Improved Related-Key Impossible Differential Attacks on Reduced-Round AES-192

Wentao Zhang[1], Wenling Wu[2], Lei Zhang[2], and Dengguo Feng[2]

[1] State Key Laboratory of Information Security,
Graduate University of Chinese Academy of Sciences, Beijing 100049, P.R. China
zhangwt@gucas.ac.cn
[2] State Key Laboratory of Information Security,
Institute of Software, Chinese Academy of Sciences, Beijing 100080, P.R. China
{wwl,zhanglei1015,feng}@is.iscas.ac.cn

Abstract. In this paper, we present several new related-key impossible differential attacks on 7- and 8-round AES-192, following the work of Eli Biham et al. [6] and Jakimoski et al. [10]. We choose another relation of the related keys, start attacks from the very beginning(instead of the third round in [6]) so that the data and time complexities are improved largely, and only two related keys are needed instead of 32 in the attacks of [6]. Furthermore, we point out and correct an error in [6] when they attacked 8-round AES-192, then present our revised attacks. Finally, we give a new related-key differential attack on 7-round AES-192, which mainly uses a property of MixColumns operation of AES.

Keywords: AES, cryptanalysis, related-key differentials, impossible differentials.

1 Introduction

AES [12] supports 128-bit block size with three different key lengths (128,192,and 256 bits). Because of its importance, it's very necessary to constantly reevaluate the security of AES under various cryptanalytic techniques. In this paper, we study the security of 192-bit key version of AES(AES-192) against the related-key impossible differential attack.

Related-key attacks [2] allow an attacker to obtain plaintext-ciphertext pairs by using related(but unknown) keys. The attacker first searches for possible weaknesses of the encryption and key schedule algorithms, then choose appropriate relation between keys and make two encryptions using the related keys expecting to derive the unknown key information. Differential cryptanalysis [1] analyzes the evolvement of the difference between a pair of plaintexts in the following round outputs in an iterated cipher. Related-key differential attack [11] combines the above two cryptanalytic techniques together, and it studies the development of differences in two encryptions under two related keys. Furthermore, impossible differential attacks [3] use differentials that hold with probability 0(or non-existing differentials) to eliminate wrong key material and leave the right key candidate. In this case, the combined attack is called related-key impossible differential attack.

E. Biham and A.M. Youssef (Eds.): SAC 2006, LNCS 4356, pp. 15–27, 2007.

There are several impossible differential attacks on AES [4,7,8]. The best impossible differential attack on AES-192 is on 7 rounds [7].

If we view the expanded keys as a sequence of words, then the key schedule of AES-192 applies a non-linear transformation once every six words, whereas the key schedules of AES-128 and AES-256 apply non-linear transformations once every four words. This property brings better and longer related-key differentials of AES-192, so directly make AES-192 more susceptible to related-key attacks than AES-128 and AES-256. In the last few years, the security of AES-192 against related-key attacks has drawn much attention from cryptology researchers [5,6,9,10]. In [10], Jakimoski et al. presented related-key impossible differential attacks on 7- and 8-round AES-192. Following the work of [10], Biham et al.[6] gave several new related-key impossible differential attacks also on 7- and 8-round AES-192, which substantially improved the data and time complexity of those in [10]. Both in [5] and [9], the security of AES-192 against the related-key boomerang attack were studied. The best known related-key attack on AES-192 hitherto is due to Biham et al.[5], and it is applicable to a 9-round variant of AES-192.

Table 1. Comparison of Some Previous Attacks with Our New Attacks

Source	Number of Rounds	Data Complexity	Time Complexity	Number of Keys	Attack Type
Ref.[5]	9	2^{86} RK-CP	2^{125}	256	RK Rectangle
Ref.[7]	7	2^{92}CP	2^{186}	1	Imp.Diff
Ref.[10]	7	2^{111}RK-CP	2^{116}	2	RK Imp.Diff
	8	2^{88}RK-CP	2^{183}	2	
Ref.[6]	7	2^{56}RK-CP	2^{94}	32	RK Imp.Diff
	8	$2^{68.5}$RK-CP	$*2^{184}$	32	
	8	2^{92}RK-CP	$*2^{159}$	32	
	8	2^{116}RK-CP	$*2^{134}$	32	
This paper	7	2^{52}RK-CP	2^{80}	2	RK Imp.Diff
	8	$2^{64.5}$RK-CP	2^{177}	2	
	8	2^{88}RK-CP	2^{153}	2	
	8	2^{112}RK-CP	2^{136}	2	
This paper	7	2^{37}RK-CP	2^{145}	2	RK Diff

RK – Related-key, CP – Chosen plaintext,
Time complexity is measured in encryption units.

