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

Subidh Ali[1] and Debdeep Mukhopadhyay[1]

Department of Computer Sc. and Engg, IIT Kharagpur, West Bengal, India.
{subidh,debdeep}@cse.iitkgp.ernet.in

Abstract. In this paper we present a speed up of the existing fault attack [2] on the Advanced Encryption Standard (AES) using single faulty cipher. The paper suggests a parallelization technique to reduce the complexity of the attack from 2^{32} to 2^{30}.

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

1 Introduction

In [1] the author reduced AES key space to 2^{32} by injecting a single byte fault at the eighth round input. This work pointed out that the fault attack against AES can be performed with a single byte fault. This idea was made more realistic in the work of [2] which proposed a method to reduce the AES key space to a mere 2^8 values using a single instance of a byte fault.

This paper revisits the work done in [2] and further reduce the complexity of the attack. The attack proposed in [2] uses a two phase attack algorithm: the first phase, which is an adoption of [1], reduces the AES key space to an expected value of 2^{32} using a single instance of a byte fault induced at the input of the eighth round. In the second phase, these reduced key space is further filtered by taking into account the relation between the tenth and ninth round keys of AES as per the key schedule of AES. The final remaining key space is 2^8 values. In this work, we show that the second step of the above attack can be parallelized, exploiting a property in the equations involved in the relations between the fault values and the ninth and tenth round keys of AES. This parallelization helps in working with a smaller key space of around 2^{30} in place of 2^{32}, as performed in [2].

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.

2 Existing Fault Analysis

In this section we describe the most recent fault attack developed in this class [2]. The attack reduces the key space of AES to 2^8 using an algorithm of time complexity 2^{32}. The algorithm uses a two phase method and is based on the assumption that a random byte fault is injected at the input of the eighth round of AES. The elegance of the attack lies in the fact that it needs only one faulty ciphertext to retrieve the key.

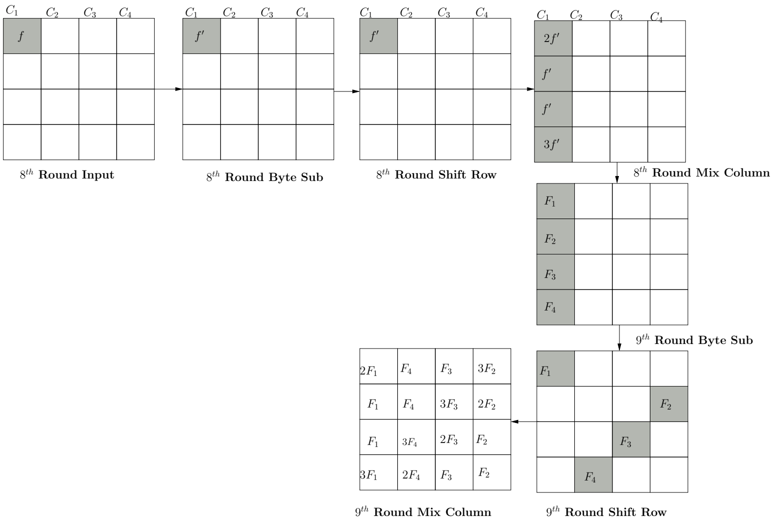

Fig. 1: Propagation of byte fault induced at the input of eighth round

2.1 First Phase of the Attack

Figure 1 shows the propagation of single byte fault induced at the input of the eighth round of AES.

At the ninth round output, these faults are propagated to rest of the twelve bytes. Let us consider CT and CT' be the fault free and faulty output ciphertext where CT is represented by,

$$\mathbf{K_{10}} = \begin{pmatrix} k_0 & k_4 & k_8 & k_{12} \\ k_1 & k_5 & k_9 & k_{13} \\ k_2 & k_6 & k_{10} & k_{14} \\ k_3 & k_7 & k_{11} & k_{15} \end{pmatrix} \qquad \mathbf{K_9} = \begin{pmatrix} k'_0 & k'_4 & k'_8 & k'_{12} \\ k'_1 & k'_5 & k'_9 & k'_{13} \\ k'_2 & k'_6 & k'_{10} & k'_{14} \\ k'_3 & k'_7 & k'_{11} & k'_{15} \end{pmatrix}$$

All the elements of CT, CT' and K_{10} are in F_{2^8}.

