Improved Impossible Differential Attacks on Large-Block Rijndael*

Qingju Wang[1,2], Dawu Gu[1], Vincent Rijmen[2], Ya Liu[1], Jiazhe Chen[3], and Andrey Bogdanov[4]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China.
[2] KU Leuven, ESAT/COSIC and iMinds, Belgium.
{qingju.wang,vincent.rijmen}@esat.kuleuven.be, dwgu@sjtu.edu.cn
[3] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, School of Mathematics, Shandong University, Jinan, 250100, China.
[4] Technical University of Denmark, Department of Mathematics, Denmark.
a.bogdanov@mat.dtu.dk

Abstract. In this paper, we present more powerful 6-round impossible differentials for large-block Rijndael-224 and Rijndael-256 than the ones used by Zhang et al. in ISC 2008. Using those, we can improve the previous impossible differential cryptanalysis of both 9-round Rijndael-224 and Rijndael-256. The improvement can lead to 10-round attack on Rijndael-256 as well. With $2^{198.1}$ chosen plaintexts, an attack is demonstrated on 9-round Rijndael-224 with $2^{195.2}$ encryptions and $2^{140.4}$ bytes memory. Increasing the data complexity to 2^{216} plaintexts, the time complexity can be reduced to 2^{130} encryptions and the memory requirements to $2^{93.6}$ bytes. For 9-round Rijndael-256, we provide an attack requiring $2^{229.3}$ chosen plaintexts, 2^{194} encryptions, and $2^{139.6}$ bytes memory. Alternatively, with $2^{245.3}$ plaintexts, an attack with a reduced time of $2^{127.1}$ encryptions and a memory complexity of $2^{90.9}$ bytes can be mounted. With $2^{244.2}$ chosen plaintexts, we can attack 10-round Rijndael-256 with $2^{253.9}$ encryptions and $2^{186.8}$ bytes of memory.

Keywords: block cipher, impossible differential attack, Rijndael, large block

1 Introduction

Rijndael [11] is a block cipher designed by Joan Daemen and Vincent Rijmen built upon a Substitution Permutation Network (SPN). A subset of Rijndael variants has been standardized as Advanced Encryption Standard (AES) by the U.S. National Institute of Standards and Technology (NIST) [14] in 2002.

* This work was supported by the National Natural Science Foundation of China (No. 61073150), and in part by the Research Council K.U.Leuven: GOA TENSE, the IAP Program P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and in part by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II.

Rijndael follows the design principles of Square [9]. In its full version, both the block and the key sizes can range from 128 to 256 bits in steps of 32 bits. For AES, the block size of Rijndael is restricted to 128 bits. This paper deals with non-AES Rijndael variants, that is, *large-block Rijndael-b*, with $b > 128$ indicating the block size and key size in bits.

AES is probably the most well-studied block cipher, having received about 15 years of extensive public scrutiny by now. Square attacks, impossible differential attacks, boomerang attacks, rectangle attacks and meet-in-the-middle attacks in both the single-key and related-key settings are just several prominent examples of cryptanalysis techniques applied to AES [1, 4–7, 12, 17, 20, 22–24, 26–28].

The large-block Rijndael is arguably less analyzed, being a highly relevant cipher though. Among others, an important motivation for the study of large-block Rijndael is the deployment of Rijndael-like permutations in the design of hash functions, Whirlwind [2] and SHA-3 finalist Grøstl [16] constituting some especially interesting instances. We mention here several multiset and integral cryptanalytic results [13, 15, 18, 21], as well as impossible differential cryptanalysis [19, 25]. In terms of the impossible differential cryptanalysis – the major object of our study in this paper – the best attack has been proposed by Zhang et al. [25] which cryptanalyzes 9-round Rijndael-224 and Rijndael-256 with 2^{209} and $2^{208.8}$ encryptions, respectively.

Impossible differential cryptanalysis, which was proposed by [3,8], is a widely used cryptanalytic technique. The attack starts with finding a certain input difference that can never result in a certain output difference, which makes up an *impossible differential*. Usually, impossible differentials have truncated input and output differences. By adding rounds before and/or after the impossible differential, one can collect pairs with certain plaintext and ciphertext differences. If there exists a pair that meets the input and output values of the impossible differential under some subkey bits, these bits must be wrong. In this way, we discard as many wrong keys as possible and exhaustively search the rest of the keys. The early abort technique is usually used during the key recovery phase, that is, one does not guess all the subkey bits at once, but guesses some subkey bits instead to discard some pairs that do not satisfy certain conditions step by step. In this case, we can discard the unwished pairs as soon as possible to reduce the time complexity.

Our Contributions. In this paper, we present more powerful 6-round impossible differentials for Rijndael-224 and Rijndael-256. Using these impossible differentials, we can improve the existing impossible differential cryptanalyses of both Rijndael-224 and Rijndael-256. In addition, the improvement can result in a 10-round attack on Rijndael-256.

Our impossible differentials for both Rijndael-224 and Rijndael-256 have more active bytes in the output difference and, therefore, the number of subkey bytes needed to be guessed during the key recovery phase can range with more options, while the probability for a pair of plaintexts to pass the test of sieving wrong pairs is higher compared to [25].

For 9-round Rijndael-256, utilizing the new impossible differential and depending on the number of subkey bytes needed to be guessed in key recovery

