Improved Single-Key Attacks on 9-Round AES-192/256

Leibo Li[1], Keting Jia[2] and Xiaoyun Wang[1,3⋆]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, School of Mathematics, Shandong University, Jinan 250100, China
lileibo@mail.sdu.edu.cn

[2] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

[3] Institute for Advanced Study, Tsinghua University, Beijing 100084, China
ktjia,xiaoyunwang@mail.tsinghua.edu.cn

Abstract. This paper focuses on key-recovery attacks on 9-round AES-192 and AES-256 under single-key model with the framework of the meet-in-the-middle attack. A new technique named *key-dependent sieve* is introduced to further reduce the size of lookup table of the attack, and the 9-round AES-192 is broken with 2^{121} chosen plaintexts, $2^{187.5}$ 9-round encryptions and 2^{185} 128-bit words of memory. If the attack starts from the third round, the complexities would be further reduced by a factor of 16. Moreover, the whole attack is split up into a series of weak-key attacks. Then the memory complexity of the attack is saved significantly when we execute these weak attacks in streaming mode. This method is also applied to reduce the memory complexity of the attack on 9-round AES-256.

Keywords: AES, Block Cipher, Meet-in-the-Middle Attack, Differential Characteristic.

1 Introduction

The block cipher Rijndael was designed by Daemen and Rijmen in 1997, and was selected as the Advanced Encryption Standard (AES) in 2001 by NIST. It is a Substitution-Permutation Network (SPN) with variable key length of 128, 192, 256, which are denoted as AES-128, AES-192 and AES-256, respectively.

For the reason of its importance and popularity, the security of AES has attracted a great amount of attention from worldwide cryptology researchers. Many methods of cryptanalysis were applied to attack AES in previous years, such as impossible differential attack [15,19,16], SQUARE attack [5], collision attack [13], meet-in-the-middle attack [6,11,7,18,8,9], biclique attack [4], related-key attack and chosen-key distinguishing [1,2,3,12]. Although the attacks under

⋆ Corresponding author.

related-key model, could be used to break the full versions of AES-192 and AES-256 based on exploiting the key schedule [1,2,3], but such attacks require a very powerful assumption that the adversary can ask to modify the unknown key used in the encryption. So related-key attacks are widely used as an important method to estimate the security of a block cipher, but are not regarded as a real threat to the application of a cipher in practice. For the attacks under single-key model, up to now, the best attacks except the biclique method could reach to 7-round for AES-128, 8-round for AES-192 and 9-round for AES-256. The biclique method was used to attack the full AES with a marginal complexity over exhaustive search by Bogdanov, Khovratovich and Rechberger at ASIACRYPT 2012 [4].

In this paper, we focus on the meet-in-the-the middle attack (MITM) in the single-key model, which was deeply researched in recent years, and now may be the most efficient attack on all versions of AES [9]. The meet-in-the-middle attack was first proposed by Diffie and Hellman to attack DES [10]. For AES cipher, this method was introduced by Demirci and Selçuk at FSE 2008 [6] to improve the collision attack proposed by Gilbert and Minier [13]. They constructed a 4-round distinguisher to attack the 7-round and 8-round AES. The attack needs a small data complexity of 2^{34}, but requires a large memory of $2^{25\times 8}$ to set up precomputation table determined by 25 intermediated variable bytes. The number of parameters could be reduced to 24 bytes, if one considers to store the differentials instead of values in precomputation table. Combined with data/time/memory tradeoff, this attack was applied to analyse 7-round AES-192 and 8-round AES-256.

At ASIACRYPT 2010, Dunkelman, Keller and Shamir [11] exploited the differential enumeration and multiset ideas for MITM attacks to reduce the high memory complexity in the precomputation phase. Indeed, they showed that if a pair conforms to a truncated differential characteristic, the number of the desired 24 intermediated variable bytes will be reduced to 16. Since this attack reduces the memory complexity with the expense of increasing the data complexity to make a pair conform to the differential characteristic, it may be seen as a new data/time/memory tradeoff. Furthermore, Derbez, Fouque and Jean presented a significant improvement of Dunkelman *et al.*'s attack at EUROCRYPT 2013 [9]. Using the rebound-like idea, they showed that many values in precomputation table are not reached at all under the constraint of the truncated differential. Actually, the size of precomputation table is determined by 10-byte parameters. Based on the 4-round distinguisher, they gave the most efficient attacks on 7-round AES-128 and 8-round AES-192/256. Besides, they introduced a 5-round distinguisher to analyse 9-round AES-256.

