Related-Key Impossible Differential Attacks on 8-Round AES-192

Eli Biham[1], Orr Dunkelman*[1], Nathan Keller[2]

[1]Computer Science Department, Technion.
Haifa 32000, Israel
{biham,orrd}@cs.technion.ac.il
[2]Einstein Institute of Mathematics, Hebrew University.
Jerusalem 91904, Israel
nkeller@math.huji.ac.il

Abstract. In this paper we examine the strength of AES against the related-key impossible differential attack, following the work of Jakimoski and Desmedt [12]. We use several additional observations to substantially improve the data and time complexities of their attacks. Amongst our results, we present a related-key attack on 7-round AES-192 with data complexity of 2^{56} chosen plaintexts (instead of 2^{111}). Our attack on 8-round AES-192 has data complexity of $2^{68.5}$ chosen plaintexts (instead of 2^{88}). The time complexities of our attacks is also substantially lower than the time complexities of previous attacks.

Keywords: AES, related-key differentials, impossible differentials

1 Introduction

The *Advanced Encryption Standard* [9] is a 128-bit block cipher with variable key length (128, 192, and 256-bit keys are allowed). Since its selection, AES gradually became one of the most worldwide used block ciphers. Therefore, a constant evaluation of its security with respect to various cryptanalytic techniques is required. AES was already analyzed in many papers, each using different attacks [5, 6, 8, 10–12].

Related-key attacks [1] consider the information that can be extracted from two encryptions using related (but unknown) keys. In the attack, the attacker uses weaknesses of the encryption function and of the key schedule algorithm to derive information on the unknown keys. Related-key differential attacks [13] study the development of differences in two encryptions under two related keys and use them to derive the actual values of the keys. Usually the attacker exploits differential relations that hold with a relatively high probability, like in ordinary differential attacks [4]. However, differential relations holding with a very low (or zero) probability can also be used [2, 3, 12]. In this case, the attack is called *related-key impossible differential attack*.

* The research presented in this paper was supported by the Clore scholarship programme.

Cipher	Number of	Complexity		Number of	Attack Type
	Rounds	Data	Time	Keys	& Source
AES-192	7	2^{92} CP	2^{186}	1	Imp.Diff. [8]
(12 rounds)	7	$19 \cdot 2^{32}$ CP	2^{155}	1	SQUARE [10]
	8	$2^{128} - 2^{119}$ CP	2^{188}	1	SQUARE [10]
	7	2^{111} RK-CP	2^{116}	2	RK Imp.Diff. [12]
	8	2^{88} RK-CP	2^{183}	2	RK Imp.Diff. [12]
	8	$2^{86.5}$ RK-CP	$2^{86.5}$	4	RK Rectangle [11]
	9	2^{86} RK-CP	2^{125}	256	RK Rectangle [5]
	7	2^{56} RK-CP	2^{94}	32	RK Imp.Diff.;Sect. 3
	8	2^{116} RK-CP	2^{134}	32	RK Imp.Diff.;Sect. 4
	8	2^{92} RK-CP	2^{159}	32	RK Imp.Diff.;Sect. 4
	8	$2^{68.5}$ RK-CP	2^{184}	32	RK Imp.Diff.;Sect. 4

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

Table 1. Summary of the Previous Attacks and of Our New Attacks

In this paper we examine the security of AES against related-key impossible differential attacks. We concentrate on the 192-bit key version of AES (AES-192) since in this variant the diffusion of the key schedule is slower than in the other versions and thus the potential vulnerability to related-key attacks is bigger.

The relatively weak key schedule of AES-192 has inspired much research: In [12] Jakimoski and Desmedt presented a related-key differential attack applicable up to a 6-round AES-192 (out of the 12 rounds). An improved version of the attack (also presented in [12]) uses truncated differentials and is applicable up to a 7-round version. In addition, Jakimoski and Desmedt [12] devised several related-key impossible differential attacks that are applicable up to an 8-round AES-192. In [11] Hong et al. presented a related-key rectangle attack applicable up to an 8-round AES-192. The best known related-key attack on AES-192 was devised by Biham et al. [5] and it is applicable to a 9-round variant of the cipher.

For comparison, the best attack on AES-192 not under the related-key model is a SQUARE attack presented in [10]. It can attack up to 8 rounds of AES-192, using almost the entire code book. The time complexity of this attack is 2^{188} encryptions.