Table 1. Summary of Attacks on Rijndael-224 and Rijndael-256

Cipher	Number of Round	Complexity Time (EN)	Complexity Data(CP)	Attack type	Source
Rijndael-224	7	2^{141}	$2^{130.5}$	Multiset	[18]
	7	2^{167}	2^{138}	Imp. Diff.	[19]
	7	$2^{113.4}$	$2^{93.2}$	Imp. Diff.	[25]
	9	$2^{196.5}$	$2^{196.5}$	Integral	[21]
	9	2^{209}	$2^{212.3}$	Imp. Diff.	[25]
	9	$2^{195.2}$	$2^{198.1}$	Imp. Diff.	sect. 4
	9	2^{162}	2^{208}	Imp. Diff.	sect. 4
	9	2^{130}	2^{216}	Imp. Diff.	sect. 4
Rijndael-256	7	$2^{128}-2^{119}$	$2^{128}-2^{119}$	Part. Sum	[13]
	7	2^{141}	$2^{130.5}$	Multiset	[18]
	7	2^{44}	6×2^{32}	Integral	[15]
	7	2^{182}	2^{153}	Imp. Diff.	[19]
	7	$2^{113.2}$	2^{93}	Imp. Diff.	[25]
	8	$2^{128}-2^{119}$	$2^{128}-2^{119}$	Integral	[15]
	9	2^{204}	$2^{128}-2^{119}$	Integral	[15]
	9	$2^{174.5}$	$2^{132.5}$	Integral	[21]
	9	$2^{208.8}$	$2^{244.3}$	Imp. Diff.	[25]
	9	2^{194}	$2^{229.3}$	Imp. Diff.	subsect. 3.2
	9	$2^{159.1}$	$2^{237.3}$	Imp. Diff.	subsect. 3.3
	9	$2^{127.1}$	$2^{245.3}$	Imp. Diff.	subsect. 3.3
	10	$2^{253.9}$	$2^{244.2}$	Imp. Diff.	subsect. 3.4

CP: Chosen Plaintext; EN: Number of round encryptions

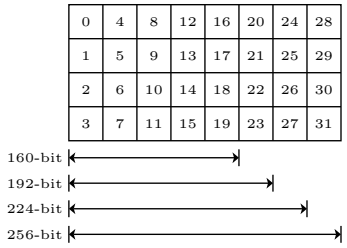
phase, three improved attacks can be obtained. If we guess the same number of subkey bytes as [25], an attack can be mounted with reduced data complexity of $2^{229.3}$ Chosen Ciphertexts (CP), time complexity 2^{194} encryptions and memory complexity $2^{139.6}$ bytes respectively. In addition, if the number of subkey bytes need to guess is less than [25], given $2^{237.3}$ CP, we can attack 9-round Rijndael-256 with $2^{159.1}$ encryptions and $2^{115.3}$ bytes of memory. If the data complexity are increased to $2^{245.3}$ CP, the time and memory complexity can be significantly reduced to $2^{127.1}$ encryptions and $2^{90.9}$ bytes. Moreover, based on the same impossible differential, considering $2^{244.2}$ CP, we can even attack 10-round Rijndael-256 with $2^{253.9}$ encryptions and $2^{186.8}$ bytes of memory accesses. As for Rijndael-224, similarly three attacks can also be mounted on 9-round with lower complexity. With $2^{198.1}$ CP, an attack is demonstrated on 9-round Rijndael-224 with $2^{195.2}$ encryptions and $2^{140.4}$ bytes memory. Take 2^{208} CP, we can attack 9-round Rijndael-224 with 2^{162} encryptions and 2^{117} bytes memory. Increasing the data complexity to 2^{216} chosen plaintexts, the time complexity can be greatly reduced to 2^{130} encryptions and the memory requirements to $2^{93.6}$ bytes.

To the best of our knowledge, these results are the best impossible differential attacks on Rijndael-224 and Rijndael-256. We summarize our results for Rijndael-224 and Rijndael-256, as well as the major previous results in Table 1.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of Rijndael and introduces the notations used in this paper. In Section 3 we first derive a new 6-round impossible differential, and then present three improved impossible differential attacks on 9-round Rijndael-256. The attack can also be extended to 10-round Rijndael-256. Then in Section 4, after a new 6-round impossible differential distinguisher is presented, we mount three improved attacks on 9-round Rijndael-224. Finally, we conclude this paper in Section 5.

2 Description of Rijndael and Notations

Rijndael has N_r rounds, which can be 10, 12, or 14 depending on the key size. In Rijndael, both the text block and the key sizes can range for 128 up to 256 bits in steps of 32 bits. The 128-bit block version of Rijndael, with the key size 128, 192 or 256, is officially known as AES [14]. The plaintext, ciphertext, subkey, and all the intermediate data are represented by a $4 \times N_b$ `state matrix` of bytes, where N_b is the number of 32-bit words in the block. The byte indexing for the `state matrix` is shown in the left part of Figure 1. The key schedule



N_b	C_0	C_1	C_2	C_3
5	0	1	2	3
6	0	1	2	3
7	0	1	2	4
8	0	1	3	4

Fig. 1. Byte Index of the State Matrix and the Shift Offsets for Each Block Length N_b

derives $(N_r + 1)$ b-bit `RoundKey(RK)` from the master key, denoted from RK_0 to RK_{N_r}. The Expanded Key is a linear array of 4-byte words and is denoted by $W[N_b * (N_r + 1)]$. The first N_k words $W[0]\|W[1]\|\cdots\|W[N_k - 1]$ are directly initialised by the N_k words of the master key, while the remaining key words, $W[i]$ for $i \in \{N_k, \cdots, N_k*(N_r+1)-1\}$ are generated by the following algorithm:

`if` $(i \bmod N) = 0$ `then` $W[i] = W[i - N_k] \oplus f(W[i-1]) \oplus Rcon[i/N_k]$
`else if` $((N_k > 6)$ and $(i \bmod N = 4))$ `then` $W[i] = W[i-N_k] \oplus g(W[i-1])$
`else` $W[i] = W[i - N_k] \oplus W[i-1]$

where $f, g : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$ are nonlinear permutations, *Rcon* denotes fixed constants depending on its input. `Roundkey` RK_i is given by the Round Key buffer words $W[N_b * i]$ to $W[N_b * (i + 1)]$.

The round function, which is repeated $(N_r - 1)$ times, involves four operations: `SubBytes(SB)`, `ShiftRows(SR)`, `MixColumns(MC)` and `AddRoundKey(ARK)`.

The `SubBytes` operation consists of the parallel application of a fixed 8-bit to 8-bit Sbox to each byte of the state. `ShiftRows` is a byte transposition that left shifts the rows of the state over different offsets. The shift offsets C_i of row i which depend on the block length N_b, are specified in the right part of Figure 1 for each block length of Rijndael. `MixColumns` is an (4×4) Maximum Distance Separable (MDS) matrix multiplication over $GF(2^8)$ for each column of the state. Obviously the branch number of this MDS matrix is five. `AddRoundKey` consists of the exclusive-or combination of the `RoundKey` with the intermediate state.

