Differential Fault Analysis of the Advanced Encryption Standard Using a Single Fault

Michael Tunstall, Debdeep Mukhopadhyay, Subidh Ali

Differential Fault Analysis of the Advanced Encryption Standard using a Single Fault

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

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

Abstract. In this paper we present a differential fault attack that can be applied to the AES using a single fault. We demonstrate that when a single random byte fault is induced at the input of the eighth round, the AES key can be deduced using a two stage algorithm. The first step has a statistical expectation of reducing the possible key hypotheses to 2^{32}, and the second step to a mere 2^8.

Keywords: Differential Fault Analysis, Fault Attack, Advanced Encryption Standard.

1 Introduction

The Advanced Encryption Standard (AES) [10] has been a de-facto standard for symmetric key cryptography since October 2000. Smart cards and secure microprocessors, therefore, typically include implementations of AES to protect the confidentiality and the integrity of sensitive information. To satisfy the high throughput requirements of such applications, these implementations are typically VLSI devices (crypto-accelerators) or highly optimized software routines (crypto-libraries).

Several applications of DFA to AES have been reported in the literature. In [3], authors describe an analysis based on faults induced in one byte of the ninth round of AES that requires 250 faulty ciphertexts. An attack reported in [1] allows an attacker to recover the secret key with around 128 to 256 faulty ciphertexts. In [2], Dusart et al. show that using a fault which affects one byte anywhere between the eighth round MixColumn and ninth round MixColumn, an attacker would be able to derive the secret key using 40 faulty ciphertexts. The authors of [12] describe an attack on AES with single byte faults that requires two faulty outputs, where a fault is induced in the input of the eighth or ninth round, extended to one 32-bit fault in the ninth round in [8].

We can note that when the assumptions are on the value of a byte (either it being faulty or uncorrupted) the number of faulty pairs is quite small. However,

it is difficult to be able to affect a given value with any certainty. When numerous faulty ciphertexts are required this problem is amplified, since an attacker needs to find a method of determining which faulty ciphertexts correspond to the desired model. We can, therefore, state that the attacks that are most likely to be realizable require the least faulty ciphertexts and assumptions on the effect of the fault.

In [9] a fault attack against AES was proposed, which suggested that a secret key can be derived using a single *byte* fault induction at the input of the eighth round. The attack exploited the inter-relations between the fault values in the state matrix after the ninth round `MixColumn` operation and reduced the number of possible keys to around 2^{32}. However it may be noted that this work, like the previous fault attacks on AES does not use the effect of the fault maximally in an information theoretic sense [7]. The work proposed in this paper improves the previous fault analysis on AES-128 and reduces the key space to its minimal possible set of hypotheses attainable using a single byte fault. In this paper, we describe the extended version of this attack, where an attacker could reduce the exhaustive search to 2^8.

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 an attack based on one of the fault models given in Section 2. In Section 3 we extend this attack. In Section 4 we compare this paper to work described in the literature, and we conclude in Section 5.

2 Background

2.1 The Advanced Encryption Standard

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 P which produces fault free (CT) and faulty ciphertexts (CT') are arranged in the following fashion

Algorithm 1: The AES-128 encryption function.

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

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

$$P = \begin{pmatrix} p_1 & p_5 & p_9 & p_{13} \\ p_2 & p_6 & p_{10} & p_{14} \\ p_3 & p_7 & p_{11} & p_{15} \\ p_4 & p_8 & p_{12} & p_{16} \end{pmatrix} \quad \mathbf{CT} = \begin{pmatrix} x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \\ x_4 & x_8 & x_{12} & x_{16} \end{pmatrix} \quad \mathbf{CT}' = \begin{pmatrix} x'_1 & x'_5 & x'_9 & x'_{13} \\ x'_2 & x'_6 & x'_{10} & x'_{14} \\ x'_3 & x'_7 & x'_{11} & x'_{15} \\ x'_4 & x'_8 & x'_{12} & x'_{16} \end{pmatrix}$$

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 $K_{10} = \{k_1, \ldots, k_{16}\}$ and $K_9 = \{k'_1, \ldots, k'_{16}\}$ that are arranged in a state matrix as described above.