In this paper we present several new related-key impossible differential attacks. The attacks use the 5.5-round impossible differential suggested by Jakimoski and Desmedt [12]. However, by making additional observations on the behavior of the key schedule, we can reduce the data complexity of our attacks by a factor of 2^{55} for the 7-round attack, and by a factor of $2^{19.5}$ for the 8-round attack. The time complexity is also reduced significantly. We summarize our results along with previously known results in Table 1.

This paper is organized as follows: In Section 2 we give a brief description of AES. In Section 3 we describe the new related-key attack on 7-round AES-

192. In Section 4 we extend the 7-round attack to attacks on 8-round AES-192. Finally, Section 5 summarizes this paper.

2 Description of AES

The advanced encryption standard [9] is an SP-network that supports key sizes of 128, 192, and 256 bits. The 128-bit plaintexts are treated as byte matrices of size 4x4, where each byte represents a value in $GF(2^8)$. An AES round applies four operations to the state matrix:

- SubBytes (SB) – applying the same 8x8 S-box 16 times in parallel on each byte of the state,
- ShiftRows (SR) – cyclic shift of each row (the i'th row is shifted by i bytes to the left),
- MixColumns (MC) – multiplication of each column by a constant 4x4 matrix over the field $GF(2^8)$, and
- 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 (using a whitening key). As all other works on AES, we shall assume that reduced-round variants also have the MixColumns operation omitted from the last round.

The number of rounds depends on the key length: 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. The rounds are numbered $0, \ldots, Nr-1$, where Nr is the number of rounds ($Nr \in \{10, 12, 14\}$). For sake of simplicity we shall denote AES with n-bit keys by AES-n, i.e., AES with 192-bit keys (and thus with 12 rounds) is denoted by AES-192.

The key schedule of AES-192 takes a 192-bit key and transforms it into 13 subkeys of 128 bits each. The subkey array is denoted by $W[0, \ldots, 51]$, where each word of $W[\cdot]$ consists of 32 bits. The first six words of $W[\cdot]$ are loaded with the user supplied key. The remaining words of $W[\cdot]$ are updated according to the following rule:

- For $i = 6, \ldots, 51$ do
  - If $i \equiv 0 \bmod 6$ then $W[i] = W[i-6] \oplus SB(W[i-1] \lll 8) \oplus RCON[i/6]$,
  - else $W[i] = W[i-1] \oplus W[i-6]$.

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

The best known attack on AES-192 is a SQUARE attack on 8 rounds [10]. The attack requires almost the entire code book ($2^{128} - 2^{119}$ chosen plaintexts) and has a time complexity equivalent to 2^{188} encryptions. The SQUARE attack applied to 7-round AES-192 requires $19 \cdot 2^{32}$ chosen plaintexts and has a time complexity of 2^{155} encryptions.

The best impossible differential attack on AES-192 is on 7-round AES-192 [8]. Its data complexity is 2^{92} chosen plaintexts and its time complexity is 2^{186} encryptions.

There are several related-key attacks on AES-192. A related-key impossible differential attack on an 8-round variant is presented in [12]. This attack requires 2^{88} related-key chosen plaintexts and has a running time of 2^{183} encryptions. The attack uses two related keys.

A related-key rectangle attack on 8-round AES-192 using four related keys is presented in [11]. It requires $2^{86.5}$ chosen plaintexts (encrypted under four keys) and has a time complexity equivalent to $2^{86.5}$ encryptions.

Another related-key rectangle attack on AES-192 is presented in [5]. This attack can be applied up to nine rounds using 2^{86} related-key chosen plaintexts encrypted under 256 keys. Its time complexity is 2^{125} encryptions.

The related-key attacks exploit a weakness in the key schedule algorithm of AES-192. Unlike AES-128 and AES-256, the key schedule algorithm of AES-192 applies a nonlinear component (SubBytes) once every six key words (or once every round and a half), instead of once every four key words (once every round). This leads to the introduction of better and longer related-key differentials.

2.1 Notations Used in the Paper

In our attacks we use the following notations: x_i^I denotes the input of round i, while x_i^S, x_i^{Sh}, x_i^M, and x_i^O denote the intermediate values after the application of SubBytes, ShiftRows, MixColumns, and AddRoundKey operations of round i, respectively. Of course, the relation $x_{i-1}^O = x_i^I$ holds.

We denote the subkey of round i by subscript k_i, and the first (whitening) key is k_{-1}, i.e., the subkey of the first round is 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, by first XORing the data with an equivalent key and only then applying the MixColumns operation. We denote the equivalent subkey for the changed version by w_i, i.e., $w_i = MC^{-1}(k_i)$.