If we use the inter-relation between the faulty bytes of column c_1 at the end of the ninth round `MixColumn` as in Figure 1, we will have following four equations:

$$\begin{aligned} 2F_1 &= S^{-1}(x_0 \oplus k_0) \oplus S^{-1}(x'_0 \oplus k_0) \\ F_1 &= S^{-1}(x_{13} \oplus k_{13}) \oplus S^{-1}(x'_{13} \oplus k_{13}) \\ F_1 &= S^{-1}(x_{10} \oplus k_{10}) \oplus S^{-1}(x'_{10} \oplus k_{10}) \\ 3\,F_1 &= S^{-1}(x_7 \oplus k_7) \oplus S^{-1}(x'_7 \oplus k_7) \end{aligned} \tag{1}$$

This system equations yield 2^8 expected values of key quartet $\{k_0, k_{13}, k_{10}, k_7\}$. Similarly solving the other three system of equations generated from columns c_2, c_3, and c_4 will give other three key quartet values. Therefore the entire AES key space reduces to an expected value of $(2^8)^4 = 2^{32}$.

In the second phase of the attack these 2^{32} possible keys are further filtered out using the inter-relations of the fault values after the eighth round `MixColumn`, which are not used in the above equations.

2.2 Second Phase of the Attack

In order to further reduce the key hypotheses the relationship between the ninth round key and the tenth round key is used.

The AES key schedule is invertible. Therefore ninth round key $\mathbf{K_9}$ can be expressed in terms of elements of tenth round key $\mathbf{K_{10}}$. The value of $\mathbf{K_9}$ can be expressed as

$$\begin{pmatrix} k_0 \oplus S(k_{1}3 \oplus k_9) \oplus h_{10} & k_4 \oplus k_0 & k_8 \oplus k_4 & k_{12} \oplus k_8 \\ k_1 \oplus S(k_{14} \oplus k_{10}) & k_5 \oplus k_1 & k_9 \oplus k_5 & k_{13} \oplus k_9 \\ k_2 \oplus S(k_{15} \oplus k_{11}) & k_6 \oplus k_2 & k_{10} \oplus k_6 & k_{14} \oplus k_{10} \\ k_3 \oplus S(k_{12} \oplus k_8) & k_7 \oplus k_3 & k_{11} \oplus k_7 & k_{15} \oplus k_{11} \end{pmatrix} .$$

From Figure 1, we can observe that the fault values in the first column of the state matrix at the output of the eighth round `MixColumn` is $(2f', f', f', 3f')$, 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:

$$\begin{aligned} 2\,f' ==\, & S^{-1}\Big(14\,\big(S^{-1}(x_0 \oplus k_0) \oplus k_0 \oplus S(k_{1}3 \oplus k_9) \oplus h_{10}\big) \oplus 11\big(S^{-1}(x_{13} \oplus k_{13}) \oplus k_1 \oplus S(k_{14} \oplus k_{10})\big) \oplus \\ & 13\,\big(S^{-1}(x_{10} \oplus k_{10}) \oplus k_2 \oplus S(k_{15} \oplus k_{11})\big) \oplus 9\,\big(S^{-1}(x_7 \oplus k_7) \oplus k_3 \oplus S(k_{12} \oplus k_8)\big)\Big) \oplus \\ & S^{-1}\Big(14\,\big(S^{-1}(x'_0 \oplus k_0) \oplus k_0 \oplus S(k_{1}3 \oplus k_9) \oplus h_{10}\big) \oplus 11\big(S^{-1}(x'_{13} \oplus k_{13}) \oplus k_1 \oplus S(k_{14} \oplus k_{10})\big) \oplus \\ & 13\,\big(S^{-1}(x'_{10} \oplus k_{10}) \oplus k_2 \oplus S(k_{15} \oplus k_{11})\big) \oplus 9\,\big(S^{-1}(x'_7 \oplus k_7) \oplus k_3 \oplus S(k_{12} \oplus k_8)\big)\Big) \end{aligned} \tag{2}$$

