Conference Paper

Improved related-key impossible differential attacks on 8-round AES-256

Author(s): Soleimany, Hadi; Sharifi, Alireza; Aref, Mohammadreza

Publication Date: 2010

Permanent Link: https://doi.org/10.3929/ethz-a-005997213

Rights / License: In Copyright - Non-Commercial Use Permitted
Improved Related-Key Impossible Differential Attacks on 8-Round AES-256

Hadi Soleimany, Alireza Sharifi, Mohammadreza Aref
EE Department, Sharif University of Technology, Tehran, Iran
E-mail: hadi.soleimany@gmail.com, asharifi@alum.sharif.edu, aref@sharif.edu

Abstract—In this paper, we propose two new related-key impossible differential attacks on 8-round AES-256, following the work of Zhang, et al. First, we propose a carefully chosen relation between the related keys, which can be extended to 8-round subkey differences. Then, we construct a 5.5-round related-key impossible differential. Using the differential, we present two new attacks on the 8-round AES-256 with 32 and 64 bit structures. Our 8-round AES-256 attacks leads to the best known attack on AES-256 with 2 related keys. The time complexity of the proposed related-key impossible differential attacks on 8-round AES-256 is $2^{122.5}$ and its data complexity is $2^{124.5}$.

Keywords: AES-256, related-key differentials, cryptanalysis, impossible differential

I. INTRODUCTION

Rijndael [1] is an iterated block cipher with variable key and block lengths of 128 to 256 bits in steps of 32 bits. Rijndael versions with a block length of 128 bits, and key lengths of 128, 192 and 256 bits have been adopted as the Advanced Encryption Standard (AES). Differential cryptanalysis [2] analyzes the evolution of the difference between a pair of plaintexts in the following round outputs (differentials) in an iterated block cipher. The basic idea of impossible differential attack is to look for differentials that hold with probability 0 (or impossible differentials) to eliminate wrong keys and keep the right key. Related-key attacks [3], concentrate on the information which can be obtained from two encryptions using related (but unknown) keys. Related-key impossible differential attack [4] combines related-key attack and impossible differential cryptanalysis to make the attack more efficient.

The first impossible differential attack against AES was applied to 5 rounds of the AES-128 by Biham and Keller [5]. In [4], the first related-key impossible differential attack on 192-bit variants was proposed. Zhang, et. al applied three new related-key impossible differential attacks on 8-round AES-192 [6] and AES-256 [7] and concluded AES-256 has better resistance than AES-192 using the same cryptanalytic approach [7]. In this paper, we show that 8 round AES-256 can be attacked more efficient than 8 round AES-192 from overall complexity. We present 2 related-key impossible differential attacks on 8-round AES-256 with 2 related keys. Our 8-round AES-256 attacks leads to the best known attack on 8-round AES-256 with 2 related keys.

The paper is organized as follows: In Section II we briefly describe the AES algorithm. A new related-key impossible differential property of the AES-256 is introduced in Section III. In Section IV, using 64-bit structures, we propose a related-key impossible differential attack on the 8-round AES-256. In Section V we compare the performance of our attacks with the previous ones.

II. A BRIEF DESCRIPTION OF AES

In AES [1] a 128-bit plaintext is represented by a $4 \times 4$ matrix of bytes, where each byte represents a value in $GF(2^8)$. An AES round is composed of four operations: SubBytes (SB), ShiftRows (SR), MixColumns (MC) and AddRoundKey (AK). The MixColumns operation is omitted in the last round and an initial key addition is performed before the first round for whitening. We also assume that the MixColumns operation is omitted in the last round of the reduced-round variants. The number of rounds is variable depending on the key length, 10 rounds for 128-bit key, 12 for 192-bit key and 14 for 256-bit key.