We denote the z'th column of x_i by $x_{i,Col(z)}$, i.e., $w_{0,Col(0)} = MC^{-1}(k_{0,Col(0)})$. We also denote the byte in the y'th row and the z'th column of the state matrix x (of round i) by byte $x_{i,y,z}$ where $y, z \in \{0, 1, 2, 3\}$. For example, $x_{2,0,3}^M$ denotes the fourth byte in the first row of the intermediate value after the application of the MixColumns transformation in round 2. Another notation for bytes of some intermediate state x_i is an enumeration $\{0, 1, 2, \ldots, 15\}$ where the byte $x_{i,y,z}$ corresponds to byte $4z + y$ of x_i.

In the paper we also use the notation x_i= $((x_{i,Col(0)}), (x_{i,Col(1)}),(x_{i,Col(2)}), (x_{i,Col(3)}))$. The column j of x_i is represented as $(x_{i,0,j}, x_{i,1,j},x_{i,2,j},x_{i,3,j})$.

3 Related-Key Impossible Differential Attacks on 7-Round AES-192

3.1 A 5.5-round Related-key Impossible Differential of AES-192

First we recall the related-key impossible differential presented in [12] that we use in our attacks. The impossible differential starts at the middle of round 2 and

Round (i)	$\Delta k_{i,Col(0)}$	$\Delta k_{i,Col(1)}$	$\Delta k_{i,Col(2)}$	$\Delta k_{i,Col(3)}$
-1	$(0,0,0,f)$	$(0,0,0,0)$	$(a,0,0,0)$	$(a,0,0,0)$
0	$(a,0,0,0)$	$(a,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$
1	$(a,0,0,0)$	$(0,0,0,0)$	$(a,0,0,0)$	$(0,0,0,0)$
2	$(0,0,0,0)$	$(0,0,0,0)$	$(a,0,0,0)$	$(a,0,0,0)$
3	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$
4	$(a,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$
5	$(0,0,0,0)$	$(0,0,0,0)$	$(a,0,0,0)$	$(a,0,0,0)$
6	$(a,0,0,0)$	$(a,0,0,0)$	$(0,0,0,b)$	$(0,0,0,b)$
7	$(a,0,0,b)$	$(0,0,0,b)$	$(a,0,0,b)$	$(0,0,0,b)$
8	$(0,0,c,b)$	$(0,0,c,0)$	$(a,0,c,b)$	$(a,0,c,0)$
9	$(0,0,c,b)$	$(0,0,c,0)$	$(0,d,c,b)$	$(0,d,0,b)$

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

Table 2. Subkey Differences Required for the 5.5-Round Impossible Differential

ends just after round 7. Note that in [12] the differential was used in rounds 0–4 (including the whitening key).

Consider rounds 2–7 of AES-192. Throughout the attack we assume that the subkey differences in these six rounds and the surrounding rounds are as presented in Table 2. We shall address the conditions on the difference between the keys to achieve these subkey differences later.

The related-key impossible differential is of 5.5 rounds, and is built in a miss-in-the-middle manner [2]. A 4.5-round related-key differential with probability 1 is "concatenated" to a 1-round related-key differential with probability 1, in the inverse direction, where the intermediate differences contradict one another. The 5.5-round related-key impossible differential is

$$\Delta x_2^M = ((0,0,0,0),(0,0,0,0),(a,0,0,0),(a,0,0,0)) \not\to$$
$$\Delta x_7^O = ((?,?,?,?),(0,0,0,b),(?,?,?,?),(?,?,?,?)),$$

where ? denotes any value.

The first 4.5-round differential is obtained as follows: The input difference $\Delta x_2^M = ((0,0,0,0),(0,0,0,0),(a,0,0,0),(a,0,0,0))$ is canceled by the subkey difference at the end of round 2. The zero difference $\Delta x_3^I = 0$, is preserved through all the operations until the AddRoundKey operation of round 4, and hence $\Delta x_4^M = 0$. The subkey difference in k_4 becomes the data difference, i.e., $\Delta x_5^I = ((a,0,0,0),(0,0,0,0),(0,0,0,0),(0,0,0,0))$. This difference is in a single byte, and thus, the difference after the first three operations of round 5 is in all the four bytes of a column, i.e., $\Delta x_5^M = ((y,z,w,v),(0,0,0,0),(0,0,0,0),(0,0,0,0))$ where y,z,w,v are unknown non-zero byte values. After the subkey addition this difference becomes $\Delta x_5^O = ((y,z,w,v),(0,0,0,0),(a,0,0,0),(a,0,0,0)$.