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 which 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.

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**
  $x_{(i<<2)+1} \leftarrow x_{(i<<2)+1} \oplus S(x_{(((i+1)\wedge 3)<<2)+4})$
end
$x_1 \leftarrow x_1 \oplus h_r$
for $i \leftarrow 1$ **to** 16 **do**
  if $(i-1) \mod 4 \neq 0$ **then**
    $x_i \leftarrow x_i \oplus x_{i-1}$
  end
end
return X

2.2 The Fault Model

The implementation of AES we target is an iterative one, i.e. where a round function is executed in a loop as described in Algorithm 1. An attacker can typically predict at what point in time certain events take place, e.g. when a particular round commences. Moreover, the time certain events take can often be determined by analyzing a suitable side channel.

The fault model that we consider is the same as that used in many other papers, for example [9], where we assume that the effect of an induced fault is to change one byte to a random value.

For example, an attacker could attempt to use a glitch in the clock to create a fault at the input of a particular round with a certain probability. An iterative design helps in this regard, as the attacker is able to control the timing of fault induction by simply counting the number of clock edges from the start of an encryption.

3 The Fault Analysis

3.1 The First Step of the Fault Attack

If a fault is induced in a byte of the state matrix, which is then input to the eighth round, the `MixColumn` operation at the end of the round propagates this fault to the

entire column of the state. The `ShiftRow` operation at the beginning of the following round will then shift these bytes to occupy different columns. The next `MixColumn` operation will then propagate the fault to the remaining twelve bytes.

This process is shown in Figure 1 where we show the diffusion of a byte fault induced at the input of the eighth round. The XOR difference of the state matrices of the two results, one fault free and the other faulty, is shown. This is what we use as basis for a differential fault analysis.

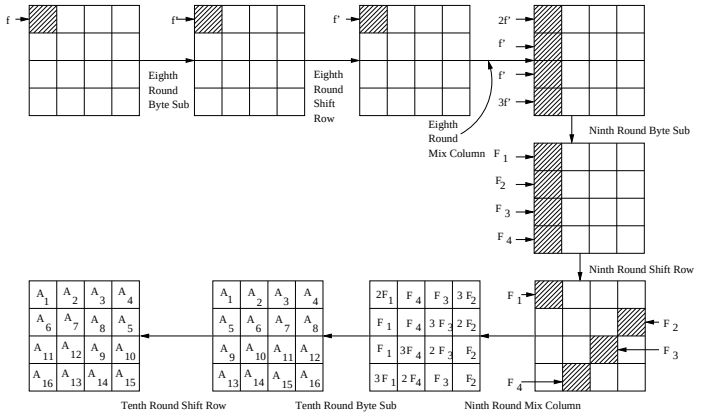

Fig. 1: Propagation of Fault Induced in the input of eighth round of AES.

If, given a fault in the input to the eighth round, we consider the state of the differences after the ninth round shift row, we can obtain the following set of equations that include the values of the key bytes k_1, k_8, k_{11} and k_{14}, thus giving an expression for 32 bits of $\mathbf{K_{10}}$.

$$\left.\begin{aligned} 2\,\delta_1 &= S^{-1}(x_1 \oplus k_1) \oplus S^{-1}(x'_1 \oplus k_1) \\ \delta_1 &= S^{-1}(x_{14} \oplus k_{14}) \oplus S^{-1}(x'_{14} \oplus k_{14}) \\ \delta_1 &= S^{-1}(x_{11} \oplus k_{11}) \oplus S^{-1}(x'_{11} \oplus k_{11}) \\ 3\,\delta_1 &= S^{-1}(x_8 \oplus k_8) \oplus S^{-1}(x'_8 \oplus k_8) \end{aligned}\right\},$$

Where δ_1, k_1, k_8, k_{11} and k_{14} are all unknown values $\in \{0, \ldots, 255\}$.

