ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.11, November 2007, pp.2893−2901
DOI: 10.1360/jos182893

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

对低轮AES-256 的相关密钥-不可能差分密码分析*

张文涛[1+], 吴文玲[2], 张 蕾[2]

[1](中国科学院 研究生院 信息安全国家重点实验室,北京 100049)

[2](中国科学院 软件研究所 信息安全国家重点实验室,北京 100080)

Related-Key Impossible Differential Attacks on Reduced-Round AES-256

ZHANG Wen-Tao[1+], WU Wen-Ling[2], ZHANG Lei[2]

[1](State Key Laboratory of Information Security, Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

[2](State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-88256218, Fax: +86-10-88258713, E-mail: zhangwt@gucas.ac.cn

Zhang WT, Wu WL, Zhang L. Related-Key impossible differential attacks on reduced-round AES-256. *Journal of Software*, 2007,18(11):2893−2901. http://www.jos.org.cn/1000-9825/18/2893.htm

Abstract: In this paper, the strength of AES-256 against the related-key impossible differential attack is examined. Firstly, a carefully chosen relation between the related keys is presented, which can be extended to 8-round (even more rounds) subkey differences. Then, a 5.5-round related-key impossible differential is constructed. Finally, an attack on 7-round AES-256 and four attacks on 8-round AES-256 are presented.

Key words: AES-256; cryptanalysis; related-key differential; impossible differential

摘 要: 研究 AES-256 抵抗相关密钥-不可能差分密码分析的能力.首先给出相关密钥的差分,该差分可以扩展到8轮(甚至更多轮)子密钥差分;然后构造出一个5.5轮的相关密钥不可能差分特征.最后,给出一个对7轮AES-256的攻击和4个对8轮AES-256的攻击.

关键词: AES-256;密码分析;相关密钥差分;不可能差分

中图法分类号: TP309 **文献标识码**: A

1 Introduction

AES[1] supports 128-bit block size with three different key lengths(128, 192, and 256 bits). In this paper, we examine the strength of 256-bit key version of AES (AES-256) against the related-key impossible differential attack, following the work of Refs.[2,3].

Related-key attacks[4] 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

* Supported by the National Natural Science Foundation of China under Grant Nos.60373047, 90604036 (国家自然科学基金); the National Basic Research Program of China under Grant No.2004CB318004 (国家重点基础研究发展计划(973)); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z470 (国家高技术研究发展计划(863))

Received 2006-05-19; Accepted 2006-08-16

chooses an appropriate relation between keys and makes two encryptions using the related keys expecting to derive the unknown key information. The complexity of the attack depends on the ability of the attacker to predict the propagation of the key difference during the key schedule. Related-key differential attacks[5] study the development of differences in two encryptions under two related keys. Impossible differential attacks[6,7] use differentials that hold with probability 0 (or non-existing differentials) to eliminate the wrong key material and leave the right key candidate. And in this case, the attack is called related-key impossible differential attack.

Among the three variants of AES, the key schedule of AES-256 is a little different from those of AES-128 and AES-192. Let *NK* denote the number of words of the seed key, thus *NK*=4 for AES-128, *NK*=6 for AES-192 and *NK*=8 for AES-256. In the key schedule of AES-128 and AES-192, the non-linear transformation is applied only once every *NK* word. While for the key schedule of AES-256, non-linear transformation is applied twice every *NK* words. Thus, it seems that AES-256 is relatively more immune to related-key attacks compared with AES-128 and AES-256. The best known related-key attack on AES-256 uses boomerang cryptanalysis and it is applicable to a 10-round variant of AES-256[8]. And the best known non-related-key attack on AES-256 uses integral cryptanalysis[9], and is applicable to 8-round AES-256, which needs almost the whole data of plaintext space.

Because of the importance of AES, it's very necessary to constantly reevaluate the security of AES under various cryptanalytic techniques. Concerning related-key impossible differential cryptanalysis, there has already several works against reduced-round AES-192[1,2,10,11], whereas few work on AES-256. In this paper, we will study the strength of AES-256 against related-key impossible differential attack.