This difference evolves after the SubBytes and ShiftRows of round 5 into $\Delta x_6^{Sh} = ((y',0,0,0),(0,0,0,v'),(a',0,w',0),(a'',z',0,0))$, where y',z',w',a', and a'' are unknown non-zero values. Hence, Δx_6^M =$((N,N,N,N)$, (N,N,N,N), $(?,?,?,?)$, $(?,?,?,?))$ where N denotes non-zero differences (possibly distinct). Finally, after

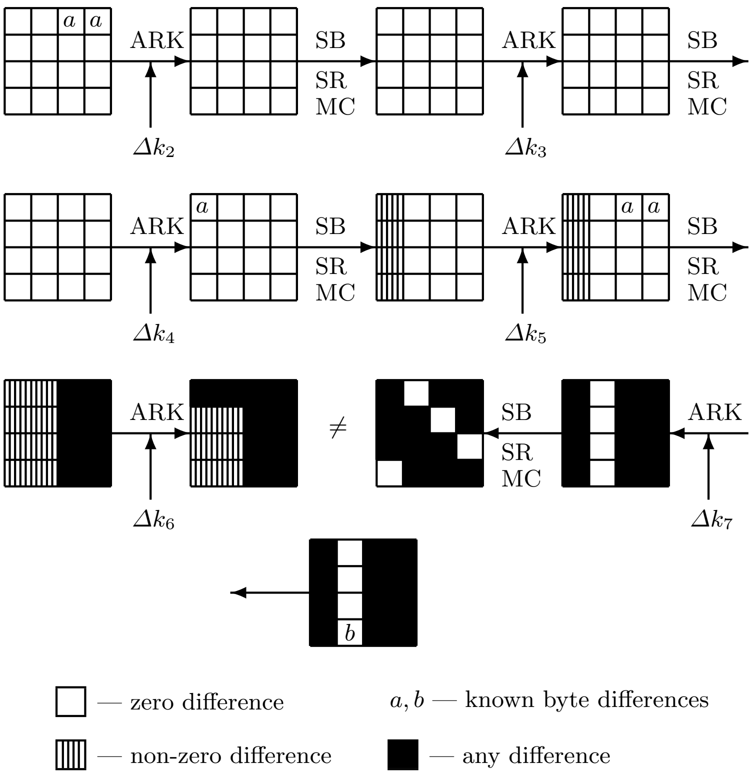

Fig. 1. The 5.5-Round Impossible Differential Used in the Attack

the key addition this difference evolves to $\Delta x_6^O = ((?, N, N, N), (?, N, N, N), (?, ?, ?, ?), (?, ?, ?, ?))$.

Hence, the input difference $\Delta x_2^M = ((0, 0, 0, 0), (0, 0, 0, 0), (a, 0, 0, 0), (a, 0, 0, 0))$ evolves with probability one into a non-zero difference in bytes 1,2,3,5,6, and 7 of x_6^O. The propagation of the differences is shown in Figure 1.

The second differential ends after round 7 with output difference $\Delta x_7^O = ((?, ?, ?, ?), (0, 0, 0, b), (?, ?, ?, ?), (?, ?, ?, ?))$. When rolling back this difference through the AddRoundKey operation, we get the difference $\Delta x_7^M = ((?, ?, ?, ?), (0, 0, 0, 0), (?, ?, ?, ?), (?, ?, ?, ?))$, which leads to a zero output difference of the MixColumns operation in the second column. Hence, the input difference to round 7 is $\Delta x_7^I = ((?, ?, ?, 0), (0, ?, ?, ?), (?, 0, ?, ?), (?, ?, 0, ?))$. This difference contradicts the first differential as with probability one $x_{6,3,0}^O = x_{7,3,0}^I$ has a non-zero difference while the second differential predicts that this byte has a zero difference with probability 1. This contradiction is emphasized in Figure 1.

3.2 A 7-round Related-Key Impossible Differential Attack

Using the above impossible differential we can attack a 7-round variant of AES-192. We attack rounds 2–8 of the cipher, using a pair of related keys that has the subkey differences described earlier. Our attack is based on the following two observations:

1. If the input difference of the differential holds, then the plaintext difference in eight of the 16 bytes is known, while in the other eight bytes almost any difference can be used. Thus, our attack can use structures in order to bypass round 2.
2. It is sufficient to guess only one subkey byte of the last round ($k_{8,3,2}$) in order to check out whether the output difference of the impossible differential holds.