These $(N_r - 1)$ rounds are surrounded by an whitening layer consisting of `AddRoundKey` only, and the last round with `MixColumns` operation omitted. We also assume that this is the same case for the reduced Rijndael we are focusing on throughout this paper. Here we only give a brief description of Rijndael, for more detailed specification of the cipher, we refer to [10, 11].

We will also use the technique that the operations of `MixColumns` and `AddRoundKey` can be interchanged under some conditions [11]. Here we introduce some notations as well for later use in the following.

X_i : the state of the i-th round;
ΔX_i : the difference for state of the i-th round;
X_i^I : the input state of the i-th round;
RK_i : the subkey of the i-th round;
RK_i^* : the value of the subkey of the i-th round after the inverse of the `MixColumns` operation;
X_i^{SB} : the intermediate state after the `SubBytes` operation in the i-th round;
X_i^{SR} : the intermediate state after the `ShiftRows` operation in the i-th round;
X_i^{MC} : the intermediate state after the `MixColumns` operation in the i-th round;
X_i^W : the intermediate state after the `AddRoundKey` operation with RK_i^* in the i-th round;
X_i^O : the intermediate state after the `AddRoundKey` operation in the i-th round;
? : an indeterminate difference.

Obviously, $X_i^I = X_{i-1}^O$ hold. Note that the operation of `AddRoundKey` will be represented as ARK^* throughout this paper when the Roundkey RK^* is used.

3 Improved Impossible Differential Attacks on Rijndael-256

In this section, we first give a new 6-round impossible differential for Rijndael-256 in Section 3.1. Based on this impossible differential and depending on the number of the subkey bytes need to guess during the key recovery phase, three improved 9-round impossible differential attacks compared to [25] will be presented in Subsection 3.2 and 3.3 respectively. Using the same impossible differential, we can extend it to an attack of 10-round Rijndael-256 in Subsection 3.4.

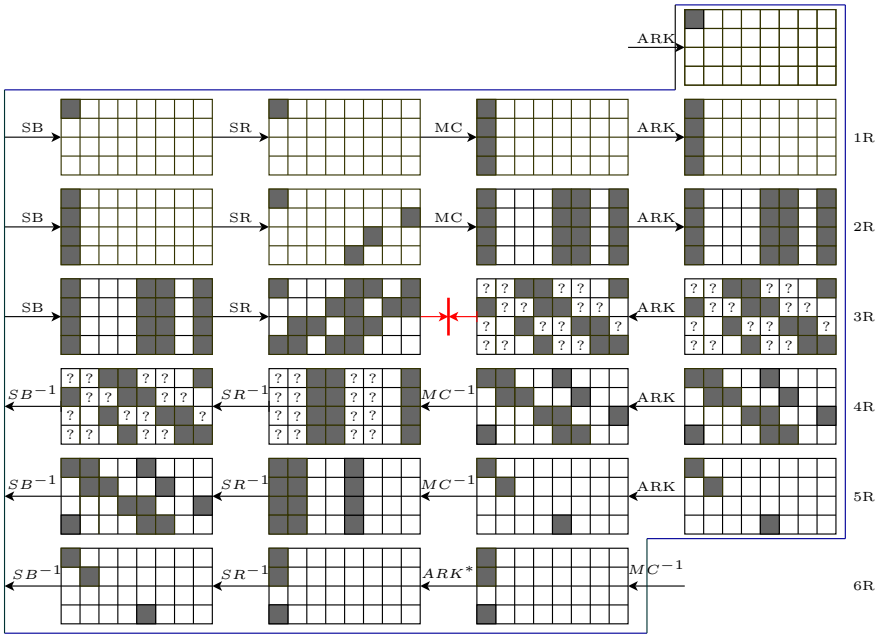


Fig. 2. The New 6-Round Impossible Differential of Rijndael-256

3.1 New 6-Round Impossible Differential on Rijndael-256

Assume we start with round 1 (denoted as 1R in Figure 2) and the input difference ΔX_1 has one active byte whereas the other bytes are zero, one illustration with the first byte active is depicted in Figure 2. Then request 2.5 rounds encryption from the `SB` operation in round 1 to the `SR` operation in round 3 to get the difference ΔX_3^{SR}. Consider the output difference with three nonzero bytes in the first column of the state, one option with the active bytes at (0,1,3) is shown in Figure 2, the other option has the active bytes at (0,2,3). Decrypt 3.5 rounds (as depicted from the operation `ARK`* in round 6 to the operation `ARK` in round 3) in order to get the difference ΔX_3^{MC}. Note there is no `AddRoundKey` operation in round 6 because the order of `MixColumns` and `AddRoundKey` operations can be interchanged as mentioned before in Section 2. For the third column of the state ΔX_3^{SR}, the number of nonzero bytes is one, while it is at most three for the nonzero bytes of ΔX_3^{MC} (it is indeterminate at byte 9). Since the branch number of the MDS matrix is five, there is a contradiction before and after the `MixColumns` operation. By similar reasoning, there is also a contradiction in the seventh column in the state before and after the `MixColumns` operation in round 3. Therefore, we make up a 6-round impossible differential for Rijndael-256.

There exist more active bytes in the output of the impossible differential compared to [25] (they has one byte), therefore we have more options in guessing the subkey bytes to meet the output of the impossible differential while adding extra rounds after the impossible differential distinguisher. There are three active

bytes in one column after MC^{-1} at the bottom of the impossible differential, thus the number of the subkey bytes we need to guess in order to calculate the output after MC^{-1} can range from two to four, therefore three attacks can be mounted using this impossible differential.