In this paper, we present several new related-key impossible differential attacks, following the work of [6] and [10]. In [6], the authors expressed: “We note that due to the special structure of the key schedule, the best round to start the attack with is round 2 of the original AES”. However, we can choose another key difference of the two related keys, and start the attacks from the very beginning, so greatly improve the data and time complexities of their attacks, and only two related keys are needed instead of 32 in [6]. Furthermore, we point out an error in [6] when they attacked 8-round AES-192, and then present our attacks. Lastly, we present a new related-key differential attack on 7-round AES-192, which can be regarded as a byproduct during the preparation of this paper, and it utilizes another property, ie., the specific property of Mixcolumns operation of AES.

Amongst our results, we reduce the data complexity of our attack by a factor of 2^4 and time complexity by a factor of 2^{14} compared with that in [6] for 7-round AES-192. The results are also improved in various degrees for several attacks on 8-round AES-192. Finally, a new related-key differential attack on 7-round AES-192 is presented, it needs more time, but the data complexity is reduced greatly. In all the attacks, only two related keys are needed. We summarize our results along with some previously known ones against AES-192 in Table 1. Note that there is an error in [6](which will be explained later), so the evaluated time complexities on 8 rounds are not right in [6], we mark them with "$*$".

Here is the outline of this paper. In Section 2, we give a brief description of AES. In Section 3, we choose another key difference of the two related keys, and present the corresponding subkeys difference of AES-192. Then a new 5.5-round related-key impossible differential is gained. Using this impossible differential, Section 4 presents an attack on 7-round AES-192; Section 5 firstly describes an error in [6], then presents three variants of our attacks on 8-round AES-192. Section 6 presents a new related-key differential attack on 7-round AES-192. Finally, Section 7 summarizes this paper.

2 Description of AES

The AES algorithm encrypts or decrypts data blocks of 128 bits by using keys of 128, 192 or 256 bits. The 128-bit plaintexts and the intermediate state are treated as byte matrices of size 4×4. Each round is composed of four operations:

- SubBytes(SB): applyinging the S-box on each byte.
- ShiftRows(SR): cyclically shifting each row (the i'th row is shifted by i bytes to the left, $i = 0, 1, 2, 3$).
- MixColumns(MC): multiplication of each column by a constant 4×4 matrix M over the field $GF(2^8)$, where M is

$$\begin{pmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{pmatrix}$$

  And the inverse of M is

$$\begin{pmatrix} 0e\ 0b\ 0d\ 09 \\ 09\ 0e\ 0b\ 0d \\ 0d\ 09\ 0e\ 0b \\ 0b\ 0d\ 09\ 0e \end{pmatrix}$$

- AddRoundKey(ARK): XORing the state and a 128-bit subkey.

The MixColumns operation is omitted in the last round, and an additional AddRoundKey operation is performed before the first round. We also assume that the MixColumns operation is omitted in the last round of the reduced-round variants.

The number of rounds is dependent on the key size, 10 rounds for 128-bit keys, 12 for 192-bit keys and 14 for 256-bit keys.

The key schedule of AES-192 takes the 192-bit secret key and expands it to thirteen 128-bit subkeys. The expanded key is a linear array of 4-byte words and is denoted by $G[4 \times 13]$. Firstly, the 192-bit secret key is divided into 6 words $G[0], G[1] \ldots G[5]$. Then, perform the following:

For $\ i = 6, \ldots 51$, do
If $(i \equiv 0 \mod 6)$, then $G[i] = G[i-6] \oplus SB(G[i-1] \lll 8) \oplus RCON[i/6]$
Else $\ G[i] = G[i-6] \oplus G[i-1]$

where $RCON[\cdot]$ is an array of predetermined constants, $\lll$ denotes rotation of a word to the left by 8 bits.

2.1 Notations

In the rest of this paper, we will use the following notations: x_i^I denotes the input of the i'th round, while x_i^S, x_i^R, x_i^M and x_i^O respectively denote the intermediate values after the application of SubBytes, ShiftRows, MixColumns and AddRoundKey operations of the i'th round. Obviously, $x_{i-1}^O = x_i^I$ holds.

Let k_i denote the subkey in the i'th round, and the initial whitening subkey is k_0. In some cases, the order of the MixColumns and the AddRoundKey operation in the same round is changed, which is done by replacing the subkey k_i with an equivalent subkey w_i, where $w_i = MC^{-1}(k_i)$.

Let $(x_i)_{Col(l)}$ denote the l'th column of x_i, where $l = 0, 1, 2, 3$. And $(x_i)_j$ the j'th byte of $x_i (j = 0, 1, \ldots 15)$, here Column(0) includes byte 0,1,2 and 3, Column(1) includes byte 4,5,6 and 7, etc.

3 A 5.5-Round Related-Key Impossible Differential of AES-192

We choose a new difference between two related keys as follows:

$((a, 0, 0, 0), (0, 0, 0, 0), (a, 0, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0))$.