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.

For sake of simplicity, we currently assume that the values of a, b, c and f are known, i.e., we have two related keys K_1 and K_2 with the required subkey differences. This does not hold, but we shall deal with this issue later.

In order to make the attack faster we first perform a precomputation: For all the 2^{64} possible pairs of values of the two last columns of x_2^M, i.e., $x_{2,Col(2)}^M$ and $x_{2,Col(3)}^M$ with difference $((a,0,0,0),(a,0,0,0))$, compute the values of the eight bytes $1,2,6,7,8,11,12$, and 13 of x_2^I. Store the pairs of 8-byte values in a hash table H_p indexed by the XOR difference in these bytes.

The algorithm of the attack 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 $P_2 \in S_2$, $P_1 \oplus P_2 = (?,0,0,?)$,$(?,?,0,0)$,$(a,?,?,0)$,$(0,0,?,?)$, 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 $C_2^* = C_2 \oplus ((0,0,0,0),(0,0,0,0),(0,0,0,b)$ $,(0,0,c,0))$.
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 1,4, and 14.
5. Guess the value of the subkey byte $k_{8,3,2}$ 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_1.
   (b) Decrypt the byte $x_{8,3,2}$ in all the ciphertexts to get the intermediate values before the subkey addition at the end of round 7.
   (c) For every pair C_1, C_2^* in the same bin of the hash table, check whether the corresponding intermediate values 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) 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 $1, 2, 6, 7, 8, 11, 12$, and 13).[1]

(f) If A is not empty, output the values in A along with the guess of $k_{8,3,2}$

The total amount of possible pairs C_1, C_2^* is m^2. The filtering in Step 4 is done using a 24-bit condition, thus, we expect about $2^{-24}m^2$ pairs in every bin of the hash table. In Step 5 we have an additional 8-bit filtering (for every possible value of $k_{8,3,2}$ separately) and therefore about $2^{-32}m^2$ pairs remain for a given subkey guess of $k_{8,3,2}$. Each pair deletes 1 subkey candidate on average out of the 2^{64} candidates. Hence, after $m' = 2^{-32}m^2$ pairs the expected number of remaining subkeys is $2^{64}(1 - 1/2^{64})^{m'}$. For $m' = 2^{70}$ the expected number is about e^{-20} and we can expect that only the right subkey remains. Moreover, for wrong guesses of $k_{8,3,2}$ no subkey is expected to 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.

The time complexity of the attack is dominated by Step 5(e). In this step m' 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 $k_{8,3,2}$). Therefore, the time complexity is 2^{79} memory accesses, which are equivalent to about 2^{73} encryptions. The precomputation requires about 2^{62} encryptions and the required memory is about 2^{69} bytes. The data complexity of the attack is 2^{52} chosen plaintexts.

Note that in the attack we assumed that the values of a, b, c, and f are known. We deal with these values and add the required corrections in the attack in the next subsection.

3.3 Overcoming the Nonlinearity of the Key Schedule Algorithm

Our attack uses a pair of related keys such that the subkey differences between them are presented in Table 2. However, due to the nonlinearity of the key schedule there is no key difference that can assure these subkey differences. In particular, while the value a can be chosen by the attacker, the values b and c that are results of application of SubBytes transformation are unknown given the initial key difference. This problem was already dealt in [12]. The solution we present in Section 4 is similar to the one presented in [12].

In our attack we have an additional problem: Since we have a round before the differential (instead of adding the round after the differential, as was done in [12]), we cannot choose $\Delta k_2 = ((0,0,0,0),(0,0,0,0),(a,0,0,0),(a,0,0,0))$ to be the first four columns of the key difference.

The key difference is $\Delta k = ((0,0,0,f),(0,0,0,0),(a,0,0,0),(a,0,0,0),(a,0,0,0),(a,0,0,0))$, and as noted before, the value f is unknown. In comparison, in [12] there are no unknown bytes in the key difference since the attack starts in round 0

[1] Not all entries contain values. It is expected that only 36% of these entries suggest values to discard. However, once an entry of H_p is non-empty, it suggests at least two values to discard.

of AES. Due to the differential properties of the SubBytes transformation, f can assume 127 values with approximately the same probability.

The values of b and c are unknown but since the both of them are results of application of the SubBytes transformation, we know that given a, there are only 127 possible values of b, and given b there are only 127 possible values of c. Hence, we can repeat the attack for all the values of b and for every value of b, for all the values of c. The expected number of remaining wrong suggestions in the original attack (about e^{-20}) assures that for wrong guesses of b and c, with high probability no subkey will be suggested.