The work in this paper follows those in Refs.[1,2], and starts the attack from the very beginning. Firstly, we carefully choose an appropriate relation between the two related keys, which can be extended to 8-round (even more rounds) subkeys difference using the key schedule of AES-256. Next, we construct a 5.5-round related-key impossible differential characteristic. Using this characteristic, we then present an attack against 7-round AES-256 and four attacks against 8-round AES-256. The results in this paper and other main results on AES-256 are summarized in Table 1. Compared with the results in Ref.[3], we can also see that AES-256 have a better resistance than AES-192 using the same cryptanalytic approach.

Table 1 Summary of the attacks on AES-256

Source	Number of rounds	Data complexity	Time complexity	Number of keys	Attack type
Ref.[9]	8	$2^{128}-2^{119}$ CP	2^{104}	1	Integral attack
	9	2^{85} RK-CP	5×2^{224}	256	RK attack
Ref.[8]	10	$2^{114.9}$ RK-CP	$2^{171.8}$	256	RK rectangle
This paper	7	2^{52} RK-CP	2^{87}	2	RK Imp.Diff
	8	2^{53} RK-CP	2^{215}		
	8	2^{64} RK-CP	2^{191}		
	8	2^{88} RK-CP	2^{167}		
	8	2^{112} RK-CP	2^{143}		

RK: Related-Key, CP: Chosen plaintext,
Time complexity is measured in encryption units.

Here is the outline of the paper. In Section 2, we give a brief description of AES. In Section 3, we carefully choose a key difference between the two related keys, and present the corresponding 8-round subkeys difference. Then a 5.5-round related-key impossible differential is constructed. Section 4 gives an attack against 7-round AES-256. Section 5 presents four variants of the attacks against 8-round AES-256. Finally, Section 6 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-256 is slightly different from those of AES-128 and AES-192. It takes the 256-bit secret key and expands it to 15 128-bit subkeys, and SubBytes operation is applied twice every 8 words. The expanded key is a linear array of 4-byte words and is denoted by $G[4\times15]$. Firstly, the 256-bit seed key is divided into 8 words $G[0],G[1],\ldots,G[7]$. Then, perform the following:

For $i=8,\ldots,59$, do

If ($i\equiv0 \bmod 8$), then $G[i]=G[i-8]\oplus SB(G[i-1]<<<8)\oplus RCON[i/8]$

Else if ($i\equiv4 \bmod 8$), then $G[i]=G[i-8]\oplus SB(G[i-1])$

Else $G[i]=G[i-8]\oplus G[i-1]$

where *RCON*[ ] is an array of predetermined constants, <<< denotes rotation of a word to the left by 8 bits. Figure 1 exhibit the key schedule algorithm of AES-256.

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, $x_{i-1}^O = x_i^I$ always 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 repalacing 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,…,15), here Column(0) includes bytes 0, 1, 2 and 3, Column(1) includes bytes 4, 5, 6 and 7, etc.