Hence, the subkey differences in the first 8 rounds are as presented in Table 2, which will be used in our attacks later.

Throughout the attacks in our paper, we assume that the subkey differences are as presented in Table 2. A 5.5-round related-key impossible differential can be built like in [6] and [10]. Firstly, a 4.5-round related-key differential with probability 1 in the forward direction, then a 1-round related-key differential with probability 1 in the reverse direction, where the intermediate differences contradict each other. The 5.5-round related-key impossible differential is:

$$\Delta x_1^M = ((0,0,0,0), (0,0,0,0), (a,0,0,0), (a,0,0,0)) \xrightarrow{\text{5.5-round}}$$
$$\Delta x_6^O = ((?,?,?,?), (?,?,?,?), (?,?,?,?), (0,0,0,b))$$

Table 2. Subkey Differences Required for the Attacks in this paper

Round(i)	$\Delta k_{i,Col(0)}$	$\Delta k_{i,Col(1)}$	$\Delta k_{i,Col(2)}$	$\Delta k_{i,Col(3)}$
0	$(a,0,0,0)$	$(0,0,0,0)$	$(a,0,0,0)$	$(0,0,0,0)$
1	$(0,0,0,0)$	$(0,0,0,0)$	$(a,0,0,0)$	$(a,0,0,0)$
2	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$
3	$(a,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$
4	$(0,0,0,0)$	$(0,0,0,0)$	$(a,0,0,0)$	$(a,0,0,0)$
5	$(a,0,0,0)$	$(a,0,0,0)$	$(a,0,0,0)$	$(a,0,0,0)$
6	$(a,0,0,b)$	$(0,0,0,b)$	$(a,0,0,b)$	$(0,0,0,b)$
7	$(a,0,0,b)$	$(0,0,0,b)$	$(a,0,c,b)$	$(a,0,c,0)$
8	$(0,0,c,b)$	$(0,0,c,0)$	$(a,0,c,b)$	$(a,0,c,0)$

a, b and c are non-zero byte differences.

The above differential holds with probability 0, where a and b are non-zero values, "?" denotes any value.

The first 4.5-round differential is obtained as follows: the input difference Δx_1^M is canceled by the subkey difference of the first round. The zero difference Δx_2^I is preserved through all the operations until the AddRoundKey operation of the third round, as the subkey difference of the second round is zero. Thus, we can get $\Delta x_4^I = \Delta k_3 = ((a,0,0,0),(0,0,0,0),(0,0,0,0),(0,0,0,0))$, where only one byte is active. Then the next three operations in the fourth round will convert the active byte to a complete column of active bytes, and after the AddRoundKey operation with k_4, we will get $\Delta x_4^O = ((N,N,N,N),(0,0,0,0),(a,0,0,0),(a,0,0,0))$, where N denotes a non-zero byte(possibly distinct). Applying the SubBytes and ShiftRows operations of the 5'th round, Δx_4^O will evolve into $\Delta x_5^R = ((N,0,0,0),(0,0,0,N)$,$(N,0,N,0),(N,N,0,0))$, where only one byte is active both in Column 0 and Column 1. Hence, $\Delta x_5^M = ((N,N,N,N),(N,N,N,N)$, $(?,?,?,?),(?,?,?,?))$. Finally, after the key addition with k_5, we can get $\Delta x_5^O = ((?,N,N,N),(?,N,N,N),(?,?,?,?),(?,?,?,?))$.

The second differential ends after the 6'th round with output difference $\Delta x_6^O = ((?,?,?,?),(?,?,?,?),(?,?,?,?),(0,0,0,b))$. When rolling back this difference through the AddRoundKey and MixColumn operations, we get the difference in the last column of Δx_6^R is zero. Hence, $\Delta x_6^I = ((?,0,?,?),(?,?,0,?),(?,?,?,0)$, $(0,?,?,?))$. It's obvious that $\Delta x_6^I = \Delta x_5^O$ with probability 1. However, we can see that $(\Delta x_5^O)_1$ is a non-zero byte in the first 4.5-round differential, while $(\Delta x_6^I)_1$ is a zero byte in the second differential, this is a contradiction.

Using the above 5.5-round impossible differential, we can start our attacks from the very beginning. Hence, compared with the attacks in [6], there is no unknown bytes in the key difference, whereas one unknown byte in the key difference in [6]. Therefore, our attacks can proceed with one less byte guessing, which makes the time complexity reduced at least by a factor of 2^7. Moreover, only two related keys are needed, which makes the data complexity reduced by a factor of 2^4 immediately.

4 A 7-Round Related-Key Impossible Differential Attack

Using the above impossible differential, we can attack a 7-round variant of AES-192.