The above system of equations can be used to reduce the possibilities for these 32 bits of the key. An attacker would select a value for δ_1 and determine which values of k_1, k_8, k_{11} and k_{14} satisfy the equations using four independent exhaustive searches. Each equation will return 0, 2, or 4 hypotheses [11]. If any of the four equations cannot be satisfied, i.e. there is an impossible differential [6], then any hypotheses for that value of δ_1 can be discarded.

As noted in [4, 8] one can apply the same technique to recover information on the remaining bytes of the last sub key. That is, information on the remaining key bytes can be derived by using the following sets of equations: In order to obtain information

on k_2, k_5, k_{12} and k_{15} an attacker can use

$$\begin{aligned} 3\,\delta_2 &= S^{-1}(x_5 \oplus k_5) \oplus S^{-1}(x'_5 \oplus k_5) \\ 2\,\delta_2 &= S^{-1}(x_2 \oplus k_2) \oplus S^{-1}(x'_2 \oplus k_2) \\ \delta_2 &= S^{-1}(x_{15} \oplus k_{15}) \oplus S^{-1}(x'_{15} \oplus k_{15}) \\ \delta_2 &= S^{-1}(x_{12} \oplus k_{12}) \oplus S^{-1}(x'_{12} \oplus k_{12}) \end{aligned} .$$

In order to obtain information on k_3, k_6, k_9 and k_{16} an attacker can use the following equations:

$$\begin{aligned} \delta_3 &= S^{-1}(x_9 \oplus k_9) \oplus S^{-1}(x'_9 \oplus k_9) \\ 3\,\delta_3 &= S^{-1}(x_6 \oplus k_6) \oplus S^{-1}(x'_6 \oplus k_6) \\ 2\,\delta_3 &= S^{-1}(x_3 \oplus k_3) \oplus S^{-1}(x'_3 \oplus k_3) \\ \delta_3 &= S^{-1}(x_{16} \oplus k_{16}) \oplus S^{-1}(x'_{16} \oplus k_{16}) \end{aligned}$$

Finally, in order to obtain information on k_4, k_7, k_{10} and k_{13} an attacker can use the following equations:

$$\begin{aligned} \delta_4 &= S^{-1}(x_{13} \oplus k_{13}) \oplus S^{-1}(x'_{13} \oplus k_{13}) \\ \delta_4 &= S^{-1}(x_{10} \oplus k_{10}) \oplus S^{-1}(x'_{10} \oplus k_{10}) \\ 3\,\delta_4 &= S^{-1}(x_7 \oplus k_7) \oplus S^{-1}(x'_7 \oplus k_7) \\ 2\,\delta_4 &= S^{-1}(x_4 \oplus k_4) \oplus S^{-1}(x'_4 \oplus k_4) \end{aligned}$$

It can be noted that the equations have an identical structure, and, therefore, the solutions are of similar nature. An evaluation of each set of equations will be expected to return 2^8 unique hypotheses for the key bytes concerned. Therefore, an attacker would expect to have 2^{32} key hypotheses for the secret key used.

3.2 Analysis of the first step of the fault attack

The first step of the fault attack uses four sets of equations to reduce the key space of AES. In this section we determine the expected number of key hypotheses that an attacker will have at each stage of an attack.

In order to analyze the number of valid hypotheses in the first stage of the attack we consider the first set of equations given in Section 3.1. In this set of equations δ_1 is $\in \{1, \ldots, 255\}$. If δ_1 is equal to zero then one could say that the expected fault has not been injected. If δ_1 is zero it would imply that x_1 is equal to x'_1 and all 256 key hypotheses are possible. Let us first consider the first equation in this set:

$$2\,\delta_1 = S^{-1}(x_1 \oplus k_1) \oplus S^{-1}(x'_1 \oplus k_1)$$

We know the values of x_1 and x'_1 from the correct and faulty ciphertexts respectively. For a given value of $2\,\delta_1$ there will 0, 2 or 4 valid key hypotheses. The mean hypotheses for all $\delta_1 \in \{1, \ldots, 255\}$ is approximately one, and, therefore, 256 key hypotheses when all $\delta_1 \in \{1, \ldots, 255\}$ are considered.