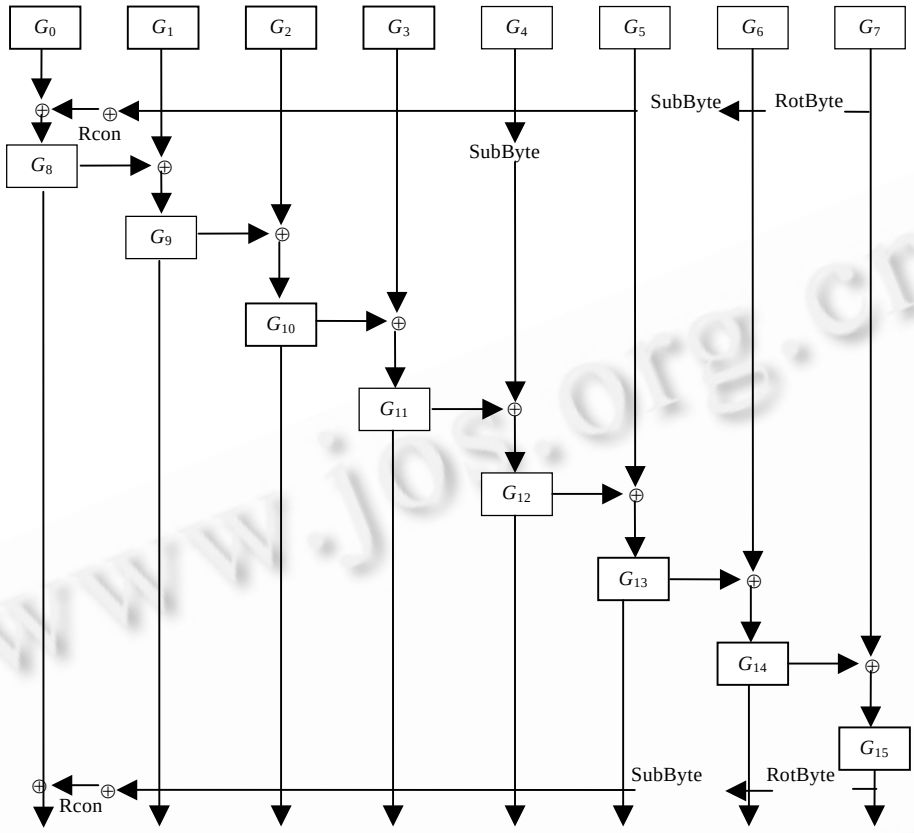


Fig.1 Key schedule algorithm of AES-256

3 A 5.5-Round Related-Key Impossible Differential of AES-256

We will choose the difference between the two related keys as follows:

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

Hence, through the key schedule, the subkey differences in the first 8 rounds are presented in Table 2, which will be used in our attacks. Note that the subkey differences can be extended to more rounds and only one more unknown byte comes forth when one round is added.

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	(0,0,0,0)	(a,0,0,0)	(0,0,0,0)	(0,0,0,0)
1	(a,0,0,0)	(a,0,0,0)	(0,0,0,0)	(0,0,0,0)
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)	(0,0,0,0)	(0,0,0,0)
5	(a,0,0,0)	(a,0,0,0)	(a,0,0,0)	(a,0,0,0)
6	(0,0,0,b)	(0,0,0,b)	(0,0,0,b)	(0,0,0,b)
7	(a,0,0,c)	(0,0,0,c)	(a,0,0,c)	(0,0,0,c)
8	(0,0,d,b)	(0,0,d,0)	(0,0,d,b)	(0,0,d,0)

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

Throughout the attacks in this paper, we assume that the subkey differences are as presented in Table 2. And the attacks presented in this paper start from the very beginning, only two related keys are needed.

In the following, we will present a 5.5-round related-key impossible differential as those in Refs.[2,3]. Firstly, we present 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:

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

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 constructed 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 key 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 $\Delta x_4^O=((N,N,N,N),(0,0,0,0),(0,0,0,0),(0,0,0,0))$, where N denotes a non-zero byte (possibly distinct). Applying the SubBytes and ShiftRows of the 5th round, Δx_4^O will evolve into $\Delta x_5^S=((N,0,0,0),(0,0,0,N),(0,0,N,0),(0,N,0,0))$, where only one byte is active in each of the four Columns. Hence, $\Delta x_5^M=((N,N,N,N),(N,N,N,N),(N,N,N,N),(N,N,N,N))$. Finally, after the key addition with k_5, we can get $\Delta x_5^O=((?,N,N,N),(?,N,N,N),(?,N,N,N),(?,N,N,N))$. Hence, the input difference $\Delta x_1^M=((a,0,0,0),(a,0,0,0),(0,0,0,0),(0,0,0,0))$ evolves with probability 1 into a non-zero difference in bytes 1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14 and 15 of Δx_5^O.

The second differential ends after the 6'th round with output difference in any of the four columns $(\Delta x_6^O)_{Col(i)}=(0,0,0,b)$, where i=0,1,2,3. Take Column 2 for example, when rolling back this difference through the AddRoundKey and MixColumn operations, we get the difference in Column 2 of Δx_6^R as 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 $(\Delta x_5^O)_2$ is a non-zero byte in the first 4.5-round differential, while $(\Delta x_6^I)_2$ is a zero byte in the second differential, this is a contradiction.

Note that we can choose any of the four columns of Δx_6^O as $(0,0,0,b)$. It's possible to get better results by choosing one column other than others in a certain attack when the values of b, c and d are unknown, this is because the differential properties of the key schedule algorithm may be used to reduce the key material guess. But in this paper, we can choose any column in Δx_6^O as $(0,0,0,b)$, and have no influence on the attack complexity.

4 A 7-Round Related-Key Impossible Differential attack

Using the above impossible differential, we present an attack on 7-round variant of AES-256 in this section. At first, we assume that the values of a, b, c and d 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 conditions on the related keys to achieve these subkey differences at the end of this section.