Our contribution. Although the MITM attack has been improved and perfected a lot by Dunkelman et al. and Derbez et al. [11,9], we notice that some key relations could be further exploited to improve the results of previous at-

tacks. In this paper, based on the properties of the key schedule, we construct a stronger 5-round distinguisher, which supports us to give more efficient attacks on 9-round AES-192/256.

In [9], Derbez *et al.* proposed a 5-round distinguisher of AES with the memory complexity of 2^{208}, so it seems infeasible for the attack on AES-192. However, by studying the key relationship of the distinguisher, we find that many values in the previous precomputation table could be filtered, and the size of the table is only 2^{192}. Especially, if the attack starts from the third round, the size of precomputation table would be further reduced by a factor of 2^8. Subsequently, combing with the classic data/time/memory trade-off, we present an attack on 9-round AES-192 with about 2^{121} chosen plaintexts, $2^{187.5}$ encryptions and 2^{185} 128-bit storages. For the attack on AES-192 starting from the third round, the data, time and memory complexities are reduced to 2^{117}, $2^{183.5}$ and 2^{181}, respectively. Since the new technique takes advantage of the subkeys involved in distinguisher as the filter conditions to reduce the size of precomputation table, we call it *key-dependent sieve*.

In the second part of the paper, we show that the whole attack is able to be sorted into a series of weak-key attacks by using of the shared key information in the online and offline phases, where every weak-key attack takes an independent sub-table included in the precomputation table. That supports us to reduce the memory complexity of the attack without any cost of the data and time complexities, since we can perform the attack in streaming mode by working on each weak attack independently and releasing the memories afterwards. For 9-round attacks on AES-192 and AES-256, the memory complexities are reduced by 2^8 and 2^{32} times, respectively. Although the data and time complexities are not reduced in such case, it is meaningful for us to save the memory requirement of the attack, specially, for the attack that the memory complexity takes over the dominant term.

Table 1 summaries our results along with some major previous results of AES-192 and AES-256 under single-key model. The rest of this paper is organized as follows. Section 2 gives a brief description of AES and some related works. In section 3, we propose the single-key attacks on 9-round AES-192. Section 4 presents an interesting method to reduce the memory complexity of the attack. Finally, we conclude the paper in section 5.

2 Preliminaries

This section first gives a brief description of AES and denotes some notations and definitions used throughout the paper. Finally, we introduce some related works of AES with meet-in-the-middle attack.

Table 1. Summary of the Attacks on AES-192/256 in the Single-key Model

Cipher	Rounds	Attack Type	Data	Time	Memory	Source
AES-192	8	MITM	2^{113}	2^{172}	2^{129}	[11]
	8	MITM	2^{113}	2^{172}	2^{82}	[9]
	8	MITM	2^{113}	2^{140}	2^{130}	[8]
	9	Bicliques	2^{80}	$2^{188.8}$	2^{8}	[4]
	9	MITM	2^{121}	$2^{187.5}$	2^{185}	Section 3.2
	9	MITM	2^{121}	$2^{186.5}$	$2^{177.5}$	Section 4.1
	9 (3-11)	MITM	2^{117}	$2^{183.5}$	2^{181}	Section 3.3
	9 (3-11)	MITM	2^{117}	$2^{182.5}$	$2^{165.5}$	Section 4.1
	Full	Bicliques	2^{80}	$2^{189.4}$	2^{8}	[4]
AES-256	8	MITM	2^{113}	2^{196}	2^{129}	[11]
	8	MITM	2^{113}	2^{196}	2^{82}	[9]
	8	MITM	$2^{102.83}$	2^{156}	$2^{140.17}$	[8]
	9	Bicliques	2^{120}	$2^{251.9}$	2^{8}	[4]
	9	MITM	2^{120}	2^{203}	2^{203}	[9]
	9	MITM	2^{121}	$2^{203.5}$	$2^{169.9}$	Section 4.2
	Full	Bicliques	2^{40}	$2^{254.4}$	2^{8}	[4]