A. Notations

In this paper we use the following notations: $X_i^j$ denotes the input block of round $i$, while $X_i^S$, $X_i^R$, $X_i^M$ and $X_i^O$ denotes intermediate values after applying SubBytes, ShiftRows, MixColumns and AddRoundKey operations of round $i$, respectively. Obviously, $X_i^{O,i-1} = X_i^j$ holds for $i \geq 2$. We denote the subkey of the $i$-th round by $k_i$ and the initial whitening subkey by $k_0$. In some cases, we are interested in interchanging the order of the MixColumns operation and the Subkey Addition. As these operations are linear, they can be interchanged, first XORing the data with an equivalent key and then applying the MixColumns operation. We denote equivalent subkey for the modified version by $w_i$, i.e. $w_i = MC^{-1}(k_i)$, and $X_i^W$ denotes the intermediate value after applying AddRoundKey with equivalent subkey. Let $X_{i,col}(j)$ denotes the $j$-th column of $x_i$ where $j \in \{0,1,2,3\}$. We also denote the byte in the $m$-th row and $n$-th column of $X_i$ by $X_{i,m,n}$ where $m, n \in \{0,1,2,3\}$. Another notation for bytes of $x_i$ is an enumeration $\{0,1,2,\ldots,15\}$ where the byte $X_{i,m,n}$ corresponds to byte $4n + m$ of $X_i$.

\textsuperscript{0}This work was partially supported by Iran Telecommunications Research Center and the cryptography chair of the Iranian NSF.
III. 5.5-ROUND RELATED-KEY IMPOSSIBLE DIFFERENTIAL PROPERTY OF AES-256

In this paper, using a property of MixColumns operation, we propose a new 5.5-round related-key impossible differential property which our attack is based on. First of all we use the following definitions: A byte which has different values (nonzero difference) in a pair is called an active byte while passive byte is a byte with zero difference in a pair. Now we state and prove the MixColumns property:

**Theorem 3.1:** A pair of columns at the input of MixColumns operation which contains two passive bytes cannot lead to two passive bytes and one or two active bytes within the output column.

*Proof:* Suppose that \( \Delta X = (\Delta X_1, \Delta X_2, \Delta X_3, \Delta X_4) \) is the difference of input column and \( \Delta Y = (\Delta Y_1, \Delta Y_2, \Delta Y_3, \Delta Y_4) \) is the corresponding output difference. Using MixColumns operation we have:

\[
\begin{align*}
\Delta Y_1 &= 02 \cdot \Delta X_1 + 03 \cdot \Delta X_2 + 01 \cdot \Delta X_3 + 01 \cdot \Delta X_4 \\
\Delta Y_2 &= 01 \cdot \Delta X_1 + 02 \cdot \Delta X_2 + 03 \cdot \Delta X_3 + 01 \cdot \Delta X_4 \\
\Delta Y_3 &= 01 \cdot \Delta X_1 + 01 \cdot \Delta X_2 + 02 \cdot \Delta X_3 + 02 \cdot \Delta X_4 \\
\Delta Y_4 &= 03 \cdot \Delta X_1 + 01 \cdot \Delta X_2 + 01 \cdot \Delta X_3 + 02 \cdot \Delta X_4
\end{align*}
\]

where "\(^{\circ}\)" is modular multiplication of Rijndael [1]. Without loss of generality, suppose \( X_1 \) and \( X_2 \) are two passive bytes, i.e. \( \Delta X_1 = \Delta X_2 = 0 \), we would have:

\[
\begin{align*}
\Delta Y_1 &= 01 \cdot \Delta X_3 + 01 \cdot \Delta X_4 \\
\Delta Y_2 &= 03 \cdot \Delta X_3 + 01 \cdot \Delta X_4 \\
\Delta Y_3 &= 02 \cdot \Delta X_3 + 03 \cdot \Delta X_4 \\
\Delta Y_4 &= 01 \cdot \Delta X_3 + 02 \cdot \Delta X_4
\end{align*}
\]

So if two bytes of output column, for example \( Y_1 \) and \( Y_2 \) have zero difference, i.e. \( \Delta Y_1 = \Delta Y_2 = 0 \), we will have the following system of equations:

\[
\begin{align*}
01 \cdot \Delta X_3 + 01 \cdot \Delta X_4 &= 0 \\
03 \cdot \Delta X_3 + 01 \cdot \Delta X_4 &= 0
\end{align*}
\]

It is obvious that the only solution of the above system is \( \Delta X_3 = \Delta X_4 = 0 \) and consequently \( \Delta Y_3 = \Delta Y_4 = 0 \), i.e. the output column cannot have one or two active bytes.

Consider the difference between two related keys as follows:

\[
\Delta K = K_1 \oplus K_2 = [(a,0,0,0),(a,0,0,0),(a,0,0,0),(a,0,0,0)++]
\]

where \( ^{a^+} \) is known nonzero value and \( ^{?} \) denotes any value.

Such a difference results in the round subkey differences as shown in Table I.