At first, we assume that the values of a, b, c are all known, ie., we have two related keys K_1 and K_2 with the required subkey differences listed in Table 2. We will deal with the conditions on the related keys to achieve these subkey differences at the end of this section.

The attack procedure is quite similar to that in [6]. However, the attack complexity will be reduced significantly.

4.1 The Attack Procedure

Precomputation: For all the 2^{64} possible pairs of values of the last two columns of x_1^M(ie.,$(x_1^M)_{Col(2)}$ and $(x_1^M)_{Col(3)}$) both with difference $(a, 0, 0, 0)$, compute the 8 byte values in bytes 1, 2, 6, 7, 8, 11, 12, and 13 of plaintext P. Store the pairs of 8-byte values in a hash table H_p indexed by the XOR differences in these bytes.

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 $P_2 \in S_2$, $P_1 \oplus P_2 = ((a, ?, ?, 0), (0, 0, ?, ?), (?, 0, 0, ?), (?, ?, 0, 0))$, where ? denotes any byte value.
2. Ask for the encryption 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.
3. For all ciphertexts $C_2 \in T_2$, compute $C_2^* = C_2 \oplus ((0, 0, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0), (a, 0, 0, 0))$.
4. Insert all the ciphertexts $C_1 \in T_1$ and the values $\{C_2^* | C_2 \in T_2\}$ into a hash table indexed by bytes 6,9,and 12.
5. Guess the value of the subkey byte $(k_7)_3$ and perform the followings:
   (a) Initialize a list A of the 2^{64} possible values of the bytes 1, 2, 6, 7, 8, 11, 12, and 13 of k_0.
   (b) Decrypt the byte $(x_7^O)_3$ in all the ciphertexts to get the intermediate values before the subkey addition in the 6'th round.
   (c) For every pair C_1, C_2^* in the same bin of the hash table, check whether the corresponding intermediate values calculated in Step 5(b) are equal. If no, discard the pair.
   (d) For every remaining pair C_1, C_2^*, consider the corresponding plaintext pair and compute $P_1 \oplus P_2$ in the eight bytes 1, 2, 6, 7, 8, 11, 12, and 13. Denote the resulting value by P'.
   (e) If the bin P' in H_p is nonempty, access this bin. For each pair (x, y) in that bin remove from the list A the values $P_1 \oplus x$ and $P_1 \oplus y$, where P_1 is restricted to eight bytes (plaintext bytes 1, 2, 6, 7, 8, 11, 12, and 13).
   (f) If A is not empty, output the values in A along with the guess of $(k_7)_3$.

4.2 Analysis of the Attack Complexity

From the two pools of m plaintexts each, m^2 possible ciphertexts pairs (C_1, C_2^*) can be derived. After the filtering in step 4, there remains about $2^{-24}m^2$ pairs in each bin of the hash table. In Step 5, we have an additional 8-bit filtering for every possible value of $(k_7)_3$ separately, so about $2^{-32}m^2$ pairs will remain for a given subkey guess of $(k_7)_3$. Each pair deletes one subkey candidate on average, and there are 2^{64} subkey candidates in all, so the expected number of remaining subkeys is $2^{64}(1 - 1/2^{64})^{m'}$ in step 5(f). If $m' = 2^{70}$, the expected number is about $e^{-20} = 2^{-28.85}$, and we can expect that only the right subkey will remain. Hence, we get the value of $64 + 8 = 72$ subkey bits. In order to derive $m' = 2^{70}$, we need $m = 2^{51}$ chosen plaintexts in each of the two pools. So the data complexity of the attack is 2^{52} chosen plaintexts.

The time complexity is dominated by Step 5(e). In this step, $m' = 2^{70}$ pairs are analyzed, leading to one memory access on average to H_p and one memory access to A. This step is repeated 2^8 times (once for one guess of $(k_7)_3$). Therefore, the time complexity is 2^{79} memory accesses, which is equivalent to about 2^{73} encryptions. The precomputation requires about 2^{62} encryptions and the required memory is about 2^{69} bytes.

In the above attack, we assumed that the values of a, b and c are known. Here,the value a can be chosen by the attacker. The value b is the result of application of SubBytes operation, so there are 127 possible values of b given the value of a. And the attack can proceed without knowing the value of c. Hence,we only need to repeat the attack for all the possible values of b. Therefore,the total time complexity is multiplied by 2^7, the data and memory complexity remain unchanged.

To sum up, the total complexity of the above attack is as follows: The data complexity is 2^{52} chosen plaintexts, the time complexity is 2^{80} encryptions, and the required memory is 2^{69} bytes.

5 Three 8-Round Related-Key Impossible Differential Attacks

In this section, we point out an error in [6] when they attacked 8-round AES-192. Then, present our own attacks.