The same can be said for each of the four equations in the set given above. However, for a given value of δ_1 each of the four equations would be expected to return approximately one hypothesis for a key byte. These values will give one hypothesis for the quartet of key bytes $\{k_1, k_8, k_{11}, k_{14}\}$. Given that an attacker will have to take into account all the values in $\{0, \ldots, 255\}$ there will be 256 possible values for the quartet $\{k_1, k_8, k_{11}, k_{14}\}$. After an attacker has analyzed the four equations defined in Section 3.1 there would be an expected 2^{32} key hypotheses.

3.3 The Second Step of the Fault Attack

In order to further reduce the key hypotheses we use the relationship between the ninth round key and the tenth round key.

We consider the key-scheduling algorithm (see Algorithm 2), the ninth round key, K_9, generates the tenth round key, K_{10}. The key schedule is invertible and $\mathbf{K_9}$ can be expressed in terms of elements of $\mathbf{K_{10}}$. The value of $\mathbf{K_9}$ can be expressed as

$$\begin{pmatrix} k_1 \oplus S(k_{14} \oplus k_{10}) \oplus h_{10} & k_5 \oplus k_1 & k_9 \oplus k_5 & k_{13} \oplus k_9 \\ k_2 \oplus S(k_{15} \oplus k_{11}) & k_6 \oplus k_2 & k_{10} \oplus k_6 & k_{14} \oplus k_{10} \\ k_3 \oplus S(k_{16} \oplus k_{12}) & k_7 \oplus k_3 & k_{11} \oplus k_7 & k_{15} \oplus k_{11} \\ k_4 \oplus S(k_{13} \oplus k_9) & k_8 \oplus k_4 & k_{12} \oplus k_8 & k_{16} \oplus k_{12} \end{pmatrix}.$$

We can observe that the fault values in the first column of the state matrix at the output of the eighth round `MixColumn` is $(2\,f', f', f', 3\,f')$, where f' is a non-zero arbitrary value in $\mathbb{F}_{2^8}$. Using the `InverseMixColumn` operation and using the inter-relations between the fault values, we can define the following equation:

$$\begin{aligned} 2\,f' = S^{-1}\Big(&14\left(S^{-1}(x_1 \oplus k_1) \oplus ((k_1 \oplus S(k_{14} \oplus k_{10}) \oplus h_{10}))\right) \oplus 11\left(S^{-1}(x_8 \oplus k_8) \oplus \right. \\ &\left.(k_2 \oplus S(k_{15} \oplus k_{11}))\right) \oplus 13\left(S^{-1}(x_{11} \oplus k_{11}) \oplus (k_3 \oplus S(k_{16} \oplus k_{12}))\right) \oplus \\ &9\left(S^{-1}(x_8 \oplus k_8) \oplus (k_4 \oplus S(k_{13} \oplus k_9))\right)\Big) \oplus S^{-1}\Big(14\left(S^{-1}(x'_1 \oplus k_1)\right. \\ &\left. \oplus ((k_1 \oplus S(k_8 \oplus k_{10}) \oplus h_{10}))\right) \oplus 11\left(S^{-1}(x'_8 \oplus k_8) \oplus (k_2 \oplus S(k_{15} \oplus k_{11})\right) \oplus \\ &13\left(S^{-1}(x'_{11} \oplus k_{11}) \oplus (k_3 \oplus S(k_{16} \oplus k_{12}))\right) \oplus 9\left(S^{-1}(x'_8 \oplus k_8) \oplus \right. \\ &\left.(k_4 \oplus S(k_{13} \oplus k_9))\right)\Big) \end{aligned}$$

Similarly, we can define the following equations:

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

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

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

The second stage of the attack is coupled with the first stage, and can be used to further reduce the number of key hypotheses.