3.2 9-Round Attack on Rijndael-256 with Lower Data Complexity

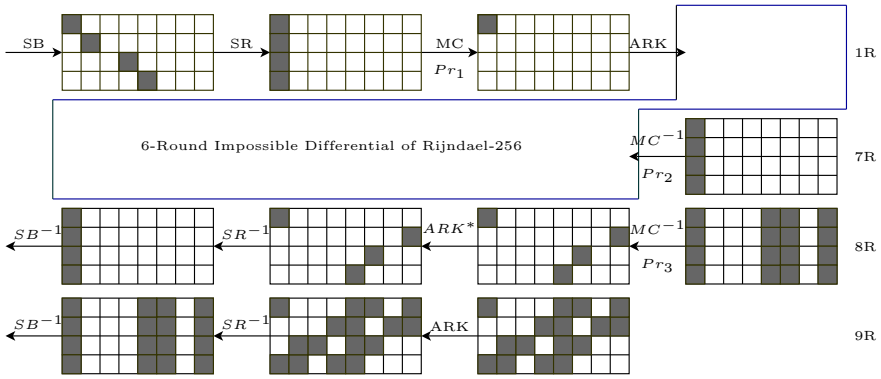


Fig. 3. Improved 9-Round Attack on Rijndael-256 with Lower Data Complexity

In this subsection we present the attacks on 9-round Rijndael-256 utilizing the 6-round impossible differential in Subsection 3.1. In our attack, we guess the same number (i.e. four) of subkey bytes of RK_8^* as [25] in the *key recovery phase*. As a result, 16 bytes of subkey RK_9 will have to be guessed to partially decrypt round 9 in order to calculate X_8^W (as shown in Figure 3), which will also provide a 128-bit condition for ciphertexts in the *data collection phase*. The number of active bytes at the end of the new impossible differential distinguisher will filter out more wrong pairs during the *key recovery phase*. Therefore, an improved attack with significantly reduced data complexity compared to [25] will be result in. The detailed procedures of the attack will be described as follows.

Data Collection. We first choose 2^n structures of plaintexts. In each structure the plaintexts range over all 32-bit values at bytes (0,5,14,19), while the other bytes can take certain fixed values. Each structure includes about $(2^{32})^2/2 = 2^{63}$ pairs of plaintexts, therefore $2^n \cdot 2^{63} = 2^{n+63}$ pairs of plaintexts will be prepared. Encrypt these pairs and keep the one whose ciphertext difference are zero at bytes (1,2,4,5,8,9,11,12,14,21,23,24,26,27,30,31). The probability of such ciphertexts is about $2^{-8\cdot 16} = 2^{-128}$, thus the expected number of the remaining pairs after this phase is about $2^{n+63-128} = 2^{n-65}$.

The sieving of the ciphertexts can be done by birthday attack. As a result, the time complexity of this phase is about 2^{n+32}. In addition, we need $2^{n-65} \cdot 5 \cdot 32 = 2^{n-57.7}$ bytes memory to store these pairs.

Key Recovery. In order to check if the pairs generated in *data collection phase* satisfy the impossible differential in Figure 3, we need to guess certain bytes of subkey $(RK_9, RK_0, , RK_8^*)$ during the *key recovery phase*. The details are described in the following:

Step 1 For all the pairs of plaintext obtained in the data collection phase, we guess the 32-bit subkey $(RK_{9,0}, RK_{9,29}, RK_{9,22}, RK_{10,19})$ and partially decrypt round 9 to compute the first column of ΔX_8^W. Check if the differences at byte (1,2,3) are zero. If it is not the case, discard the pair. The probability of this event is 2^{-24}. After this step the expected number of remaining pairs is about $2^{n-65-24} = 2^{n-89}$.

Step 2 For every guess of the 32-bit subkey $(RK_{9,16}, RK_{9,13}, RK_{9,6}, RK_{9,3})$, we partially decrypt round 9 to compute the fifth column of ΔX_8^W. Check if the differences at byte (0,1,2) are zero. If it is not the case, discard the pair. The probability of this event is 2^{-24}. After this step the expected number of remaining pairs is about $2^{n-89-24} = 2^{n-113}$.

Step 3 For every guess of the 32-bit subkey $(RK_{9,20}, RK_{9,17}, RK_{9,10}, RK_{9,7})$, we partially decrypt round 9 to compute the sixth column of ΔX_8^W. Check if the differences at byte (0,1,3) are zero. If it is not the case, discard the pair. The probability of this event is 2^{-24}. After this step the expected number of remaining pairs is about $2^{n-113-24} = 2^{n-137}$.

Step 4 For every guess of the 32-bit subkey $(RK_{9,28}, RK_{9,25}, RK_{9,18}, RK_{9,15})$, we partially decrypt round 9 to compute the eight column of ΔX_8^W. Check if the differences at byte (0,2,3) are zero. If it is not the case, discard the pair. The probability of this event is 2^{-24}. After this step the expected number of remaining pairs is about $2^{n-137-24} = 2^{n-161}$.

Step 5 We need to guess the 32-bit of subkey $(RK_{0,0}, RK_{0,5}, RK_{0,14}, RK_{0,19})$ for all the remaining pairs, and partially encrypt round 1 to get the first column of ΔX_1^{MC}. Check if the difference at byte (1,2,3) are zero. If it is not the case, discard the pair. The probability of this event is about $4 \cdot (2^8 - 1)/2^{32} \approx 2^{-22}$. Thus after this step the remained pairs is about $2^{n-161-22} = 2^{n-183}$.

Step 6 For every guess of the 16-bit subkey $(RK_{8,0}^*, RK_{8,29}^*, RK_{8,22}^*, RK_{8,19}^*)$, partially decrypt round 8 to compute the first column of ΔX_7^W. Check if the differences at the third byte is zero. If it is correct, delete all the 32-bit subkey guesses of RK_8^* since such a differential is impossible, each subkey guess that proposes such a differential is a wrong key. After analyzing all the 2^{n-183} remaining pairs, if there still remains value of RK_8^*, output the 192-bit subkey guess of (RK_0, RK_8^*, RK_9) as the correct key. Our experiments provide the evidence that the probability of the pairs pass this step is about $Pr_2 = 2 \cdot 2^{-8} = 2^{-7}$.