5.1 An Error in the 8-Round Attacks of [6]

In Section 4 of [6], the authors presented three variant attacks on 8-round AES-192. In the first version, 13 key bytes are guessed: bytes 0,2,3,5,6,7,8,9,10,12,13 and 15 of k_9, and byte 11 of w_8(ie., $(w_8)_{3,2}$ in [6]). For peeling off the last round, only those ciphertext pairs which have zero difference (before the subkey addition with k_9) in the remaining four bytes 1,4,11 and 14 are treated. Then rolling back through the ShiftRows and SubBytes operations, difference in the four bytes of Column 1 are all zero, ie.,$(\Delta x_9^I)_{Col(1)} = (0, 0, 0, 0)$. This property still holds

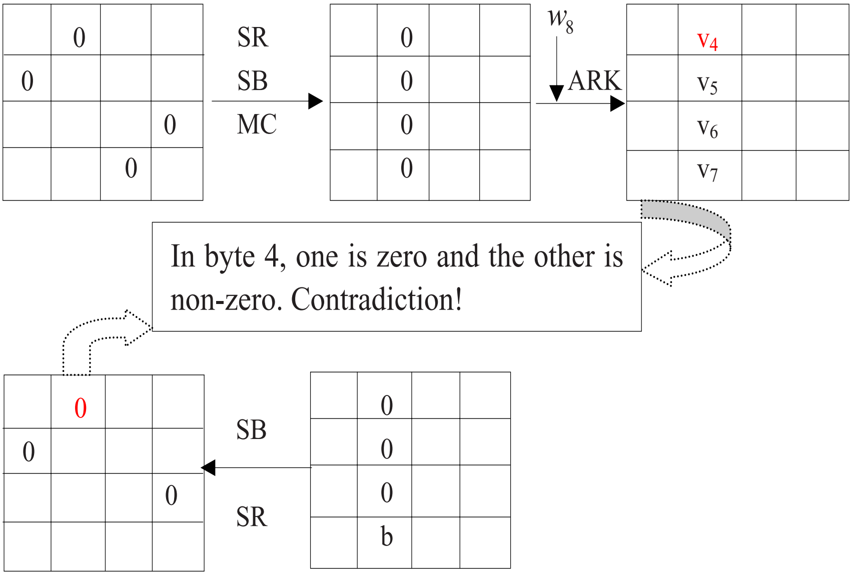
0
0
0
0
SR
SB
MC
0
0
0
0
w_8
ARK
v4
v5
v6
v7
In byte 4, one is zero and the other is non-zero. Contradiction!
0
0
0
SB
SR
0
0
0
b

Fig. 1. An Error in Ref.[6]

when applying the MixColumns operation in round 8. Then applying the subkey addition with w_8 in round 8, difference in the four bytes of Column 1 will equal to the corresponding byte of Δw_8. Especially, we can get $(\Delta x_8^R)_4 = (\Delta w_8)_4$. Note that $(\Delta w_8)_4$ is determined by $(\Delta k_8)_{Col(1)}$ which have one non-zero byte and three zero bytes, thus applying the inverse operation of MixColumns to $(\Delta k_8)_{Col(1)}$, we can deduce that $(\Delta x_8^R)_4{=}(\Delta w_8)_4$ is a non-zero byte. Then we can conclude that $(\Delta x_8^I)_4$ is a non-zero byte immediately.

However, when applying the impossible differential from round 2 to round 7 in [6], the attacker must filter a certain amount of plaintext pairs to satisfy the requirement of the differential, especially $(\Delta x_8^I)_4 = (\Delta x_7^O)_4$ equals to zero. This is a contradiction, which is emphasized in Figure 1, in which the first pane denotes the intermediate state before the subkey addition with k_9, the last pane denotes the input state of round 8, and the other panes denote the intermediate state between them. For simplicity, let v_4, v_5, v_6, v_7 denote byte 4,5,6,7 of w_8 respectively. Here, we will use x_8^W to denote the intermediate value after the application of AddRoundKey operation with w_8 in round 8. From the above analysis, in order to correctly use the 5.5-round differential, we must filter a certain amount of pairs which satisfy that $(\Delta x_8^W)_4{=}(\Delta w_8)_4$ to make $(\Delta x_8^I)_4 = 0$. Then, after the Mixcolumns operation in round 8, this byte has relation with all the four bytes in Column 1. Thus, it seems that four more key bytes of k_9 should be guessed in order to calculate the byte difference $(\Delta x_8^W)_4$ from ciphertext pairs. However, we can deal with this problem by guessing only one more key byte, and treat only ciphertext pairs that have zero difference in the other three bytes.

For the second and third version of the attacks on 8-round AES-192 in [6], there exists similar errors. And we will adopt the same technique to reduce the amount of key bytes guess.