3.4 Analysis of the second step of the fault attack

The expected number of hypotheses produced by the second step of the attack follows a similar reasoning to the analysis of the first step, given in Section 3.2.

If we consider the second equation defined in Section 3.3, it can be rewritten as

$$
f' = A \oplus B\,,
$$

where A and B are defined as

$$
\begin{aligned}
A = S^{-1}\Big(\mathtt{9}\left(S^{-1}(x_{13} \oplus k_{13}) \oplus (k_{13} \oplus k_9)\right) \oplus \\
\mathtt{14}\left(S^{-1}(x_{10} \oplus k_{10}) \oplus (k_{10} \oplus k_{14})\right)\Big) \oplus \mathtt{11}\left(S^{-1}(x_7 \oplus k_7) \oplus \\
(k_{15} \oplus k_{11})\right) \oplus \mathtt{13}\left(S^{-1}(x_4 \oplus k_4) \oplus (k_{16} \oplus k_{12})\right)\Big)
\end{aligned}
$$

and

$$
\begin{aligned}
B = S^{-1}\Big(\mathtt{9}\left(S^{-1}(x'_{13} \oplus k_{13}) \oplus (k_{13} \oplus k_9)\right) \oplus \\
\mathtt{14}\left(S^{-1}(x'_{10} \oplus k_{10}) \oplus (k_{10} \oplus k_{14})\right)\Big) \oplus \mathtt{11}\left(S^{-1}(x'_7 \oplus k_7) \oplus\ .\right. \\
\left.(k_{15} \oplus k_{11})\right) \oplus \mathtt{13}\left(S^{-1}(x'_4 \oplus k_4) \oplus (k_{16} \oplus k_{12})\right)\Big)
\end{aligned}
$$

We can consider A and B to be random values in $\mathbb{F}_{2^8}$. For a given values of f' the difference between A and B will be equal to f' with a probability of $\frac{1}{2^8}$. Using the same reasoning, the probability of all four equations being valid is $\left(\frac{1}{2^8}\right)^4 = \frac{1}{2^{32}}$.

We have to consider all the possible values of f', i.e. $\{0, \ldots, 255\}$. A given key hypothesis will, therefore, be valid for some arbitrary value of f' with a probability of $2^8 \times \frac{1}{2^{32}} = \frac{1}{2^{24}}$. The first step of the attack is expected to return 2^{32} hypotheses each of which still be under consideration at the end of the second step with a probability of $\frac{1}{2^{24}}$. One would, therefore, expect the second step of the attack to produce 2^8 possible key hypotheses.

3.5 Attacking Other Bytes

In the previous sections we describe an attack where we base our Differential Fault Analysis on the knowledge that a fault has been induced in the first byte of the state matrix. However, we can note that the analysis returns a very small number of hypotheses. We can, therefore, conduct 16 independent analyses under the assumption that a fault is induced each of the 16 bytes of of the state at the beginning of the eighth round. An attacker would expect this to produce $2^4 \times 2^8 = 2^{12}$ valid key hypotheses, which is still a trivial exhaustive search.

4 Comparison with Previous Work

There are several versions of fault-based differential cryptanalysis that are able to reduce the number of key hypotheses from two faults injected into an implementation of AES, as described in [5, 9, 12]. However, the analysis proposed in this paper is more effective, since the resulting exhaustive search can be reduced to a trivial size using one fault. The number of key hypotheses returned by previous work would be somewhat time consuming. The advantage of the proposed attack is that it does not need to reproduce a successful attack in order to able to determine a secret key. Acquiring multiple faulty ciphertexts can be problematic as faults are only successful with a certain probability, and the effect cannot always be predetermined. This would mean that an attacker could potentially have to search among numerous faulty ciphertexts to find a pair that both have the desired fault.