In this 7-round attack, we will choose the impossible differential in which Column 3 (the last column) of Δx_6^O is $(0,0,0,b)$.

4.1 The attack procedure

Precomputation: For all the 2^{64} possible pairs of values of the first two columns of x_1^M (ie., $(x_1^M)_{Col(0)}$ and

$(x_1^M)_{Col(1)}$) with difference (a,0,0,0), compute the 8 byte values in bytes 0,3,4,5,9,10,14 and 15 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 attack procedure 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,?,?)), 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. Insert all the ciphertexts $C_1 \in T_1$ and the values $C_2 \in T_2$ into a hash table indexed by bytes 6, 9 and 12.
4. 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 0, 3, 4, 5, 9, 10, 14 and 15 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 and C_2 in the same bin of the hash table derived in Step 3, check whether the corresponding intermediate values are equal. If no, discard the pair.
   (d) For every remaining pair C_1 and C_2, consider the corresponding plaintext pair and compute $P_1 \oplus P_2$ in the eight bytes 0, 3, 4, 5, 9, 10, 14 and 15. Denote the resulting value by P'.
   (e) Access the bin P' in H_p, and 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 0, 3, 4, 5, 9, 10, 14 and 15).
   (f) If A is not empty, output the values in A along with the guess of $(k_7)_{15}$.

4.2 Analysis of the attack complexity

There are m plaintexts each in S_1 and S_2, which can form m^2 possible ciphertext pairs (C_1,C_2). In Step 3, the filtering is done using a 24-bit condition, thus there are about $2^{-24}m^2$ pairs in each bin of the hash table. In Step 4, 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 out of the 2^{64} candidates. Therefore, the expected number of remaining subkeys is $2^{64}(1-1/2^{64})^{m'}$ in Step 4(f). And for $m'=2^{70}$, the expected number is about $e^{-20}=2^{-28.85}$, so we can expect that only the right subkey will remain. Hence, we get the value of 72 subkey bits. In order to get $m'=2^{70}$, 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 of the attack is dominated by Step 4(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 any 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 SubByte operation, so there are 127 possible values of b given the value of a. Similarly, c is also the result of application of SubByte operation, so there are 127 possible values of c given the value of b.

Hence, we need to repeat the attack for all the values of b and c. Therefore, the total time complexity is multiplied by 2^{14}, 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^{87} encryptions, and the required memory is 2^{69} bytes.

5 Four 8-Round Related-Key Impossible Differential attacks

In this section, we will give four variant attacks on 8-round AES-256. All the four attacks presented are based on the 7-round attack in Section 4. As in Refs.[2,3], 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 bits 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. 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 $(\Delta x_6^O)_{Col(1)}=(0,0,0,b)$, then after the SubBytes and ShiftRows operations in the 7'th round, bytes 1, 4, 14 must be zero. Next, applying the key addition with w_7, we can get $(\Delta x_7^W)_1=(w_7)_1$, $(\Delta x_7^W)_4=(w_7)_4$ and $(\Delta x_7^W)_{14}=(w_7)_{14}$.

In order to satisfy the above conditions and make less subkey material guess, we treat only ciphertext pairs that have certain properties. Take for example, to make $(\Delta x_7^W)_4=(w_7)_4$, we only choose ciphertext pairs that satisfy $(\Delta x_7^O)_{Col(1)}=(0,0,0,z_7)$, where z_7 is uniquely determined by $(\Delta w_7)_4$ to make the above condition hold, ie. $MC^{-1}(0,0,0,z_7)=((\Delta w_7)_4,?,?,?)$. Similarly, we can decide the values of two bytes z_0 and z_{12}, which make $(\Delta x_7^W)_1=(w_7)_1$ and $(\Delta x_7^W)_{14}=(w_7)_{14}$ respectively.

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

The First Attack Guess all the 16 bytes of k_8, then peel off the 8'th round, and applying the above 7-round attack.

Here we can use the differential properties of the key schedule algorithm. The value of c can be determined by b and $(k_6)_{15}=(k_8)_{11}\oplus(k_8)_{15}$. Hence, we only need to repeat the attack for all the possible values of b and d respectively.