In the following, we will present our attacks on 8-round AES-192.

5.2 Our Attacks on 8-Round AES-192

In the following, we will give three variants of attacks on 8-round AES-192, which are all based on the 7-round attack in Section 4. As in [6], the main difference between them is a data-time trade-off. In all the 8-round attacks, we guess part of the last round subkey k_8, peel off the last round and apply the 7-round attack. In order to reduce the amount of key material guess, we also change the order of the MixColumns and the AddRoundKey operations in the 7'th round, this is done by replacing the subkey k_7 with an equivalent subkey w_7. And we use x_7^W to denote the intermediate value after the application of AddRoundKey operation with w_7 in the 7'th round.

If a pair of ciphertexts satisfy the condition that $(\Delta x_6^O)_{Col(3)} = (0,0,0,b)$, then after the SubBytes and ShiftRows operations of the 7'th round, bytes 6,9 and 12 of Δx_7^R must be zero. Next, applying the key addition with w_7, we can get $(\Delta x_7^W)_6 = (\Delta w_7)_6$,$(\Delta x_7^W)_9 = (\Delta w_7)_9$, and $(\Delta x_7^W)_{12} = (\Delta w_7)_{12}$.

In order to satisfy the above conditions and guess less subkey material, we treat only ciphertext pairs that have certain properties. Take for example, to make $(\Delta x_7^W)_6 = (\Delta w_7)_6$, we only choose ciphertext pairs that satisfy $(\Delta x_7^O)_{Col(1)} = (z_4, 0, 0, 0)$, where z_4 is uniquely determined by $(\Delta w_7)_6$ to make the above condition hold, ie., $MC^{-1}(z_4, 0, 0, 0) = (?, ?, (\Delta w_7)_6, ?)$. Similarly, we can decide the values of bytes z_8(or z_{11}) and z_{12}, which make $(\Delta x_7^W)_9 = (\Delta w_7)_9$ and $(\Delta x_7^W)_{12} = (\Delta w_7)_{12}$ respectively.

The attack can be performed in one out of three possible ways.

The First Attack. Guess bytes 0,1,2,4,5,7,8,10,11,12,13,14, and 15 of k_8, partially decrypt these bytes in the last round. These subkey bytes allow us to partially decrypt the last round in Columns 0,1 and 2. And we only treat ciphertext pairs that have zero difference in the remaining 3 bytes(before the key addition with k_8, the same below). This condition allows us to use 2^{-24} of the possible ciphertext pairs. Then the difference Δx_8^I is known, we first check whether the difference in byte 12 equals to z_{12}. This filtering is done using a 8-bit condition, which makes the remaining ciphertext pairs satisfy the condition that byte 12 in Δx_6^O is zero. Next, applying the inverse of MixColumns operation to Column 1 and Column 2 of Δx_8^I, calculate byte 6 and 9 of Δx_7^W and check whether they are equal to $(w_7)_6$ and $(w_7)_9$ respectively. This filtering thus makes bytes 13 and 14 of Δx_6^O equal to zero too, and uses a 16-bit condition. Then, guess byte 3 of w_7 and continue partial decryption to find out whether $(\Delta x_6^O)_{15} = b$ holds, which is done using a 8-bit condition. After this filtering, the remaining ciphertext pairs can be used to discard wrong subkey guesses like in the 7-round attack.

In this variant of the attack, we guess a total of 112 subkey bits. And a portion of $2^{-24-8-16-8} = 2^{-56}$ of the pairs can be used in the attack to discard the wrong subkey guesses.

Here we can use the differential properties of the key schedule algorithm. The value of b can be determined by a and $(k_5)_{12} = (k_8)_4 \oplus (k_8)_{12}$, the value of c can be determined by b and $(k_7)_7 = (k_8)_{11} \oplus (k_8)_{15}$.

About $2^{63.5}$ plaintexts in each pool are needed to derive about $2^{63.5+63.5-56} = 2^{71}$ data pairs for every guess of the 112-bit key material guess in the last two rounds. Each pair discards one possible value for the eight bytes guess of subkey k_0 on average. Therefore, the probability that some wrong subkey guess remains is at most $2^{64}e^{-128} \approx 2^{-120}$, and the expected number of subkey suggestions is approximately $2^{-120}2^{112} = 2^{-8}$. Hence, with a high probability only the right value will remain. The data complexity of this attack is about $2^{64.5}$ chosen plaintexts. The time complexity is about $2^{71} \times 2^{112}/2^6 = 2^{177}$ and the required memory is about 2^{69} bytes.