The process steps of the *key recovery phase* above are described in Table 2, whereas the second column lists the bytes need to be guessed in the correspond-

ing round for each step. The third column stands for the number of remained pairs after sieving in each step, and the time complexity of each step will be measured in the fourth column in Table 2. Note that when evaluating the time complexity of the recovery, it is measured by one round encryption. Similar tables will be adopted to describe the steps of the *key recovery phase* throughout this paper.

Table 2. Key Recovery Processes of the Attack on Rijndael-256 with lower Data Complexity

Step	Guessed Bytes	#Pairs Kept	Time Complexity
1	$RK_9 : 0, 29, 22, 19$	$2^{n-65-24} = 2^{n-89}$	$2^{32} \cdot 2^{n-65} \cdot 2/8 = 2^{n-35}$
2	$RK_9 : 16, 13, 6, 3$	$2^{n-89-24} = 2^{n-113}$	$2^{64} \cdot 2^{n-89} \cdot 2/8 = 2^{n-27}$
3	$RK_9 : 20, 17, 10, 7$	$2^{n-113-24} = 2^{n-137}$	$2^{96} \cdot 2^{n-113} \cdot 2/8 = 2^{n-19}$
4	$RK_9 : 28, 25, 18, 15$	$2^{n-137-24} = 2^{n-161}$	$2^{128} \cdot 2^{n-137} \cdot 2/8 = 2^{n-11}$
5	$RK_0 : 0, 5, 14, 19$	$2^{n-161-22} = 2^{n-183}$	$2^{160} \cdot 2^{n-161} \cdot 2/8 = 2^{n-3}$
6	$RK_8^* : 0, 29, 22, 19$	-	$2^{192} \cdot 2 \cdot [1 + (1 - 2^{-7}) + (1 - 2^{-7})^2 + \cdots + (1 - 2^{-7})^{2^{n-183}}]/8$

Analysis of the Attack Take $n = 197.3$, after analyzing all the remaining pairs, there will be about $2^{192} \cdot (1-2^{-7})^{2^{n-183}} = 2^{-36.2}$ wrong subkeys of RK_0 left, we can get rid of the wrong subkeys by $2^{187.8}$ trail encryptions. Therefore the data complexity will be $2^{n+32} = 2^{229.3}$, the time complexity will be $2^{197.2}/9 \approx 2^{194}$ 9-round encryptions, the memory required is about $2^{139.6}$ bytes.

3.3 9-Round Attack on Rijndael-256 with lower Time Complexity

We will use the same new 6-round impossible differential as the previous section, which helps to get rid of more pairs. As mentioned in Subsection 3.1, the number of subkey bytes need to be guessed in round 8 can be reduced compared to [25], i.e. two or three bytes of RK_8^*. Here we take two bytes for example. As a result, it will be the same case for round 9, which means fewer columns need to be decrypted. Meanwhile, 192 bits are zero for the ciphertexts, which provides an stronger condition of ciphertexts for sieving wrong pairs compared to 128 bits in [25]. As a result an improved attack with the time complexity greatly reduced can be mounted on 9-round Rijndael-256. Because of the similarity of the attack with the one in Subsection 3.2, only a brief description of this attack will be demonstrated as follows:

In the *data collection phase*, we take the same structures as in Subsect 3.2, thus 2^{n+63} pairs of plaintexts will be generated. there exists the 192-bit condition for ciphertext to discard wrong pairs, thus the expected number of the remaining pairs is $2^{n+63-192} = 2^{n-129}$ at the end of this phase.

In the *key recovery phase*, we only guess 8 bytes of RK_9, 4 bytes of RK_0 and 2 bytes of RK_8^* to check if the impossible differential will be satisfied for the

remaining pairs. When filtering out wrong pairs, we obtain the probabilities that the pairs pass the tests in round 8, round 1 and round 7 are $Pr_3 = (2^{-24})^2 = 2^{-48}$, $Pr_1 = 4 \cdot (2^8 - 1)/2^{32} \approx 2^{-22}$ and $Pr_2 = 2 \cdot 2^{-8} \approx 2^{-7}$ respectively. The expected number of remaining pairs after this phase is about 2^{n-199}. The steps and the time complexity evaluation of this phase are given in Table 3. Take

Table 3. Key Recovery Processes of the Improved Attack on Rijndael-256 with lower Time Complexity

Step	Guessed Bytes	#Pairs Kept	Time Complexity
1	$RK_9 : 0, 29, 22, 19$	$2^{n-129-24} = 2^{n-153}$	$2^{32} \cdot 2 \cdot 2^{n-129}/8 = 2^{n-99}$
2	$RK_9 : 28, 25, 18, 15$	$2^{n-153-24} = 2^{n-177}$	$2^{64} \cdot 2 \cdot 2^{n-153}/8 = 2^{n-91}$
3	$RK_0 : 0, 5, 14, 19$	$2^{n-177-22} = 2^{n-199}$	$2^{96} \cdot 2 \cdot 2^{n-177}/8 = 2^{n-83}$
4	$RK_8^* : 0, 29$	-	$2^{112} \cdot 2 \cdot [1 + (1 - 2^{-7}) + (1 - 2^{-7})^2 + \cdots + (1 - 2^{-7})^{2^{n-199}}]/16$

$n = 213.3$, the data complexity is $2^{n+32} = 2^{245.3}$ CP, the time complexity will be $2^{130.3}/9 \approx 2^{127.1}$ 9-round encryptions, the memory required is about $2^{90.9}$ bytes.

Moreover, as mentioned at the beginning of this subsection, it is also possible to guess three bytes of the subkey RK_8^* to calculate ΔX_7^W in order to check if the impossible differential can be satisfied. As a result 12 bytes of RK_9 have to be guessed to partially decrypt round 9 in the key recovery phase. In this case, the data complexity is about $2^{237.3}$ CP, the time complexity is about $2^{159.1}$ 9-round encryption, and the memory is about $2^{115.3}$ bytes.

3.4 10-Round Impossible Differential Attack on Rijndael-256

Based on the same impossible differential as in the previous subsection, we will extend two rounds backwards and forwards respectively, an attack on 10-round Rijndael-256 will be led with complexity less than exhaustive search. We adopt the 9-round attack with lower time complexity in Subsection 3.3 to act as our internal 9-round attack, on which we make some modification. In addition, we will take the key schedule into consideration. The brief attack will be given out as follows.