Here we require $m'=2^{72}$, then the probability that some wrong subkey guess remains is about $2^{64}e^{-256}=2^{-304}$. Therefore the expected remained data is approximately $2^{-304}2^{136}=2^{-168}$, thus we can expect that only the right subkey will remain. Hence, 2^{53} chosen plaintexts is needed, and the time complexity is $2^{14}\times2^{73}\times2^{128}=2^{215}$ encryptions, and the required memory is 2^{69} bytes.

The Second Attack Guess bytes 0, 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 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 1, 2 and 3. And treat only 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 0 is z_0. This filtering is done using an 8-bit condition. Thus, the remaining ciphertext pairs satisfy the condition that byte 5 in Δx_6^O is zero. Next, calculate the difference in bytes 4 and 14 of Δx_7^W and check whether it equals to $(w_7)_4$ and $(w_7)_{14}$ respectively. This filtering thus makes bytes 4 and 6 in Δx_6^O zero too, and uses a 16-bit condition. Then, guess byte 11 of w_7 and continue partial decryption to find out whether $(\Delta x_6^O)_7=b$ holds, which is done using an 8-bit condition. After this filtering, the remaining ciphertext pairs can be used to discard the wrong subkey guesses as in the 7-round attack.

In this variant of the attack, we guess a total of 112 subkey bits. And a portion of 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 c can be determined by b and $(k_6)_{15}=(k_8)_{11}\oplus(k_8)_{15}$. Hence, we only need to repeat the attack for all the possible values of b and d

respectively.

Choose a structure of 2^{64} plaintexts which differ only at the eight bytes 0, 3, 4, 5, 9, 10, 14 and 15, and having all the possible values in these bytes. Encrypt it under 2 related keys each, which is equivalent to 2^{65} chosen plaintexts. One structure proposes about $2^{64}\times 2^{64}\times 1/2=2^{127}$ pairs of plaintexts. About 1 structure is needed to get 2^{71} data pairs for every guess of the 112 bits in the last two rounds, which can be used to delete the wrong subkey guess. Each pair discards one possible value for the eight byte guess of subkey k_0 on average. Therefore, the probability that some wrong subkey guess remains is at most $2^{64}e^{-128}\approx 2^{-120}$. Hence, the expected number of subkey suggestions is approximately $2^{-120}2^{112}=2^{-8}$. Hence, with a high probability only the right value remains. The data complexity of this attack is about 2^{64} chosen plaintexts. The time complexity is about $2^{14}\times 2^{71}\times 2^{112}/2^{6}=2^{191}$ and the required memory is about 2^{69} bytes.

The Third Attack Guess bytes 0, 2, 3, 5, 6, 8, 9, 11, 12 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 0, 7 are z_0 and z_7 respectively. This filtering is done using a 16-bit condition. Thus, the remaining ciphertext pairs satisfy the conditon that bytes 4 and 5 in Δx_6^O are all zero. Next, calculate the difference in byte 14 of Δx_7^W and check whether it equals to $(w_7)_{14}$. This filtering uses an 8-bit condition. Then, guess byte 11 of w_7 and continue partial decryption to find out whether $(\Delta x_6^O)_7=b$ holds. This is done using an 8-bit condition. After this filtering, the remaining ciphertext pairs can be used to discard the wrong subkey guesses as in the 7-round attack.

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 the wrong subkey guesses.

As in the first attack, the value of c can be determined by b and subkeys guess of k_8. Hence, we only need to repeat the attack for all the possible values of b and d respectively.

About 2^{24} structures are needed to get about 2^{71} data pairs which can be used to delete the wrong subkey guesses. Hence, the data complexity of this attack is about 2^{88} chosen plaintexts. The time complexity is about $2^{14}\times 2^{71}\times 2^{88}/2^{6}=2^{167}$.

The Fourth Attack Guess bytes 0, 2, 5, 8, 11, 12 and 15 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 0, 7 and 12 are z_0, z_7 and z_{12} respectively. This filtering is done using a 24-bit condition. Thus, the remaining ciphertext pairs all satisfy that bytes 4,5,6 in Δx_6^O are all zero. Then, guess byte 11 of w_7 and continue partial decryption to find out whether $(\Delta x_6^O)_7=b$ holds. This is done using an 8-bit condition. After this filtering, the remaining ciphertext pairs can be used to discard wrong subkey guesses as in the 7-round attack.

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 c can be determined by b and $(k_6)_{15}=(k_8)_{11}\oplus(k_8)_{15}$. Hence, we only need to repeat the attack for all the possible values of b and d respectively.