2.1 A Brief Description of AES

The advanced encryption standard (AES) [17] is a 128-bit block cipher, which uses variable key sizes and the number of rounds (N_r) depend on the key sizes, i.e., 10 rounds for 128-bit key size, 12 rounds for 192-bit key size and 14 rounds for 256-bit key size. The 128-bit internal state is treated as a byte matrix of size 4×4, and each byte represents a value in $GF(2^8)$. The round function is composed of four basic operations:

- SubBytes (SB) is a nonlinear byte-wise substitution that applies an 8 by 8 S-box to every byte.
- ShiftRows (SR) is a linear operation that rotates on the left of the $i-$th row by i bytes.
- MixColumns (MC) is a matrix multiplication over a finite field applied to each column.
- AddRoundKey (ARK) is an exclusive-or operation with the round subkey.

Before the first round an additional whitening ARK operation is performed, and in the last round the MC operation is omitted.

The key schedule of AES expands the master key to $N_r + 1$ 128-bit subkeys. The subkey array is denoted by $w[0, \cdots, 4 \times N_r + 3]$, where each word $w[\cdot]$ consists 32 bits. Let the number of words for master key is denoted by N_k, e.g., $N_k = 6$ for AES-192. Then the first N_k words of $w[\cdot]$ are filled with the master key. The remaining words are defined as follows:

– For $i = N_k$ to $4 \times N_r + 3$ do the following:
  - If $i \equiv 0 \bmod N_k$, then $w[i] = w[i-N_k] \oplus SB(w[i-1] \lll 8) \oplus Rcon[i/N_k]$,
  - else if $N_k = 8$ and $i \equiv 4 \bmod 8$, then $w[i] = w[i - N_k] \oplus SB(w[i-1])$,
  - Otherwise $w[i] = w[i - N_k] \oplus w[i-1]$.

where $\lll$ represents left rotation, $\oplus$ denotes the bit-wise exclusive OR (XOR) and $Rcon[\cdot]$ is an array of fixed constants. For more details about AES, we refer to [17].

2.2 Notations and Definitions

In this paper, the plaintext and ciphertext are denoted by P and C. The symbols X_i, Y_i, Z_i and W_i denote the internal states before SB, SR, MC and ARK operations in the round-i ($0 \le i \le N_r - 1$), respectively. The subkey of round i is denoted by k_i, the first key (whitening) is denoted by k_{-1}. We use the symbol u_i to represent the equivalent key with $u_i = MC^{-1}(k_i)$.

A 128-bit internal state A is represented as a 4×4 byte matrix. The symbol $A[i]$ is used to express a byte of A, where i is the ordering of bytes ($i = 0, \cdots, 15$). The symbol $A[i, \cdots, j]$ represents the $i-$th byte to the $j-$th byte of A.

As in previous works, the $\delta-$set utilized in this paper is defined as follows.

Definition 1. ($\delta-$set, [5]) *The $\delta-$set is a set of 2^8 AES states that one byte traverses all values (the active byte) and the other bytes are constants (the inactive bytes).*

We denote the $\delta-$set as $(X^0, \cdots, X^{255})$. Usually, we consider to encrypt a $\delta-$set by a function E_K and select the $i-$th byte of the ciphertexts as the output value ($0 \le i \le 15$), then the corresponding 2^8 output bytes form a 2048-bit vector $E_K(X^0)[i] \| \cdots \| E_K(X^{255})[i]$ with ordered arrangement, where $\|$ represents the bit string concatenation. Another important concept is the multiset, which was introduced by Dunkelman *et al.* in [11].

Definition 2. (Multiset of bytes [11]) *A multiset generalizes the set concept by allowing elements to appear more than once. Here, a multiset of 256 bytes can take as many as $\binom{511}{255} \approx 2^{506.7}$ different values.*

Property 1. (Differential property of $S-$box [11]) Given the input and output differences of the SubBytes operation, there exists a pair of actual values on average to satisfy these differences. This property is also applied to the inversion of SubBytes operation.