Using the above subkey differences and Theorem 3.1, we build a 5.5-round related-key impossible differential with probability equal to 1. The 5.5-round related-key impossible differential is:

\[
\begin{align*}
\Delta X_1^{M} &= ((0,? ,0 ,?),(?,? ,? ,?),(0 ,?,? ,?),(?,0 ,0 ,?)) \\
\Delta X_2^{O} &= ((a ,0 ,0 ,0),(a ,0 ,0 ,0),(0 ,0 ,0 ,0),(0 ,0 ,0 ,0))
\end{align*}
\]

where \( ^{a} \) is a known nonzero value and \( ^{?} \) denotes any value.

Let \( \Delta X_1^{M} = ((0,? ,0 ,?),(?,? ,? ,?),(0 ,?,? ,?),(?,0 ,0 ,?)) \). From Table I, \( \Delta k_1 \) is zero and it results in \( \Delta X_2^{O} = \Delta X_2^{M} \oplus \Delta k_1 = ((0,?,0,?),(?,0,0,0),(0,?,0,?),(? ,0 ,0 ,?)) \) which leads to \( \Delta X_4^{S} = ((0,0,0,0),(?,? ,? ,?),(?,0,0,0),(0,?,0,?)) \) and as a result \( \Delta X_3^{M} = ((0,0,0,0),(?,?,? ,?),(0,0,0,0),(?,?,? ,?)) \).

After adding the \( \Delta k_2 \) we have \( \Delta X_2^{O} = \Delta X_5^{M} = (0,?,0,?),(?,0,0,0),(0,?,0,?) \) and after SubBytes and ShiftRows, we get \( \Delta X_3^{S} = ((N,?,0,?),(? ,0 ,0 ,0),(N,?,0,?),(? ,0 ,? ,?)) \) where \( ^{N} \) denotes nonzero difference (possibly distinct).

The second 3.5-round differential in the reverse direction is built as follows: The output difference \( \Delta X_5^{O} = ((a,0,0,0),(0,0,0,0),(0,0,0,0),(0,0,0,0)) \) is canceled by the subkey difference of the sixth round, i.e. \( \Delta X_6^{M} = \Delta k_3 \oplus \Delta X_5^{O} = 0 \). The zero difference \( \Delta X_6^{O} \) is preserved through all the operations until the AddRoundKey operation of the fourth round, because the subkey difference of the fifth round is zero. Thus we have \( \Delta X_6^{M} = \Delta k_3 \oplus \Delta X_5^{O} = ((a,0,0,0),(a,0,0,0),(0,0,0,0),(0,0,0,0)) \) and consequently from Theorem 3.1 \( \Delta X_7^{M} = ((N,N,N,N),(N,N,N,N),(0,0,0,0),(0,0,0,0)) \).

When rolling back the \( \Delta X_7^{M} \) through the ShiftRows and SubBytes operations in the fourth round, we get \( \Delta X_8^{O} = \Delta X_4^{S} = ((N,0,0,N),(N,N,0),(0,0,N),(0,N,N)) \). Finally after applying the AddRoundKey operation of the third round which has a zero difference, we can get \( \Delta X_9^{M} = ((N,0,0,N),(N,N,0),(0,0,N),(0,N,N)) \).

It is obvious that \( \Delta X_9^{M} = M(C(\Delta X_8^{M})) \), but according to the Theorem 3.1, this is impossible, because \( \Delta X_9^{M} \) has two passive bytes \( \Delta X_4^{M} \) has two active bytes and two passive bytes.

IV. RELATED-KEY IMPOSSIBLE DIFFERENTIAL ATTACK ON 8-ROUND AES-256 USING 64-BIT STRUCTURES

Using the above related-key impossible differential, we can attack an 8-round variant of AES-256.

A. The Attack Procedure

In order to make the attack faster, we first perform a precomputation. For all possible pairs of values of \( x_1^{M}\text{col}(0) \) and \( x_1^{M}\text{col}(3) \) which have the difference \( \Delta x_1^{M}\text{col}(0) = (a,?,?,? \) and \( \Delta x_1^{M}\text{col}(3) = (?,?,0,0) \), compute the values of \( (0,1,5,6,10,11,12,15) \) for \( x_1^{M} \). Store the pairs of 8-byte values in a hash table \( H_p \) indexed by the XOR difference in these
bytes. There are $2^{64}$ possible values for the XOR difference in 8 bytes and $(2^{16})^4 \times (2^8)^4 = 2^{96}$ possible pairs of values of $x^{M}_{1, \text{col}(0)}$ and $x^{M}_{1, \text{col}(3)}$ with above condition. So $H_p$ have $2^{64}$ bins and on average there are $\frac{2^{64}}{2^{96}} = 2^{22}$ pairs in each bin.