Therefore, the total time complexity of the attack is multiplied by 2^{21} since we repeat the attack for all the possible values of f, b, c. The data and memory requirements remain unchanged.

However, if we just try all the possible values of f, we need to encrypt the plaintexts under 128 related keys since changing the value of f changes the key difference. We can partially solve this problem by using *structures of keys*. We take two structures of 16 keys each such that the difference between two keys in different structures is $\Delta k = ((0,0,0,?),(0,0,0,0),(a,0,0,0),(a,0,0,0),(a,0,0,0),(a,0,0,0))$, for a random value ?. Such structures are achieved by fixing all the bytes except for one in all the keys and choosing the value of this byte randomly. The structures induce 256 pairs of keys, and for each pair of keys, we perform the attack described above. Since the differences in the byte marked by ? are random, the probability that after the SubBytes transformation the difference will be a for at least one pair is approximately $1-(1-1/256)^{256} = 1-1/e = 0.63$. Hence, with probability 0.63 we will get the required subkey differences for at least one pair of keys and for this pair the attack succeeds.

We can improve the time complexity by a factor of 2 by performing the attack only for those pairs of keys for which the difference in the marked byte can be transformed to the difference a by the SubBytes transformation. There are 127 such differences and thus the attack is expected to be performed only 127 times.

The total complexity of the attack is therefore the following: The data complexity is 2^{52} plaintexts encrypted under 16 keys each, or a total of 2^{56} chosen plaintexts, the time complexity is 2^{94} encryptions and the required memory is 2^{69} bytes.

4 Three 8-round Impossible Differential Attacks

In this section we present three attacks on 8-round AES-192. All the three attacks are based on the 7-round attack and the main difference between them is a time-data trade-off.

Consider an 8-round version of AES-192 starting with round 2. In all the 8-round attacks we guess part of the last round subkey (k_9), peel off the last round and apply the 7-round attack. For the description of the attacks it is more convenient to change the order of the MixColumns and the AddRoundKey operations at the end of round 8. As mentioned earlier this is done by replacing the subkey k_8 with an equivalent subkey w_8. Note that since the subkey difference

Δk_8 is known, the difference between the corresponding equivalent subkeys Δw_8 is also known.

In the 7-round attack we have to check whether the difference in three bytes in the beginning of round 8 is zero and whether the difference in one specific byte is b. A zero difference at the beginning of round 8 remains such a difference until the end of the round (up to the MixColumns operation), and thus we have to check whether the difference in the corresponding three bytes in the beginning of the last round is zero. For the fourth byte, we compute its difference of the pair at the beginning of round 8.

4.1 The 8-Round Attacks

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

1. Guess 12 bytes of the last round subkey (k_9) and partially decrypt these bytes in the last round. The difference in the remaining four bytes is unknown. To know this difference without guessing more subkey material, we treat only ciphertext pairs that have zero difference in these bytes. This condition allows us to use only 2^{-32} of the possible ciphertext pairs, but this price is well worth it. As the difference Δx_8^O is known, we check whether the difference in bytes 1,4, and 14 is zero. Then, we guess one subkey byte ($w_{8,3,2}$) and continue partial decryption to find out whether the difference b holds. If all the required differences hold then this ciphertext pair 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 168 subkey bits. This leads to a very high time complexity, but to a relatively low data complexity.
2. Guess eight bytes of k_9 and use only the pairs for which the difference in the eight ciphertext bytes which are XORed with an unguessed subkey is zero. Again, after partially decrypting the ciphertexts, we guess the byte $w_{8,3,2}$ and then we are able to check the differences in the four required bytes.
   In this variant, we guess 136 subkey bits, but only a portion of 2^{-64} of the pairs can be used in the attack and thus the data complexity is higher.
3. Guess only four bytes of k_9 and use only the pairs for which the difference in the 12 ciphertext bytes that are XORed with an unguessed subkey is zero. After the partial decryption, we guess the key byte $w_{8,3,2}$ in order to check whether the impossible differential can be "satisfied".
   In this variant of the attack, we guess only 104 subkey bits, leading to a substantially lower time and memory requirements. On the other hand, we use a portion of only 2^{-96} of the possible pairs, which increases the data complexity.

Since the attacks are similar, we present in detail only the first attack. The complexities of all the three attacks are summarized in Table 1.