The time complexity of the attack in this paper is measured with the unit of an equivalent encryption operation of the 9-round AES. The memory complexity is measured with the unit of a block size (128-bit). It is emphasized that we count all operations performed during the attack, in particular, the time and memory requirements in precomputation phase.

2.3 Related Works

In this section, we recall the previously MITM attacks on AES. Firstly, we introduce the Demirci and Selçuk attack. Then two improvements given by Dunkelman *et al.* and Derbez *et al.* are shown briefly.

Demirci and Selçuk attack. Combining the MITM method, Demirci and Selçuk improved the collision attack [13] on AES. They treated the cipher E as $E_K = E^2_{K_2} \circ E^m \circ E^1_{K_1}$, and built a distinguisher in E^m based on the following 4-round AES property.

Property 2. Consider the encryption of a $\delta-$set through four full AES rounds. For each of the 16 bytes of the state, the ordered sequence of 256 values of that byte in the corresponding ciphertexts is fully determined by just 25 byte parameters. Consequently, for any fixed byte position, there are at most 2^{200} possible sequences when we consider all the possible choices of keys and δ-sets (out of 2^{2048} theoretically value).

These parameters are composed of some intermediate states of a message of the $\delta-$set, which are determined by the positions of the active byte and the corresponding output byte. If the active byte is located in $X_1[0]$ and the output byte is selected in $X_5[0]$, then, as described in Fig. 1, the 25-byte parameter is $X_2[0,1,2,3]\|X_3[0,\cdots,15]\|X_4[0,5,10,15]\|X_5[0]$. When the values in the output sequence were substituted by the corresponding differences, the value $X_5[0]$ could be omitted and the number of parameter reduces to 24.

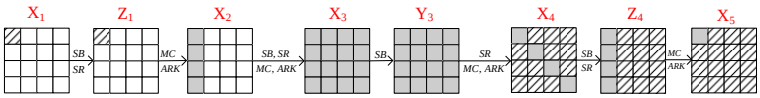

Fig. 1. The 4-round AES distinguisher used in [13], the gray cells represent 25-byte parameter

On the basic of Property 2, they gave the MITM attacks on 7-round and 8-round AES-256, respectively. For the attack on 7-round AES, one round and 2 rounds are extended in the top and bottom of the 4-round E_m, respectively. The attack is divided into two phases, precomputation phase and online phase.

1. Precomputation phase: compute all 2^{200} values of the sequence given in Property 2, and store them in a hash table.
2. Online phase:

(a) Encrypt a structure of 2^{32} chosen plaintexts such that the main diagonal can take all the 2^{32} possible values and the remaining bytes are constant.
(b) Guess values of the related subkeys in E_1, and construct a δ-set. Then partially decrypt to get the corresponding 256 plaintexts.
(c) Obtain the corresponding plaintext-ciphertext pairs from the collection data. Then guess the related subkeys in E_2, and partially decrypt the ciphertexts to get the corresponding 256-byte value of the output sequence of E_m.
(d) If a sequence value lies in the precomputation table, the guessed related subkeys in E_1 and E_2 may be right key.
(e) Exhaustive search the remaining subkeys to obtain the right key.

The data complexity of the attack in the online phase is only about 2^{32}, but the memory and time complexities for precomputation phase are too large. So the data/time/memory tradeoff was used in their work to reduce the complexities in the precomputation phase, which makes the attack to apply to 7-round AES-192. However, the time complexity in the online phase is very large, then it is impossible to rebalance for 8-round AES-192 in their work.

Dunkelman *et al.*'s Attack. At ASIACRYPT 2010, Dunkelman, Keller and Shamir [11] proposed some interesting techniques to improve the Demirci and Selçuk attack. Firstly, they proposed to use the multiset to replace the ordered sequence for the output byte, since it is enough to distinguish a proper value from the random sequences. Secondly, a novel idea named differential enumeration technique was introduced to reduce the memory complexity in the precomputation phase, at the expense of increasing the data complexity. The main idea of this technique is to fix some values of intermediate parameters by using of the truncated differential. They showed that if one considers to encrypt a $\delta-$set after four full-rounds of AES, in the case of that a message of the $\delta-$set belongs to a pair conforming the particular 4-round truncated differential characteristic described as in Fig. 2, then the corresponding output value of multiset only takes about 2^{128} possible values. Note that the gray cells in Fig. 2 are active bytes, while the white cells are inactive. Indeed, it is obvious that when a pair conforms the truncated differential as in Fig. 2, the state X_3 only takes about 2^{64} different values.