In the *data collection phase*, take 2^n structures of plaintexts, in which the plaintexts range over 128-bit values at bytes (0,3,4,5,9,12,14,16∼19,21,23,26,30,31), while the other bytes can take certain fixed values. Each structure includes about $(2^{128})^2/2 = 2^{255}$ pairs of plaintexts, therefore $2^n \cdot 2^{255} = 2^{n+255}$ pairs of plaintexts are obtained. Encrypt these pairs and keep the one whose ciphertext difference are zero at bytes (1∼14,16,17,20,21,23,24,26,27,30,31). The probability of such ciphertexts is about $2^{-8 \cdot 24} = 2^{-192}$, thus the expected number of the remaining pairs after this phase is about $2^{n+255-192} = 2^{n+63}$.

In the *key recovery phase*, as in the 9-round attack in Subsection 3.3, 8 subkey bytes of RK_{10} should be guessed. Because of the extra round backward extension,

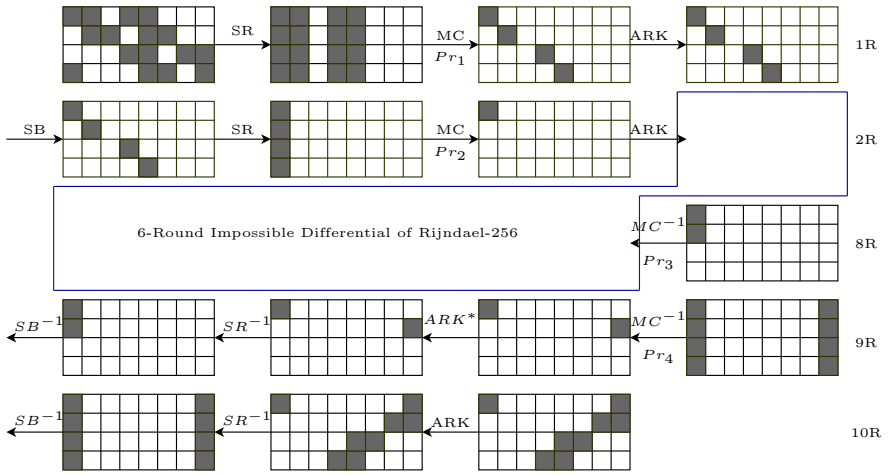


Fig. 4. 10-Round Impossible Differential Attack on Rijndael-256

16 bytes of RK_0 and 4 bytes of RK_1 will also be guessed respectively. At the end of this phase, the number of remaining pairs is 2^{n-103}. The process steps of this phase are described in Table 4. By the key schedule, we can calculate

Table 4. Key Recovery Processes of the Attack on 10-Round Rijndael-256

Step	Guessed Bytes	#Pairs Kept	Time Complexity
1	$RK_{10}: 0, 29, 22, 19$	$2^{n+63-24} = 2^{n+39}$	$2^{32} \cdot 2^{n+63} \cdot 2/8 = 2^{n+93}$
2	$RK_{10}: 28, 25, 18, 15$	$2^{n+39-24} = 2^{n+15}$	$2^{64} \cdot 2^{n+39} \cdot 2/8 = 2^{n+101}$
3	$RK_0: 0, 5, 14, 19$	$2^{n+15-24} = 2^{n-9}$	$2^{96} \cdot 2^{n+15} \cdot 2/8 = 2^{n+109}$
4	$RK_0: 4, 9, 18, 23$	$2^{n-9-24} = 2^{n-33}$	$2^{128} \cdot 2^{n-9} \cdot 2/8 = 2^{n+117}$
5	$RK_0: 12, 17, 26, 31$	$2^{n-33-24} = 2^{n-57}$	$2^{160} \cdot 2^{n-33} \cdot 2/8 = 2^{n+125}$
6	$RK_0: 16, 21, 30, 3$	$2^{n-57-24} = 2^{n-81}$	$2^{192} \cdot 2^{n-57} \cdot 2/8 = 2^{n+133}$
7	$RK_1: 0, 5, 14, 19$	$2^{n-81-22} = 2^{n-103}$	$2^{224} \cdot 2^{n-81} \cdot 2/8 = 2^{n+141}$
8	$RK_9^*: 0, 29$	-	$2^{240} \cdot 2 \cdot [1 + (1-2^{-7}) + (1-2^{-7})^2 + \cdots + (1-2^{-7})^{2^{n-103}}]/16$

$RK_{0,29}$ from $RK_{0,0}$ and $RK_{1,0}$. $RK_{0,5}$ and $RK_{1,5}$ determine $RK_{1,1}$, then $RK_{1,1}$ together with $RK_{0,30}$ determine $RK_{0,1}$. Therefore, in order to recover the key, there are 14 bytes of RK_0 left to guess. We can take $n = 116.2$, from the *data collection phase* we know that the data complexity of the attack is $2^n \cdot 2^{128} = 2^{244.2}$ Chosen Ciphertext (CP). In the *key recovery phase*, after analyzing the remaining $2^{n-103} = 2^{13.2}$ pairs, the expected number of wrong subkeys is $2^{240} \cdot (1-2^{-7})^{2^{n-103}} \approx 2^{133.5}$. With about $2^{112} \cdot 2^{133.5} = 2^{245.5}$ trail encryptions, the correct key will be recovered. The time complexity is about $2^{257.2}/10 \approx 2^{253.9}$ 10-round encryptions. The memory required to store the pairs is about $2^{186.8}$ bytes.

4 Improved Impossible Differential Attacks on Rijndael-224

In this section, we first give a new 6-round impossible differential of Rijndeal-224 (see Figure 5 in Appendix A). Utilizing the new 6-round impossible differential, we extend one round at the top and two rounds at the bottom to mount 9-round impossible differential attacks on Rijndael-224. As we can see there exist three active bytes at the bottom of the distinguisher, as a result the number of the subkey bytes need to guess in round 8 during the key recovery stage can range from two to four. Therefore three 9-round attacks on Rijndael-224 can be obtained respectively. First assume there are four bytes of subkey RK_8^* need to guess in order to check if the impossible differential distinguisher is satisfied during the key recovery phase, as depicted in Figure 6.