Just like before, we assume that the values a, b, c, d and f are known. We shall address this issue after the attack.

In the first version of the attack we guess the values of bytes 0,2,3,5,6,7,8,9,10, 12,13, and 15 of k_9 and byte $w_{8,3,2}$. The values of these subkey bytes allow us to partially decrypt the last round in Columns $0,2$, and 3 where byte $x^O_{8,3,2}$ is also partially decrypted through round 8.[2] Then, we can perform the 7-round attack for every guess. Note that we can also choose other three columns to guess as long as Column 2 is included. Our choice is optimal when the values of b,c,d are not known.

The attack 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 $P_2 \in S_2$, $P_1 \oplus P_2 = (?,0,0,?),(?,?,0,0),(a,?,?,0),$ $(0,0,?,?)$.
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 similarly the ciphertexts of the pool S_2 by T_2.
3. For all ciphertexts $C_2 \in T_2$ compute $C_2^* = C_2 \oplus ((0,0,0,0),(0,0,0,0),(0,0,0,b),$ $(0,0,0,0))$.
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 1,4, and 14.
5. For every guess of the 12 bytes 0,2,3,5,6,7,8,9,10,12,13, and 15 of k_9 and $w_{8,3,2}$ do:
   (a) Initialize a list A of the 2^{64} possible values of the bytes $1,2,6,7,8,11,12,$ and 13 of the subkey k_1.
   (b) Partially decrypt the last two rounds in all of the ciphertexts to obtain $x^I_{9,1,0}, x^I_{9,2,3}$ and $x^I_{8,3,1}$. For all the ciphertexts of T_2, XOR the value of the byte $x^I_{8,3,1}$ with b.
   (c) For all the pairs $C_1 \in T_1, C_2 \in T_2$, such that C_1 and C_2^* collide in the hash table check whether the difference in the three computed bytes equals zero. Otherwise, discard the pair.
   (d) For every remaining pair, consider the corresponding pair of plaintexts and compute $P_1 \oplus P_2$ restricted to the eight bytes $1,2,6,7,8,11,12,$ and 13.
   (e) Access H_p in the entry $P_1 \oplus P_2$ (restricted to the eight bytes) and for every pair (x,y) in the same bin compute the values $P_1 \oplus x$ and $P_1 \oplus y$. Delete these values from the list A.
   (f) If A is not empty, output the guess for the 13 bytes and the list A.

4.2 Analysis of the Attack

The analysis of the attack is similar to the analysis of the 7-round attack. We start with $m = 2^{63.5}$ plaintexts in each pool. The plaintexts compose 2^{127} possible pairs. After the initial filtering 2^{95} pairs remain. For every guess of the 104 bits in the last rounds, about 2^{71} pairs remain after the second filtering. Each pair discards one possible value for the subkey of round 1 on average. Therefore, the probability that some wrong subkey guess remains is at most $2^{64}e^{-128} = 2^{-120}$.

[2] Since we analyze only pairs for which the difference in bytes 1,4,11, and 14 of the ciphertexts is zero, we know also the difference in $x^O_{8,Col(1)}$.

Therefore, the expected number of subkey suggestions (for the 168 subkey bits) is approximately $2^{-120}2^{104} = 2^{-16}$. Hence, with a high probability only the right value remains. The remaining subkey bits can be found using auxiliary techniques.

The time complexity of the attack is dominated by the time complexity of Steps 5(d) and 5(e). For every guess of the 104 bits, we try the 2^{71} possible pairs and for each of these pairs we perform two memory accesses on average. Thus, the time complexity of this stage is about 2^{176} memory accesses, which are equivalent to about 2^{170} encryptions.

Hence, the data complexity of the attack (if b, c, d, and f are known) is $2^{64.5}$ chosen plaintexts, the time complexity is about 2^{170} encryptions and the required memory is about 2^{69} bytes.

However, the values of b, c, d and f are unknown and if we repeat the attack for all the possible guesses, the complexity will be more than the complexity of exhaustive key search.

Here we can use again the differential properties of the key schedule algorithm. We observe that the value of d is determined by the value $k_{9,2,1}$ and the value c is determined by $k_{7,3,3} = k_{9,3,0} \oplus k_{9,3,1}$. All of these subkey bytes are guessed in the beginning of the attack. Hence, for every guess of the 104 bits we have to repeat the attack only for all the possible values of b and f. As in the 7-round attack, the values of f are obtained by using structures of keys. Note that due to the low expected number of remaining subkey candidates for a single application of the attack (2^{-16}), we expect that when the attack is applied 2^{14} times, only a few subkey candidates remain.