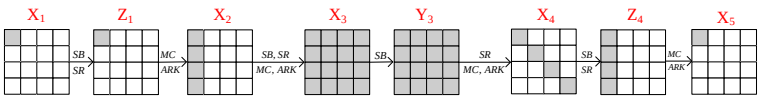

Fig. 2. The truncated differential characteristic of 4-round AES used in [11]

Hence, there are about 2^{128} possible values stored in the precomputation phase. In the online phase, more plaintexts should be chosen to make sure there exists a pair in content with the truncated differential. Thus, the data complexity is 2^{113} chosen plaintexts. This attack procedure is similar to Demirci and Selçuk attack, but a step to look for a pair satisfying the truncated differential is added, and the $\delta-$set is constructed only for such pair. Finally, they gave attacks on the 7-round AES-128 and 8-round AES-192/256. Actually, the attack can be regarded as a special data/time/memory tradeoff.

Derbez *et al.*'s Attack. More recently, Derbez, Fouque and Jean presented a significant improvement to Dunkelman *et al.*'s attack at EUROCRYPT 2013 [9]. Combining with the rebound-like view of the cipher, they showed that the number of possible values of precomputation table in Dunkelman *et al.*'s attack could be further reduced. In their work, if a message of $\delta-$set belongs to a pair conforming the 4-round truncated differential characteristic outlined in Fig. 2, the value of multiset is only determined by 10-byte variables of intermediate state $\Delta Z_1[0]\|X_2[0,1,2,3]\|\Delta X_5[0]\|Z_4[0,1,2,3]$. In other words, there are only about 2^{80} possible sequences of multiset to be stored in precomputation table. Then they improved the attacks on 7-round AES-128 and 8-round AES-192/256. Further, they proposed to use the truncated differential characteristic which contains two active byte in X_1, to balance the time and memory complexities of the attack.

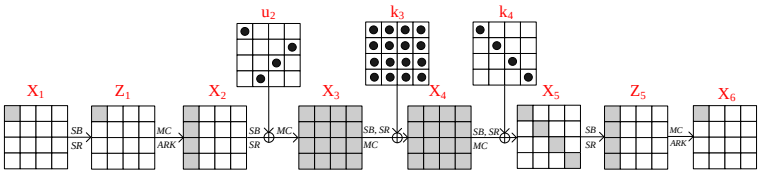

Fig. 3. The truncated differential characteristic of 5-round AES used in [9]

Moreover, they proposed to use a 5-round distinguisher to attack 9-round AES-256. The corresponding truncated differential characteristic is outlined in Fig 3, where the value of multiset is determined by 26-byte parameters

$$\Delta Z_1[0]\|X_2[0,1,2,3]\|X_3[0,\cdots,15]\|\Delta X_6[0]\|Z_5[0,1,2,3].$$

It is interesting that a value of the 192-bit subkey is known for each value of multiset in precomputation table. For the above differential characteristic, the 192-bit deduced subkey is $u_2[0,7,10,13]\|k_3[0,\cdots,15]\|k_4[0,5,10,15]$.

3 The Improved Attacks on 9-Round AES-192

In this section, we apply the improved 5-round distinguisher to attack 9-round AES-192. It turns out that, the size of hash table in precomputation phase is able to be reduced to 2^{192} from 2^{208}. More importantly, if the 5-round distinguisher starts from the fourth round, the size of hash table would be reduced to 2^{184}.

3.1 Key-Dependent Sieve and 5-Round Distinguisher of AES-192

It is obvious that the memory complexity and time complexity in the precomputation phase are the bottlenecks of the MITM attack on AES. Nevertheless, we find that some key relations are valuable to reduce the complexity in the precomputation phase, where the same key information is deduced by two approaches, the parameters and the key schedule. Then both of them may be not equal since two approaches are absolutely independent. This makes us to filter the redundant values of precomputation table and makes it possible to attack on 9-round AES-192.