In the *data collection phase*, choose structures of 2^{32} plaintexts, in which the plaintexts take all possible 32-bit values at bytes (0,5,10,19) while the others take certain fixed values. Take $2^{166.1}$ structures, about $2^{229.1}$ pairs of plaintexts will be generated. Filter out the pairs whose ciphertext difference are not zero at byte (1∼5,8,10,13,16,19,23,26). Because of this 96-bit condition for ciphertexts, the expected number of remaining pairs is $2^{133.1}$ at the end of this phase.

In the process of *key recovery phase*, we need to guess 16 bytes of subkey RK_9, 4 bytes of subkey RK_8^* and 4 bytes of RK_0 to check if the 6-round of impossible differential is satisfied. While guessing the 4 bytes of RK_8^*, the probability that a pair can pass the test is about $Pr_2 = 2^{-8}$. The rest of the steps are similar to Subsection 3.2. At the end of this phase, there exist about $2^{133.1-96-22} = 2^{15.1}$ pairs.

After analyzing the remaining $2^{15.1}$ pairs, we can get rid of $2^{192} \cdot (1 - 2^{-8})^{2^{15.1}} \approx 2^{-6.3}$ wrong pairs. With about $2^{185.7}$ encryption trails the key can be recovered. The data complexity of this attack is about $2^{198.1}$ CP, the time complexity is about $2^{198.4}/9 \approx 2^{195.2}$ encryptions, and the memory we need for storing pairs is about $2^{133.1} \cdot 5 \cdot 32 = 2^{140.4}$ bytes.

As mentioned above, we can also guess three bytes of subkey RK_8^*, given 2^{208} CP, a 9-round attack can be mounted with the time complexity and memory about 2^{162} encryptions and 2^{117} bytes respectively. Moreover in the case that two bytes of RK_8^* are guessed, the data, time and memory complexity can be 2^{216} CP, 2^{130} encryptions and $2^{93.6}$ bytes respectively.

5 Conclusion

More powerful 6-round impossible differentials for both Rijndael-224 and Rijndael-256 are presented in this paper. Based on those, we significantly improve impossible differential attacks on both Rijndael-224 and Rijndael-256. The improvement can also result in a 10-round attack on Rijndael-256.

Acknowledgments. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

1. Bahrak, B., Aref, M.R.: A Novel Impossible Differential Cryptanalysis of AES. In: proceedings of WEWoRC (2007)
2. Barreto, P.S.L.M., Nikov, V., Nikova, S., Rijmen, V., Tischhauser, E.: Whirlwind: a new cryptographic hash function. Des. Codes Cryptography 56(2-3), 141–162 (2010)
3. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In: Stern, J. (ed.) Advances in Cryptology - EUROCRYPT '99. LNCS, vol. 1592, pp. 12–23. Springer (1999)
4. Biham, E., Dunkelman, O., Keller, N.: Related-Key Impossible Differential Attacks on 8-Round AES-192. In: Pointcheval, D. (ed.) CT-RSA. Lecture Notes in Computer Science, vol. 3860, pp. 21–33. Springer (2006)
5. Biryukov, A.: The Boomerang Attack on 5 and 6-Round Reduced AES. In: Dobbertin, H., Rijmen, V., Sowa, A. (eds.) AES Conference. Lecture Notes in Computer Science, vol. 3373, pp. 11–15. Springer (2004)
6. Biryukov, A., Khovratovich, D.: Related-Key Cryptanalysis of the Full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 5912, pp. 1–18. Springer (2009)
7. Biryukov, A., Khovratovich, D., Nikolic, I.: Distinguisher and Related-Key Attack on the Full AES-256. In: Halevi, S. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 5677, pp. 231–249. Springer (2009)
8. Borst, J., Knudsen, L.R., Rijmen, V.: Two attacks on reduced idea. In: Fumy, W. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 1233, pp. 1–13. Springer (1997)
9. Daemen, J., Knudsen, L.R., Rijmen, V.: The Block Cipher Square. In: Biham, E. (ed.) FSE. Lecture Notes in Computer Science, vol. 1267, pp. 149–165. Springer (1997)
10. Daemen, J., Rijmen, V.: AES Proposal: Rijndael. In: 1st AES Conference, California, USA (1998)
11. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)
12. Demirci, H., Selçuk, A.A.: A Meet-in-the-Middle Attack on 8-Round AES. In: Nyberg, K. (ed.) FSE. Lecture Notes in Computer Science, vol. 5086, pp. 116–126. Springer (2008)
13. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved Cryptanalysis of Rijndael. In: Schneier, B. (ed.) FSE. Lecture Notes in Computer Science, vol. 1978, pp. 213–230. Springer (2000)
14. FIPS 197: Advanced Encryption Standard. Federal Information Processing Standards Publication 197, U.S. Department of Commerce/N.I.S.T (2001)
15. Galice, S., Minier, M.: Improving Integral Attacks Against Rijndael-256 Up to 9 Rounds. In: Vaudenay, S. (ed.) AFRICACRYPT. Lecture Notes in Computer Science, vol. 5023, pp. 1–15. Springer (2008)
16. Gauravaram, P., Knudsen, L.R., Matusiewicz, K., Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: Grøstl - a SHA-3 candidate. Submission to NIST (2008), `http://www.groestl.info`
17. Gilbert, H., Minier, M.: A Collision Attack on 7 Rounds of Rijndael. In: AES Candidate Conference. pp. 230–241 (2000)
18. Jr., J.N., de Freitas, D.S., Phan, R.C.W.: New Multiset Attacks on Rijndael with Large Blocks. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt. Lecture Notes in Computer Science, vol. 3715, pp. 277–295. Springer (2005)