Similarly, we can define the following equations:

$$\begin{aligned} f' =\, & S^{-1}\Big(9\,\big(S^{-1}(x_{12} \oplus k_{12}) \oplus (k_{12} \oplus k_8)\big) \oplus 14\,\big(S^{-1}(x_9 \oplus k_9) \oplus (k_{13} \oplus k_9))\big) \oplus 11\,\big(S^{-1}(x_6 \oplus k_6) \oplus \\ & (k_{14} \oplus k_{10})\big) \oplus 13\big(S^{-1}(x_3 \oplus k_3) \oplus (k_{15} \oplus k_{11})\big)\Big) \oplus S^{-1}\Big(9\,\big(S^{-1}(x'_{12} \oplus k_{12}) \oplus (k_{12} \oplus k_8)\big) \oplus \\ & 14\,\big(S^{-1}(x'_9 \oplus k_9) \oplus (k_{13} \oplus k_9))\big) \oplus 11\,\big(S^{-1}(x'_6 \oplus k_6) \oplus (k_{14} \oplus k_{10})\big) \oplus 13\,\big(S^{-1}(x'_3 \oplus k_3) \oplus (k_{15} \oplus k_{11})\big)\Big) \end{aligned} \tag{3}$$

$$\begin{aligned} f' =\, & S^{-1}\Big(13\,\big(S^{-1}(x_8 \oplus k_8) \oplus (k_8 \oplus k_4)\big) \oplus 9\,\big(S^{-1}(x_5 \oplus k_5) \oplus (k_9 \oplus k_5))\big) \oplus 14\,\big(S^{-1}(x_2 \oplus k_2) \\ & \oplus (k_{10} \oplus k_6)\big) \oplus 11\,\big(S^{-1}(x_{15} \oplus k_{15}) \oplus (k_{11} \oplus k_7)\big)\Big) \oplus S^{-1}\Big(13\,\big(S^{-1}(x'_8 \oplus k_8) \oplus (k_8 \oplus k_4)\big) \oplus \\ & 9\,\big(S^{-1}(x'_5 \oplus k_5) \oplus (k_9 \oplus k_5))\big) \oplus 14\,\big(S^{-1}(x'_2 \oplus k_2) \oplus (k_{10} \oplus k_6)\big) \oplus 11\,\big(S^{-1}(x'_{15} \oplus k_{15}) \oplus (k_{11} \oplus k_7)\big)\Big) \end{aligned} \tag{4}$$

$$\begin{aligned} 3\,f' =\, & S^{-1}\Big(11\,\big(S^{-1}(x_4 \oplus k_4) \oplus (k_4 \oplus k_{10})\big) \oplus 13\,\big(S^{-1}(x_1 \oplus k_1) \oplus (k_5 \oplus k_1))\big) \oplus 9\,\big(S^{-1}(x_{14} \oplus k_{14}) \oplus \\ & (k_6 \oplus k_2)\big) \oplus 14\,\big(S^{-1}(x_{11} \oplus k_{11}) \oplus (k_7 \oplus k_3)\big)\Big) \oplus S^{-1}\Big(11\,\big(S^{-1}(x'_4 \oplus k_4) \oplus (k_4 \oplus k_{10})\big) \oplus \\ & 13\,\big(S^{-1}(x'_1 \oplus k_1) \oplus (k_5 \oplus k_1))\big) \oplus 9\,\big(S^{-1}(x'_{14} \oplus k_{14}) \oplus (k_6 \oplus k_2)\big) \oplus 14\,\big(S^{-1}(x'_{11} \oplus k_{11}) \oplus (k_7 \oplus k_3)\big)\Big) \end{aligned} \tag{5}$$