The Second Attack. Guess bytes 0,1,4,7,10,11,12,13,14, and 15 of k_8. And treat only ciphertext pairs that have zero difference in the remaining 6 bytes. This condition allows us to use only 2^{-48} of the possible ciphertext pairs. Then the difference Δx_8^I is known, we first check whether the difference in bytes 11 and 12 are z_{11} and z_{12} respectively. This filtering is done using a 16-bit condition, which makes the remaining ciphertext pairs satisfy the conditon that bytes 12 and 13 in Δx_6^O are all zero. Next, calculate the difference in byte 6 of Δx_7^W and check whether it equals to $(w_7)_6$. This filtering uses a 8-bit condition, and makes the remaining pairs also satisfy that byte 14 of Δx_6^O equals to zero. Then, guess byte 3 of w_7 and continue partial decryption to find out whether $(\Delta x_6^O)_{15} = b$ holds. This is done using a 8-bit condition. After this filtering, the remaining ciphertext pairs can be used to discard wrong subkey guesses.

In this variant of the attack, we guess a total of 88 subkey bits. But only a portion of 2^{-80} of the pairs can be used in the attack to discard wrong subkey guesses.

As in the first attack, the value of b and c can also be determined by a and subkey guess of k_8.

Choose a pool of 2^{64} plaintexts which differ only at the eight bytes 1, 2, 6, 7, 8, 11, 12 and 13, and having all possible values in these bytes. Choose two such pools S_1 and S_2, such that for each plaintext pair $P_1 \in S_1$ and $P_2 \in S_2$, $P_1 \oplus P_2 = ((a, ?, ?, 0), (0, 0, ?, ?), (?, 0, 0, ?), (?, ?, 0, 0))$. Encrypt S_1 and S_2 under the two related keys each. Then, we can derive $2^{64} \times 2^{64} = 2^{128}$ pairs of plaintexts using 2^{65} chosen plaintexts, call such two pools a structure.

About 2^{23} structures are needed to get about 2^{71} data pairs which can be used to delete wrong subkey guesses. Hence, the data complexity of this attack is about 2^{88} chosen plaintexts. The time complexity is about $2^{71} \times 2^{88}/2^6 = 2^{153}$.

The Third Attack. Guess bytes 0,7,10,13,4,8, and 12 of k_8, partially decrypt these bytes in the last round. And treat only ciphertext pairs that have zero difference in the remaining 9 bytes. This condition allows us to use only 2^{-72}

of the possible ciphertext pairs. Then the difference Δx_8^I is known, we check whether the difference in bytes 4,8 and 12 are z_4, z_8 and z_{12} respectively. This filtering is done using a 24-bit condition. Thus, the remaining ciphertext pairs all satisfy that bytes 12,13,14 in Δx_6^O are all zero. Then, guess byte 3 of w_7 and continue partial decryption to find out whether $(\Delta x_6^O)_{15} = b$ holds. This is done using a 8-bit condition. After this filtering, the remaining ciphertext pairs can be used to discard wrong subkey guesses.

In this attack variant, we guess only 64 subkey bits. But only a portion of 2^{-104} of the pairs can be used in the attack. This leads to a relatively high data complexity, but to a lower time complexity.

The value of b can be determined by a and subkey guess of k_8. Hence, we only need to repeat the attack for all the 127 possible values of c.

About 2^{47} structures are needed to get about 2^{71} data pairs which can be used to delete wrong subkey guesses. Hence, the data complexity of this attack is about 2^{112} chosen plaintexts. The time complexity is about $2^7 \times 2^{71} \times 2^{64}/2^6 = 2^{136}$.

6 A New Related-Key Differential Attack on 7-Round AES-192

In this section, we present a new related-key differential attack on 7-round AES-192, which mainly uses the specific property of Mixcolumns operation of AES.

The subkey differences are also as presented in Table 2, and we will change the order of the MixColumns and the AddRoundKey operations in the 6'th round. Submit two plaintexts P_1 and P_2 for encryption under the two related keys respectively. Similar to the analysis in Section 3, if$\Delta x_1^M = ((0,0,0,0),(0,0,0,0),$ $(a,0,0,0),(a,0,0,0))$, then we can conclude that Δx_5^R must have the form of $((N,0,0,0),(0,0,0,N)$,$(N,0,N,0),(N,N,0,0))$, where only one byte is active both in Column 0 and Column 1. Considering Column 1, we have $(\Delta x_5^R)_{Col(1)} =$ $(0,0,0,N)$, then after the following MixColumns operation, we can get that $(\Delta x_5^M)_4$ =$(\Delta x_5^M)_5$ holds with probability 1 because of the specific MixColumns operation of AES.

In order to calculate the values of bytes 4 and 5 in Δx_5^M, we need to guess 10 subkey bytes: bytes 0,1,4,7,10,11,13, and 14 of k_7, and bytes 1,4 of w_6.