Hence, the total complexity of the attack is as follows: The data complexity is $2^{63.5}$ chosen plaintexts encrypted under 32 keys each (or a total of $2^{68.5}$ chosen plaintexts), the time complexity is 2^{184} encryptions and the memory complexity is about 2^{69} bytes.

As mentioned before, we can perform the attack when discarding more pairs in exchange for guessing less subkey material in round 9. By considering only the ciphertext pairs with zero difference in two columns (instead of only one), we reduce the time complexity of the attack to 2^{159}. On the other hand the data complexity is increased to 2^{92} chosen plaintexts. Another possible trade-off is to consider only ciphertext pairs with zero difference in three columns. This leads to an attack that requires a total of 2^{116} chosen plaintexts and has a running time equivalent to 2^{134} encryptions. The complexity of the attacks can be found in Table 1.

5 Summary and Conclusions

In this paper we have presented several new related-key impossible differential attacks on 7-round and 8-round AES-192. The data and time complexities are summarized in Table 1. Our attacks significantly improve the attacks presented in [12], but use different properties of the key schedule of AES-192. Hence, if one could combine the attacks together, then an attack on 9-round AES-192 faster than exhaustive search may be found. However, we could not find such combination at this stage.

In our attack we perform the key recovery in the round before the differential, whereas in [12] only the rounds after the differential are attacked. As a result, our attack has to overcome the nonlinearity of the key schedule. This is achieved by using 32 keys from two *structures of keys* based on the differential properties of the key schedule algorithm.

We conclude that our paper joins a series of papers identifying problems in the key schedule algorithm of AES, and more precisely, in the key schedule algorithm of AES-192. This may be of a concern for the long term security of AES, even though at the moment none of the attacks succeeds in retrieving the key of the full AES-192 better than exhaustive key search.

References

1. Eli Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Journal of Cryptology, vol. 7, number 4, pp. 229–246, Springer-Verlag, 1994.
2. Eli Biham, Alex Biryukov, Adi Shamir, *Miss in the Middle Attacks on IDEA and Khufu*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 124–138, Springer-Verlag, 1999.
3. Eli Biham, Alex Biryukov, Adi Shamir, *Cryptanalysis of Skipjack Reduced to 31 Rounds*, Advances in Cryptology, proceedings of EUROCRYPT '99, Lecture Notes in Computer Science 1592, pp. 12–23, Springer-Verlag, 1999.
4. Eli Biham, Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
5. Eli Biham, Orr Dunkelman, Nathan Keller, *Related-Key Boomerang and Rectangle Attacks*, Advances in Cryptology, proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science 3557, pp. 507–525, Springer-Verlag, 2005.
6. Alex Biryukov, *The Boomerang Attack on 5 and 6-round AES*, proceedings of Advanced Encryption Standard 4, Lecture Notes in Computer Science 3373, pp. 11–16, Springer-Verlag, 2005.
7. Alex Biryukov, David Wagner, *Slide Attacks*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 245–259, Springer-Verlag, 1999.
8. Raphael Chung-Wei Phan, *Impossible Differential Cryptanalysis of 7-round Advanced Encryption Standard (AES)*, Information Processing Letters, Vol. 91, Number 1, pp. 33-38, Elsevier, 2004.
9. Joan Daemen, Vincent Rijmen *The design of Rijndael: AES — the Advanced Encryption Standard*, Springer-Verlag, 2002.
10. Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner, Doug Whiting, *Improved Cryptanalysis of Rijndael*, proceedings of Fast Software Encryption 8, Lecture Notes in Computer Science 1978, pp. 213–230, Springer-Verlag, 2001.
11. Seokhie Hong, Jongsung Kim, Guil Kim, Sangjin Lee, Bart Preneel, *Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192*, proceedings of Fast Software Encryption 12, Lecture Notes in Computer Science 3557, pp. 368–383, Springer-Verlag 2005.

12. Goce Jakimoski, Yvo Desmedt, *Related-Key Differential Cryptanalysis of 192-bit Key AES Variants*, proceedings of Selected Areas in Cryptography 2003, Lecture Notes in Computer Science 3006, pp. 208–221, Springer-Verlag, 2004.
13. John Kelsey, Bruce Schneier, David Wagner, *Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA*, proceedings of Information and Communication Security 1997, Lecture Notes in Computer Science 1334, pp. 233–246, Springer-Verlag, 1997.