We review the 5-round distinguisher proposed by Derbez et al. in [9]. Since the size of lookup table is determined by 26 parameters, it seems infeasible to attack 9-round AES-192. However, by the key schedule of AES-192, it is obviously that the knowledge of k_3 allows to deduce the column 0 and 1 of k_2. That means the value of the equivalent subkey $u_2[0,7]$ is computed by k_3. However, $u_2[0,7]$ is already deduced by the 26-byte parameter for each value of the multiset seen Fig. 3. Thus there exists a contradiction between $u_2[0,7]$ and k_3. In other words, if a possible value of the multiset is correct, the value of $u_2[0,7]$ must be equal to the equivalent value deduced from k_3, which happens with a probability of 2^{-16}. Therefore, the size of look up table is about $\frac{2^{208}}{2^{16}}$ for 5-round distinguisher of AES-192. Because our technique filtering the wrong states is based on the key relationship, so we call it *key-dependent sieve*. Combined with data/time/memory tradeoff, we apply the 5-round distinguisher to attack 9-round AES-192. However, the time complexity of precomputation phase is too large for all possible values of lookup table, that is about $2^{192} \times 2^8$ computations. So we introduce an improved 5-round distinguisher of AES-192 in the sequel.

The 5-round differential characteristic utilized in our attack is described in Fig. 4, where the position of the active byte is defined in $W_0[12]$, and the output value of sequence is located in $Y_6[6]$.

Proposition 1. *Consider the encryption of the first* 2^5 *values* $(W_0^0, \cdots, W_0^{31})$ *of the* $\delta-$*set through 5-round AES-192, in the case of that a message pair* (W_0, W_0') *of the* $\delta-$*set conforms to the truncated differential characteristic outlined in Fig. 4, then the corresponding 256-bit ordered sequence* $Y_6^0[6] \| \cdots \| Y_6^{31}[6]$ *only takes about* 2^{192} *values (out of* 2^{256} *theoretically value).*

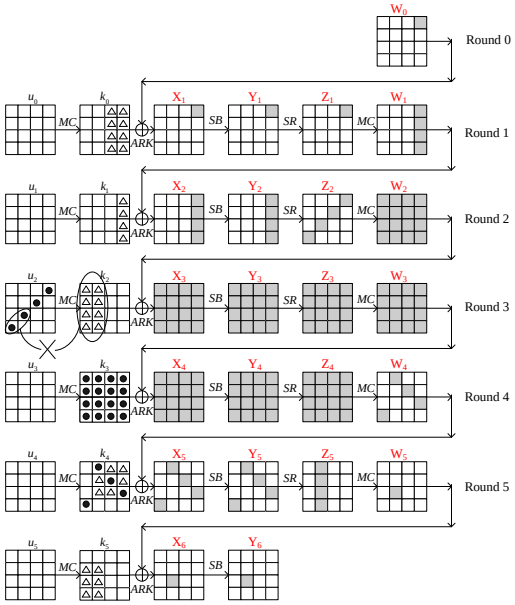

Fig. 4. The truncated differential characteristic of the 5-round AES-192

Proof. We give a brief proof of this proposition. For the encryptions of the first 2^5 values of the $\delta-$set, it is easier to conclude that the output sequence $Y_6^0\|\cdots\|Y_6^{31}$ is determined by the 43-byte variable

$$W_0[12]\|X_1[12]\|X_2[12,\cdots,15]\|X_3[0,\cdots,15]\|k_3[0,\cdots,15]\|k_4[3,4,9,14]\|k_5[6].$$

However, if a pair conforms the truncated differential characteristic outlined in Fig 4, the 40-byte value

$$X_2[12,\cdots,15]\|X_3[0,\cdots,15]\|k_3[0,\cdots,15]\|k_4[3,4,9,14]$$

is determined by the 26-byte variable

$$\Delta Z_1[12]\|X_2[12,13,14,15]\|X_3[0,\cdots,15]\|Z_5[4,5,6,7]\|\Delta X_6[6].$$