The attack procedure has many similarities to the above attacks, including the precomputation, the initial plaintexts selection and encryption, and the initialization of the list A. The difference only consists in the filtering condition of data. For each guess of the 10 key bytes, we will calculate the two values of $(x_5^M)_4$ and $(x_5^M)_5$ for each plaintext, then check whether $(\Delta x_5^M)_4$ =$(\Delta x_5^M)_5$ for each data pairs. If not, we can use it to delete the corresponding key guess from A as in the above attacks.

From the two pools of m plaintexts each, m^2 possible ciphertext pairs can be derived. For every possible guess of the 10 key bytes, about $m' = (1-2^{-8})m^2$ pairs can be used to delete the wrong subkey guesses of k_0. Each pair deletes one subkey candidate on average, so the probability that some wrong subkey guess remains is at most $2^{64}(1-1/2^{64})^{m'}$. If $m'=2^{71}$, the expected number is about 2^{-120}, and we can expect that only the right subkey will remain. Hence, we get the value of $80+64=144$ subkey bits. In order to derive $m'=2^{71}$, we need about $m=2^{36}$ chosen plaintexts in each of the two pools. So the data complexity of the attack is about 2^{37} chosen plaintexts.

The time complexity is about $2^{71}\times 2^{80}/2^{6}=2^{145}$. And the required memory is also about 2^{69} bytes.

Here, b can be calculated from a and $(k_5)_{12}$=$(k_7)_0 \oplus (k_7)_4$, and c can be calculated from b and $(k_7)_7$.

To sum up, the total complexity of the above attack is as follows: The data complexity is 2^{37} chosen plaintexts, the time complexity is 2^{145} encryptions, and the required memory is 2^{69} bytes.

Compared with the 7-round attack in Section 4, the data complexity is decreased, but the time complexity is increased greatly. Nevertheless, the attack uses another different point of AES, ie., the specific property of the MixColumns operation.

7 Summary

Up to now, better results are achieved against reduced-round AES-192 using related-key cryptanalysis in contrast to other non-related-key cryptanalysis approaches. This fact reflects some weaknesses of the key schedule algorithm of AES-192.

In this paper, we improved the attack results presented in [6] and [10] through choosing a new difference of the related keys. Furthermore, we detected an error in [6] when they attacked 8-round AES-192, then presented our revised attacks. The new chosen related-key difference made our attack start from the very beginning instead of the third round as in [6]. Hence, the number of unknown bytes in the subkey differences become less, which makes the attack complexity improved largely in this paper. The comparison of our attack results and those in [6] and [10] can be found in Table 1.

We also present a new related-key differential attack on 7-round AES-192, which mainly utilizes the property of MixColumns operation, but it is a pity that we can't extend the attack to 8-round at present, we wish that this point may be used in further attacks on AES.

Acknowledgment

We would like to thank anonymous referees for their helpful comments and suggestions. The research presented in this paper is supported by the National Natural Science Foundation of China(No.60373047, 90604036); and the National Basic Research 973 Program of China No.2004CB318004.

References

1. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology 4(1), 3–72 (1991)
2. Biham, E.: New Types of Cryptanalytic Attacks Using Related Keys. Journal of Cryptology 7(4), 229–246 (1994)
3. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 12–23. Springer, Heidelberg (1999)
4. Biham, E., Keller, N.: Cryptanalysis of Reduced Variants of Rijndael, in Official public comment for Round 2 of the AES development effort (2000) Available at `http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html`
5. Biham, E., Dunkelman, O., Keller, N.: Related-Key Boomerang and Rectangle Attacks. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 507–525. Springer, Heidelberg (2005)
6. Biham, E., Dunkelman, O., Keller, N.: Related-Key Impossible Differential Attacks on 8-Round AES-192. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 21–33. Springer, Heidelberg (2006)
7. Phan, R.C.-W.: Impossible Differential Cryptanalysis of 7-round Advanced Encryption Standard (AES). Information Processing Letters 91(1), 33–38 (2004)
8. Cheon, J.H., Kim, M., Kim, K., Lee, J.-Y., Kang, S.: Improved Impossible Differential Cryptanalysis of Rijndael and Crypton. In: Kim, K.-c. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 39–49. Springer, Heidelberg (2002)
9. Hong, S., Kim, J., Kim, G., Lee, S., Preneel, B.: Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 368–383. Springer, Heidelberg (2005)
10. Jakimoski, G., Desmedt, Y.: Related-Key Differential Cryptanalysis of 192-bit Key AES Variants. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 208–221. Springer, Heidelberg (2004)
11. Kelsey, J., Schneier, B., Wagner, D.: Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 233–246. Springer, Heidelberg (1997)
12. National Institute of Standards and Technology. Advanced Encryption Standard (AES), FIPS Publication 197 (November 26, 2001) Available at `http://csrc.nist.gov/encryption/aes`