The second stage of the attack is coupled with the first stage. All of the key hypotheses generated by the first stage are tested using the above equations. If it the key value satisfies the above equations, we store the key, else it can be discarded. This avoids storing the entire 2^{32} possible AES keys which are produced by the first step of the attack, and an attacker would expect to reduce 2^8 key hypotheses. But the complexity remain 2^{32}

In the next section we present our parallelization technique to reduce the complexity of the attack to 2^{30} from 2^{32}.

3 Parallelization of the Second Phase of the Attack

The above second phase of the analysis is based on four equations: (2), (3), (4), and (5). All the 2^{32} possible key hypotheses are tested by these four equations. The key hypotheses which are satisfied by all four equation are considered and rest are discarded.

However if we consider the above four equations in pairs we observe that each possible pair does not contain all the 16 bytes of the AES key. For example, the pair of equations (3) and (4) contains 14 key bytes excluding k_0 and k_1. This fact can be utilized to reduce the time complexity of the attack. We use this observation to split the lists of key which are exported in the first phase of the attack and subsequently filtered in the second phase.

In the first phase of the attack we have four quartets $\{k_0, k_{13}, k_{10}, k_7\}$, $\{\ k_{12}, k_9, k_6, k_3\ \}$, $\{k_8, k_5, k_2, k_{15}\}$ and $\{k_4, k_1, k_{14}, k_{11}\}$. Let us assume one value of the first quartet is (a_1, b_1, c_1, d_1). As per the property of the S-Box there will be another value of k_1 which satisfies the system of equation (1) with rest of the key byte values remaining same. Let us assume the second value of k_1 is a_2, then the four-tuple (a_2, b_1, c_1, d_1) also satisfies the system of equation (1).

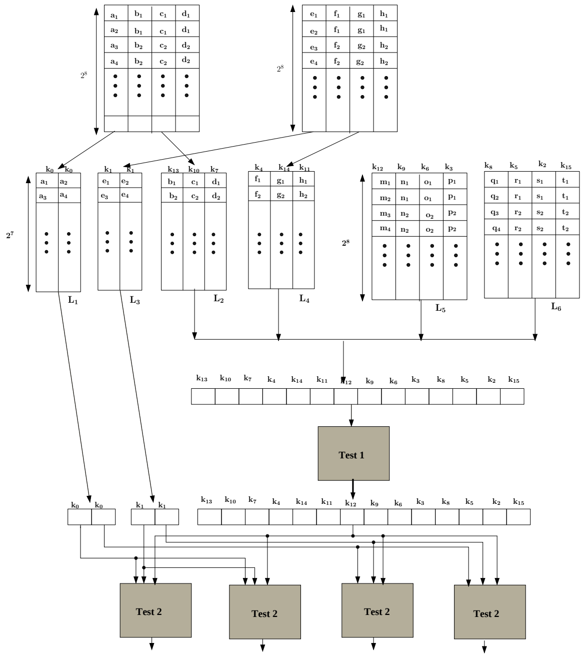

Fig. 2: Model for data-flow parallelization in the second phase

Using this idea, we can divide the list for the quartet $\{k_0, k_{13}, k_{10}, k_7\}$ into two sublists, L_1, L_2. As depicted in Figure 2 The list L_1 contains the pair values for the key byte k_0 (note that the key byte k_0 has always an even number of possible choices). The list L_2 contains the distinct values for the remaining part of the quartet, $\{k_{13}, k_{10}, k_7\}$. Thus the expected size of the lists L_1 and L_2 is 2^7 each, compared to the previous list size of 2^8 when$\{k_0, k_{13}, k_{10}, k_7\}$ were stored together.