The algorithm is as follows:

1. Generate two pools $S_1$ and $S_2$ of $m$ plaintexts each, such that for each plaintext pair $P_1 \in S_1$ and $P_2 \in S_2$, $P_1 \oplus P_2 = ((?, ?, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0))$, where '?' denotes any byte value. Here we define a structure as a set of two $2 \times 2^{64}$ plaintexts which are selected from $S_1$ and $S_2$. Such a structure proposes $2^{64} \times 2^{64} = 2^{128}$ pairs of plaintexts.

2. For the askern of the pool $S_1$ under $K_1$, and of the pool $S_2$ under $K_2$. Denote the ciphertexts of the pool $S_1$ by $T_1$, and the encrypted ciphertexts of the pool $S_2$ by $T_2$. Such a structure proposes $2^{64} \times 2^{64} = 2^{128}$ pairs of plaintexts.

3. For all ciphertexts $C_2 \in T_2$, compute $C_2^* = C_2 \oplus ((0, 0, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0))$.

4. Insert all the ciphertexts $C_1 \in T_1$ and the values $\{C_2^* \mid C_2 \in T_2\}$ into a hash table indexed by bytes 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14 and 15.

5. For each bin of the hash table with more than one ciphertext, select every pair $(C_1, C_2)$. Note that every pair $(C_1, C_2)$ in each of this hash table have zero difference in bytes 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 14 and 15, so the pairs $(C_1, C_2)$ have zero difference in bytes 1, 2, 3, 4, 5, 6, 9, 11, 14 and 15, and difference ‘a’ in bytes 4, 8 and 12. After these steps we expect to have $2^{32} \times 2^{128} \times (2^8)^{12} = 2^{96+32}$ plaintext pairs, where $2^{n}$ is the number of structures, whose corresponding ciphertext pairs are equal in bytes 1, 2, 3, 4, 5, 6, 9, 11, 14 and 15, and difference ‘a’ in bytes 4, 8 and 12.

6. Guess the 32-bit value at bytes 0, 7, 10 and 13 for the $k_s$. Decrypt partially these bytes in the last round, i.e. compute $x^{W}_{7, \text{col}(0)} = SB^{-1} \circ SR^{-1}(x^{W}_{7, \text{col}(0)} \oplus k_s(0, 7, 10, 13))$. Choose pairs whose difference $\Delta x^{W}_{7, \text{col}(0)}$ is nonzero at byte $W_{7,0,0}$ and zero at other three bytes. The probability of such a difference is $(2^{-8})^3 = 2^{-24}$.

7. Guess the value of subkey byte $w_{7,0,0}$ and compute $x^{W}_{7,0,0} = SB^{-1} \circ SR^{-1}(x^{W}_{7,0,0})$ for all remaining pairs and choose pairs whose difference $\Delta x^{W}_{7,0,0}$ are ‘a’. The probability of such a difference is $2^{-8}$. Thus, at the end of this step, we can get $2^{6} \times 2^{128} \times (2^8)^{12} \times 2^{-24} \times 2^{-8} = 2^n$ pairs which have zero difference in all bytes except the first byte which have the difference ‘a’.

8. In this step, we eliminate wrong 64-bit values at $(0,1,5,6,10,11,12,15)$ for the $k_0$ by showing that the impossible differential property holds, if these keys were used. We use the hash table $H_p$, which has made in the precomputation stage. The algorithm of this step is as follows:

- Initialize a list $A$ of the $2^{64}$ possible values at $(0,1,5,6,10,11,12,15)$ of $k_0$.
- For each remaining pairs $(P_1, P_2)$, compute $P' = P_1 \oplus P_2$ in the eight bytes $(0,1,5,6,10,11,12,15)$.
- Access the bin $P'$ in $H_p$, and for each pair $(x,y)$ in that bin, $P_1 \oplus x$ remove from the list $A$ the value, where $P_1$ is restricted to eight bytes.
- If $A$ is not empty, output the values in $A$.