5 Conclusion

This paper proposes a fault-based differential cryptanalysis of AES, that is an extended version of the attack described in [9]. An attacker would expect to be able to reduce the number of key hypotheses from 2^{128} to 2^8 with one well placed fault. As noted in [8], these attacks can be conducted without any knowledge of the plaintext being enciphered, as an attacker would just need to know the plaintexts were the same.

There are many descriptions of a fault-based differential cryptanalysis of AES that could be prevented by repeating the last two or three rounds of an implementation of AES, to verify that no exploitable fault has been inserted [1–3, 12, 13]. However, to prevent the attack described in this paper the last four rounds would need to be repeated to check no fault was injected. Moreover, given how much information can be gleaned from one fault, one would expect there are attacks that require more faulty ciphertexts that would be able to make use of faults in earlier rounds. One would, therefore, suggest that in order to protect an implementation of AES the last five rounds should be protected against fault injection.

Acknowledgements

The work described in this paper has been supported in part by the European Commission IST Programme under Contract ICT-2007-216676 ECRYPT II and EPSRC grant EP/F039638/1 “Investigation of Power Analysis Attacks”. The second author would like to acknowledge the support of Department of Science and Technology (DST) India under the Fast Track Proposals for Young Scientists for the proposal entitled ”Design and Analysis of Side Channel Attack Resistant Symmetric Key Cryptosystems”.‘

References

1. 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 *Lecture Notes in Computer Science*, pages 162–181. Springer-Verlag, 2003.

2. 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 *Lecture Notes in Computer Science*, pages 293–306. Springer-Verlag, 2003.
3. C. Giraud. DFA on AES. In H. Dobbertin, V. Rijmen, and A. Sowa, editors, *International Conference Advanced Encryption Standard — AES 2004*, volume 3373 of *Lecture Notes in Computer Science*, pages 27–41. Springer-Verlag, 2004.
4. C. Giraud and A. Thillard. Piret and Quisquater's DFA on AES revisited. Cryptology ePrint Archive, Report 2010/440, 2010. `http://eprint.iacr.org/`.
5. C. H. Kim and J.-J. Quisquater. New differential fault analysis on aes key schedule: Two faults are enough. In G. Grimaud and F.-X. Standaert, editors, *Smart Card Research and Advanced Applications — CARDIS 2008*, volume 5189 of *Lecture Notes in Computer Science*, pages 48–60. Springer-Verlag, 2008.
6. L. Knudsen. Deal — a 128-bit block cipher. Technical report no. 151. Department of Informatics, University of Bergen, Norway, 1998.
7. Yang Li, Shigeto Gomisawa, Kazuo Sakiyama, and Kazuo Ohta. An information theoretic perspective on the differential fault analysis against aes. Cryptology ePrint Archive, Report 2010/032, 2010. `http://eprint.iacr.org/`.
8. A. Moradi, M. T. Manzuri Shalmani, and M. Salmasizadeh. A generalized method of differential fault attack against AES cryptosystem. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems — CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 91–100. Springer-Verlag, 2006.
9. D. Mukhopadhyay. An improved fault based attack of the advanced encryption standard. In B. Preneel, editor, *Progress in Cryptology — AFRICACRYPT 2009*, volume 5580 of *Lecture Notes in Computer Science*, pages 421–434. Springer-Verlag, 2009.
10. 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.
11. K. Nyberg. Differentially uniform mappings for cryptography. In T. Helleseth, editor, *Advances in Cryptology — EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 55–64. Springer-Verlag, 1993.
12. 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 *Lecture Notes in Computer Science*, pages 77–88. Springer-Verlag, 2003.
13. J. Takahashi, T. Fukunaga, and K. Yamakoshi. DFA mechanism on the AES schedule. In *Fault Diagnosis and Tolerance in Cryptography 2007 — FDTC 07*, pages 62–72, 2007.