustive search. However, the number of key hypotheses can be further reduced by acquiring more faulty ciphertexts that correspond to the same fault model. An attacker can then take the intersection of the key hypotheses generated by the first and second phase independently.

References

1. H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The sorcerer's apprentice guide to fault attacks. *Proceedings of the IEEE*, 94(2):370–382, 2006.
2. E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In B. S. Kaliski, editor, *Advances in Cryptology — CRYPTO '97*, volume 1294 of *LNCS*, pages 513–525. Springer, 1997.
3. J. Blömer and J.-P. Seifert. Fault based cryptanalysis of the advanced encryption standard (AES). In R. N. Wright, editor, *Financial Cryptography — FC 2003*, volume 2742 of *LNCS*, pages 162–181. Springer, 2003.
4. D. Boneh, R. DeMillo, and R. Lipton. On the importance of checking cryptographic protocols for faults. In W. Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 of *LNCS*, pages 37–51. Springer, 1997.

5. P. Dusart, G. Letourneux, and O. Vivolo. Differential fault analysis on A.E.S. In J. Zhou, M. Yung, and Y. Han, editors, *Applied Cryptography and Network Security — ACNS 2003*, volume 2846 of *LNCS*, pages 293–306. Springer, 2003.
6. C. Giraud. DFA on AES. In H. Dobbertin, V. Rijmen, and A. Sowa, editors, *International Conference Advanced Encryption Standard — AES 2004*, volume 3373 of *LNCS*, pages 27–41. Springer, 2004.
7. D. Mukhopadhyay. An improved fault based attack of the advanced encryption standard. In B. Preneel, editor, *Progress in Cryptology — AFRICACRYPT 2009*, volume 5580 of *LNCS*, pages 421–434. Springer, 2009.
8. National Institute of Standards and Technology (NIST). Advanced Encryption Standard (AES). FIPS Publication 197, available for download at `http://www.itl.nist.gov/fipspubs/`, 2001.
9. K. Nyberg. Differentially uniform mappings for cryptography. In T. Helleseth, editor, *Advances in Cryptology — EUROCRYPT '93*, volume 765 of *LNCS*, pages 55–64. Springer, 1993.
10. G. Piret and J.-J. Quisquater. A differential fault attack technique against SPN structure, with application to the AES and KHAZAD. In C. D. Walter, Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2003*, volume 2779 of *LNCS*, pages 77–88. Springer, 2003.
11. D. Saha, D. Mukhopadhyay, and D. RoyChowdhury. A diagonal fault attack on the Advanced Encryption Standard. Cryptology ePrint Archive, Report 2009/581, 2009. `http://eprint.iacr.org/`.
12. M. Tunstall and D. Mukhopadhyay. Differential fault analysis of the advanced encryption standard using a single fault. Cryptology ePrint Archive, Report 2009/575, 2009. `http://eprint.iacr.org/`.

A Experimental Results

Clock Frequency(MHz)	Fault free	M_1	M_2	M_3	M_4	$M_1^{(1)}$	$M_2^{(1,1)}$	$M_2^{(1,2)}$	$M_2^{(1,3)}$	$M_3^{(1,1,2)}$	$M_3^{(1,1,3)}$
72.0	512	0	0	0	0	0	0	0	0	0	0
72.2	512	0	0	0	0	0	0	0	0	0	0
72.4	512	0	0	0	0	0	0	0	0	0	0
72.6	510	2	0	0	0	2	0	0	0	0	0
72.8	511	1	0	0	0	1	0	0	0	0	0
73.0	508	4	0	0	0	4	0	0	0	0	0
73.2	504	8	0	0	0	8	0	0	0	0	0
73.4	507	5	0	0	0	5	0	0	0	0	0
73.6	490	22	0	0	0	22	0	0	0	0	0
73.8	489	23	0	0	0	23	0	0	0	0	0
74.0	419	79	14	0	0	79	14	0	0	0	0
74.2	448	60	4	0	0	60	4	0	0	0	0
74.4	437	64	14	0	0	64	13	1	0	0	0
74.6	403	94	15	0	0	94	15	0	0	0	0
74.8	408	99	5	0	0	99	5	0	0	0	0
75.0	248	226	38	0	0	226	38	0	0	0	0
75.2	214	205	93	0	0	205	84	9	0	0	0
75.4	128	205	179	0	0	205	122	57	0	0	0
75.6	76	180	256	0	0	180	133	123	0	0	0
75.8	20	122	370	0	0	122	146	224	0	0	0
76.0	158	190	163	0	0	190	129	34	0	0	0
76.2	27	116	368	0	0	116	184	184	0	0	0
76.4	40	128	344	0	0	128	197	147	0	0	0
76.6	26	68	413	5	0	68	156	257	0	4	1
76.8	17	62	391	34	8	62	138	253	0	16	18

77.0	0	20	429	47	16	20	68	361	0	23	24
77.2	0	0	336	123	53	0	16	320	0	32	91
77.4	0	2	313	101	96	2	21	292	0	31	70
77.6	0	1	298	123	90	1	9	288	1	46	77
77.8	0	12	409	71	20	12	42	367	0	42	29
78.0	15	59	415	22	1	59	107	308	0	19	3
78.2	0	2	210	160	140	2	12	198	0	62	98
78.4	0	5	365	94	48	5	26	339	0	36	58
78.6	0	4	296	126	86	4	11	285	0	50	76
78.8	0	0	133	110	269	0	0	133	0	27	83
79.0	0	0	144	112	256	0	6	138	0	20	92
79.2	0	0	150	114	248	0	0	150	0	28	86
79.4	0	0	21	20	471	0	0	21	0	4	16
79.6	0	0	18	24	470	0	0	18	0	3	21
79.8	0	0	14	21	477	0	0	14	0	2	19
80.0	0	0	0	0	512	0	0	0	0	0	0