Note that there are $2^{22}$ pairs in each bin of $H_p$ on average, so in the third part of this step, we eliminate about $2^{32}$ wrong keys for each plaintext pair $(P_1, P_2)$. The probability of a wrong 64-bit value at bytes $(0,1,5,6,10,11,12,15)$ for $k_0$ is $(1 - 2^{-64})$, so after analyzing all $2^n$ pairs, we expect only $2^{64} \times (1 - 2^{-64})^{2^{32}}$ wrong values of the eight bytes of $k_0$ remain. For $n = 38.5$, the expected number is about $2^{64} \times (1 - 2^{-64})^{2^{32}} \approx 2^{64} \times (e^{-1})^{2^{32}} \approx 2^{-67}$ and we can expect that only the right subkey remains. Unless the initial guess of the 32-bit value of the last round key $k_s$ or the 8-bit value of the key $w_{7}$ is correct, it is expected that we can eliminate the whole 64-bit value of $k_0$ in this step, i.e. the list $A$ will be empty at the end of this step. Since the wrong values for $k_s, w_7, k_0$ occur with the small probability of $(2^8)^3 \times 2^8 \times 2^{-67} = 2^{-27}$. Hence if the list $A$ is not empty, we can assume that the guessed 32-bit value for $k_s$ and 8-bit value for $w_7$ are correct.
The data complexity of the attack is $2 \times 2^{n+64} = 2^{103.5}$ chosen plaintexts. The time complexity of the attack is consisted of three parts:

Step 6 requires $2 \times 2^{32} \times 2^{n+32} \times \frac{1}{10} = 2^{n+63}$ one round encryptions, because we must guess $2^{32}$ keys in this step, compute $X^W_{7,Col(0)}$ for each $2^{n+32}$ remained pairs from last steps.

Step 7 requires $2 \times 2^8 \times 2^{12} \times 2^{n+8} \times \frac{1}{8} = 2^{n+45}$ one round encryptions, because for all of guessed $2^{32}$ keys, we must guess $2^8$ for $k_8$ and compute $X^O_{6,0(0)}$ for each $2^{n+8}$ remained pairs from last steps.

In step 8, $2^{n-64}$ pairs are analyzed. For each pair we need $2^{32}$ memory accesses to $H_p$ and $2^{32}$ memory accesses to list A on average. This step is repeated $2^{40}$ times (for the guess of $w_7$ and $k_8$). Therefore the time complexity is $2^{40} \times 2^n \times (2^{32} + 2^{32}) = 2^{n+73}$ memory accesses, which are equivalent to about $2^{n+67}$ one round encryption (according to the implementations of NESSIE primitives [11]). Consequently for $n = 38.5$ the overall time complexity of the attack on 8-round AES-256 is about $2^{101.5} \times 2^{38.5} = 2^{102.5}$. The precomputation stage requires about $2^{2 \times 2^{6n}} = 2^{94}$ encryptions and the required memory is about $2^{100}$ bytes. Meanwhile, $2^{2^{n+8}+128} = 2^{101}$ bytes of memory are needed to store the list of deleted key values $k_8, w_7, k_0$ for the attack.

To achieving an attack with lower time complexity which is decreased by the factor $2^{20}$, at the cost of increasing data complexity by the factor $2^{13.5}$, we can use 32-bit structures instead of 64-bit structures. Like using 64-bit structures, we first perform a precomputation. For all possible pairs of values of $x^M_{4,Col(0)}$ which has the difference $\Delta x_{4,Col(0)} = (a, ?, ?, 0)$, compute the values of $(0,5,10,15)$ for $x^I$. Store the pairs of 4-byte values in a hash table $H_p$ indexed by the XOR difference in these bytes. There are $2^{32}$ possible values for XOR difference in 4 bytes and $(2^{16})^4 \times (2^8)^2 = 2^{38}$ possible pairs of values of $x^M_{4,Col(0)}$ with above condition. So $H_p$ have $2^{32}$ bins and on average there are $2^{16}$ pairs in each bin. The rest of the attack procedure is similar to 64-bit structure attack which we explain in this section.

### V. RESULTS AND DISCUSSION

In this paper, we proposed two new related-key impossible differential attacks against 8-round AES-256 using 64-bit and 32-bit structures. The dominant complexity of these attacks are lower than the previous related-key impossible differential attacks. Another important factor which made our attack more efficient is careful selection of two related keys difference, such that there is no unknown bytes in the subkey differences, which results in lower computational complexity.

### REFERENCES