About 2^{48} structures are needed to get about 2^{71} data pairs which can be used to delete the wrong subkey guesses. Hence, the data complexity of this attack is about 2^{112} chosen plaintexts. The time complexity is about $2^{14}\times 2^{71}\times 2^{64}/2^{6}=2^{143}$.

6 Summary

Using related-key cryptanalysis, better results are achieved against the reduced-round AES compared with other traditional cryptanalysis approaches up to now. This fact reflects some weaknesses of the key schedule algorithm of AES.

In this paper, we studied the ability of AES-256 against the related-key impossible differential attack. Among the results, we give an attack against 7-round AES-256, and four attacks against 8-round AES-256. The carefully chosen related-key difference makes our attack start from the very beginning, not from the third round as in Ref.[2], which reduces the attack complexity by a factor of 2^7 at least. Also the key schedule is used in key byte guessing to reduce the time complexity. The results are summarized in Table 1.

We have conceived and tried to attack 9-round AES-256, but failed. As the subkeys difference presented in this paper can be extended to more rounds, so more research is anticipated on AES-256 against the related-key impossible differential attack, perhaps 9 or more rounds will be reached.

References:

[1] Advanced encryption standard (AES). FIPS Publication 197, 2001. http://csrc.nist.gov/encryption/aes

[2] Biham E, Dunkelman O, Keller N. Related-Key impossible differential attacks on 8-round AES-192. In: Proc. of the CT-RSA 2006. LNCS 3860, Springer-Verlag, 2006. 21–33.

[3] Zhang WT, Wu WL, Zhang L, Feng DG. Improved related-key impossible differential attacks on reduced-round AES-192. In: SAC 2006—Proc. of the Selected Areas in Cryptography 2006. LNCS Series, Springer-Verlag, 2006.

[4] Biham E. New types of cryptanalytic attacks using related keys. Advances in Cryptology-EUROCRYPT'93. LNCS 765, Springer-Verlag, 1994. 398–409.

[5] Kelsey J, Schneier B, Wagner D. Related-Key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In: Proc. of the Information and Communication Security'97. LNCS 1334, Springer-Verlag, 1997. 233–246.

[6] Biham E, Biryukov A, Shamir A. Cryptanalysis of skipjack reduced to 31 rounds. In: Advances in Cryptology, Proc. of the EUROCRYPT'99. LNCS 1592, Springer-Verlag, 1999. 12–23.

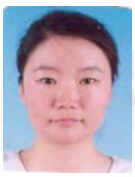
[7] Phan RCW. Impossible differential cryptanalysis of 7-round advanced encryption standard (AES). Information Processing Letters, 2004,91(1):33–38.

[8] Biham E, Dunkelman O, Keller N. Related-Key boomerang and rectangle attacks. In: Advances in Cryptology, Proc. of the EUROCRYPT 2005. LNCS 3494, Springer-Verlag, 2005. 507–525.

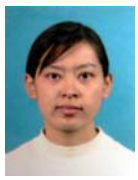
[9] Ferguson N, Kelsey J, Lucks S, Schneier B, Stay M, Wagner D, Whiting D. Improved cryptanalysis of rijndael. In: Proc. of the Fast Software Encryption 8. LNCS 1978, Springer-Verlag, 2001. 213–230.

[10] Jakimoski G, Desmedt Y. Related-Key differential cryptanalysis of 192-bit key AES variants. In: Proc. of the Selected Areas in Cryptography 2003. LNCS 3006, Springer-Verlag, 2004. 208–221.

[11] Hong SK, Kim JS, Kim G, Lee SJ, Preneel B. Related-Key rectangle attacks on reduced versions of SHACAL-1 and AES-192. In: Proc. of the Fast Software Encryption 12. LNCS 3557, Springer-Verlag 2005. 368–383.



ZHANG Wen-Tao was born in 1977. She is a lecturer at the Graduate University of Chinese Academy of Sciences. Her research area is cryptology.



ZHANG Lei was born in 1981. She is a Ph.D. candidate at the Institute of Software, the Chinese Academy of Sciences. Her research area is cryptology.

WU Wen-Ling was born in 1966. She is a researcher and doctoral supervisor at the Institute of Software, the Chinese Academy of Sciences. Her research area is cryptology.