19. Jr., J.N., Pavão, I.C.: Impossible-Differential Attacks on Large-Block Rijndael. In: Garay, J.A., Lenstra, A.K., Mambo, M., Peralta, R. (eds.) ISC. Lecture Notes in Computer Science, vol. 4779, pp. 104–117. Springer (2007)
20. Kim, J., Hong, S., Preneel, B.: Related-Key Rectangle Attacks on Reduced AES-192 and AES-256. In: Biryukov, A. (ed.) FSE. Lecture Notes in Computer Science, vol. 4593, pp. 225–241. Springer (2007)
21. Li, Y., Wu, W.: Improved Integral Attacks on Rijndael. Journal of Information Science and Engineering 27(6), 2031–2045 (2011)
22. Lu, J., Dunkelman, O., Keller, N., Kim, J.: New Impossible Differential Attacks on AES. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT. Lecture Notes in Computer Science, vol. 5365, pp. 279–293. Springer (2008)
23. Lucks, S.: Attacking Seven Rounds of Rijndael under 192-bit and 256-bit Keys. In: AES Candidate Conference. pp. 215–229 (2000)
24. Phan, R.C.W.: Impossible differential cryptanalysis of 7-round Advanced Encryption Standard (AES). Inf. Process. Lett. 91(1), 33–38 (2004)
25. Zhang, L., Wu, W., Park, J.H., Koo, B., Yeom, Y.: Improved Impossible Differential Attacks on Large-Block Rijndael. In: Wu, T.C., Lei, C.L., Rijmen, V., Lee, D.T. (eds.) ISC. Lecture Notes in Computer Science, vol. 5222, pp. 298–315. Springer (2008)
26. Zhang, W., Wu, W., Feng, D.: New Results on Impossible Differential Cryptanalysis of Reduced AES. In: Nam, K.H., Rhee, G. (eds.) ICISC. Lecture Notes in Computer Science, vol. 4817, pp. 239–250. Springer (2007)
27. Zhang, W., Wu, W., Zhang, L., Feng, D.: Improved Related-Key Impossible Differential Attacks on Reduced-Round AES-192. In: Biham, E., Youssef, A.M. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 4356, pp. 15–27. Springer (2006)
28. Zhang, W., Zhang, L., Wu, W., Feng, D.: Related-Key Differential-Linear Attacks on Reduced AES-192. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT. Lecture Notes in Computer Science, vol. 4859, pp. 73–85. Springer (2007)

A New 6-Round Impossible Differential of Rijndael-224 and 9-Round Attack with lower Data Complexity

Assume we start with round 1 and there is only one nonzero byte of the input difference ΔX_1 whereas the other bytes are zero. One options is depicted in Figure 5 with nonzero byte at the first byte position. Then encrypt the input for 2.5 rounds from the `SB` operation in 1R to the `SR` operation in 3R to get the difference ΔX_3^{SR}. Given the output difference with three nonzero bytes in the first column, whereas the other bytes are zero. For Rijndael-224, the only option exists is given in Figure 5. After 3.5 rounds decryption (as depicted from the operation `ARK`* in round 6 to the operation `ARK` in round 3 in order to get the difference ΔX_3^{MC}). For the first column of the state X_3^{SR}, the number of nonzero bytes of ΔX_3^{SR} is one, while the maximum number of nonzero bytes of ΔX_3^{MC} is three. Since the branch number of the MDS matrix is five, there exists an contradiction. Therefore, we make up a 6-round impossible differential for Rijndael-224.

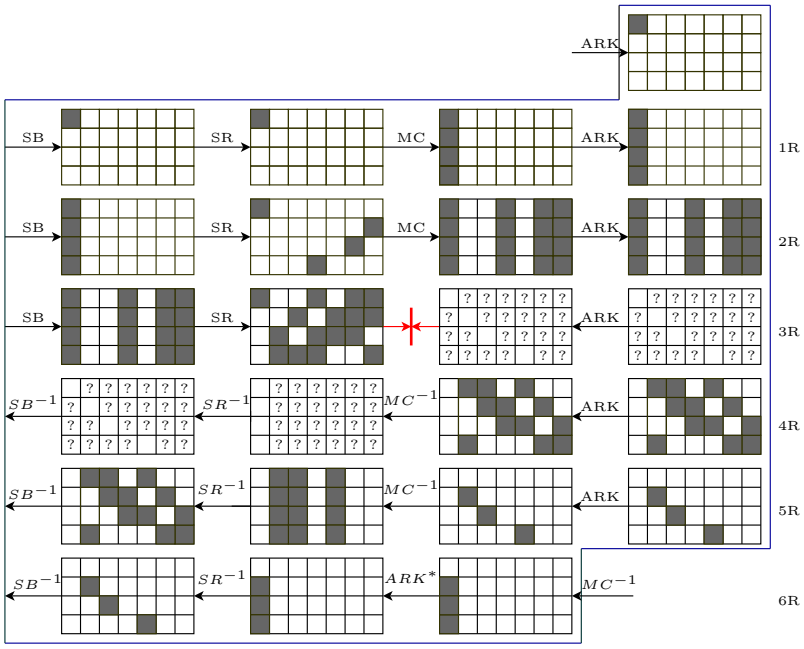


Fig. 5. The New 6-Round Impossible Differential of Rijndael-224

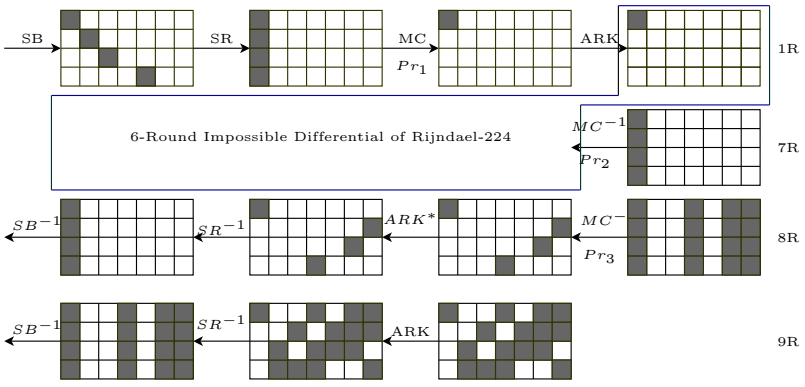


Fig. 6. The Improved 9-Round Attack on Rijndael-224 with lower Data Complexity