Similarly, we store the possible values of quartet $\{k_4, k_1, k_{14}, k_{11}\}$ in two lists, L_3 and L_4. Here L_3 stores the pair values for the key byte k_5, while the list L_4 contains the distinct values for the key bytes $\{k_4, k_{14}, k_{11}\}$. Here also the expected size of the lists are 2^7. The other two quartets $\{\ k_{12}, k_9, k_6, k_3\ \}$, $\{k_8, k_5, k_2, k_{15}\}$ are stored in list L_5 and L_6. Both the lists have expected size of 2^8.

Next we select the key bytes from the six lists, $L_1, L_2, L_3, L_4, L_5, L_6$ to solve the equations of the second phase of the attack such that the time complexity is reduced.

Because of the observations regarding the pair of equations (2) and (5); and (3) and (4), the second phase can be divided into two parts. In part one we test the keys generated from the first phase of the attack by the pair of equations (3) and (4). In Figure 2 this is denoted as *Test1*. As the two equations for *Test1* does not required key bytes k_0 and k_1 we only consider all possible keys generated from lists L_2, L_4, L_5, L_6. There are 2^{30} such possible keys. In the second part we, combine each of the 14 byte keys satisfying *Test1* with one of the four possible values arising out of the four combination of the pair of values for k_0 in L_1 and k_1 in L_3. These keys are further tested in parallel by the equations (2) and (5). In Figure 2 we refer to this test as *Test2*.

The size of the lists L_2 and L_4 is 2^7; and the size of lists L_5 and L_6 is 2^8. Therefore the number of possible keys generated from this four lists is $2^7 \times 2^7 \times 2^8 \times 2^8 = 2^{30}$. These 2^{30} keys are fed as input to *Test1* which is expected to reduce the key hypotheses by 2^8. Therefore each instance of *Test2* will receive input of $(\frac{2^{30}}{2^8}) = 2^{22}$ expected key hypotheses. The chance of each key satisfying *Test2* is 2^{-16} which implies each instance of *Test2* will result in 2^6 key hypotheses.

The above attack procedure is summarized in Algorithm 1.

```
Algorithm 1: Parallelized Fault Attack on AES
    Input: 128 bit faulty and fault free ciphertexts $C$ and $C'$
    Output: $2^8$ possible key hypotheses
    Step 1: Produce four lists storing $\{k_0, k_{13}, k_{10}, k_7\}$, $\{ k_{12}, k_9, k_6, k_3 \}$, $\{k_8, k_5, k_2, k_{15}\}$ and $\{k_4, k_1, k_{14}, k_{11}\}$
    Step 2: Store the key bytes in six lists $L_1$ to $L_6$ (as mentioned before).
    Step 3: Each of the possible 14 byte keys generated by combining list $L_2, L_4, L_5$, and $L_6$, is tested by the equations (3) and (4) (Test1).
    Step 4: Each of the 14 byte keys satisfying Test1 is combined with four possible $k_0, k_1$ pair values taken from four columns of the lists $L_1$ and $L_3$.
    Step 5: Run in parallel the four instances of Test2 (check equations (2) and (5)) each with one of the four 16 byte keys of Step 4 as a input.
```

It may be easily observed that the time required is because of step 3, which is equal to 2^{30}.

4 Conclusion

The work in this paper improves the previous time complexity of the attacks [2] from 2^{32} to 2^{30} using a parallelization technique thereby making the attack four times faster on an average.

References

1. Debdeep Mukhopadhyay. An improved fault based attack of the advanced encryption standard. In Bart Preneel, editor, *AFRICACRYPT*, volume 5580 of *Lecture Notes in Computer Science*, pages 421–434. Springer, 2009.
2. Michael Tunstall and Debdeep Mukhopadhyay. Differential fault analysis of the advanced encryption standard using a single fault. Cryptology ePrint Archive, Report 2009/575, 2009. `http://eprint.iacr.org/`.