Here, the knowledge of $\Delta Z_1[12]\|X_2[12,13,14,15]\|X_3[0,\cdots,15]$ supports to compute the intermediate difference ΔX_4. For the backward direction, the knowledge of $Z_5[4,5,6,7]\|\Delta X_6[6]$ supports to compute ΔY_4. Then according to Property 1, we get one value of intermediate state $X_4\|Y_4$ on average for the fixed difference $\Delta X_4\|\Delta Y_4$. Apparently, the 192-bit value of subkey $u_2[3,6,9,12]\|k_3[0,\cdots,15]\|$ $k_4[3,4,9,14]$ is also deduced for every 26-byte variable. According to key schedule of AES-192, we can compute the other value $u_2[3,6]$ form k_3. By the key-dependent sieve, there are 2^{192} possible values for 26-byte parameters.

By the key schedule, the value of subkey denoted by triangles in Fig. 4 are deduced by the value of the 192-bit subkey denoted by blackspot in Fig. 4. Here, we only focus on the subkey $k_0[12]\|k_1[12,13,14,15]\|k_5[6]$. For any 26-byte parameter, $k_1[12,13,14,15]$ is used to compute $X_1[12]$, and $k_0[12]$ is used to compute $W_0[12]$. Besides, the values of $X_6[6]$ and $Y_6[6]$ are computed by the value of $k_5[6]$. Therefore, the whole 43-byte variable is deduced by 26-byte parameter. So there are 2^{192} possible values of 43-byte variables, which means the number of possible sequences $Y_6^0[6]\|\cdots\|Y_6^{31}[6]$ is approximately 2^{192}. □

Note that in Derbez *et al.*'s attack, the multiset technique was used to omit the influence of 16-bit subkey belongs to k_0 and k_5. If the truncated differential characteristic is selected as in Fig. 4, the 16-bit subkey would be $k_0[12]\|k_5[6]$. Here, we prove that such 16-bit information could be deduced by 192-bit subkey $u_2[3,6,9,12]\|k_3[0,\cdots,15]\|k_4[3,4,9,14]$. Then we extend the distinguisher to the W_0 in the forward, and Y_6 in the backward. Thus, we use an ordered sequence instead of the multiset. For the output value of encrypting the $\delta-$set, the ordered sequence includes 2048-bit information, while a multiset only contains about 507-bit information. Indeed, only the first 32-byte value of the $\delta-$set is enough to distinguish a proper sequence with the probability of $\frac{2^{192}}{2^{256}} = 2^{-64}$, and the data and time complexities are reduced by 2^3 times in the attack.

3.2 The Key Recovery Attack on 9-Round AES-192

We propose an attack on 9-round AES-192 by adding one round on the top and three rounds on the bottom of the 5-round distinguisher (Fig. 5). The attack is composed of two phases: precomputation phase and online phase. In the precomputation phase, we get all possible 256-bit sequences described as Proposition 1 by using the rebound-like technique, which is described as follows.

Precomputation phase. For each 128-bit k_3, do the following steps.

1. Compute the subkey $u_2[3,6]\|k_1[12,13,14,15]\|k_0[12]$ by the key schedule.
2. Traverse $\Delta X_6[2]\|Z_5[4,5,6,7]$ to compute $\Delta X_5[3,4,9,14]\|X_5[3,4,9,14]$, and store $X_5[3,4,9,14]$ in a table T_0 indexed by $\Delta X_5[3,4,9,14]$. There are about 2^8 values of $X_5[3,4,9,14]$ for each index.
3. For all 64-bit value of difference $\Delta Y_2[12,\cdots,15]\|\Delta X_5[3,4,9,14]$, we apply the super-sbox technique [14] to connect the differences $\Delta Y_2[12,\cdots,15]$ and $\Delta X_5[3,4,9,14]$, and deduce the intermediate value $X_3\|W_4$. Then $Y_2[14,15]$ is obtained by X_3 and $u_2[3,6]$. Store these values with the index of 48-bit value $\Delta Y_2[12,\cdots,15]\|Y_2[14,15]$ in a table T_1. There are about 2^{16} values of $\Delta X_5[3,4,9,14]\|X_3\|W_4[3,4,9,14]$ corresponding to the index $\Delta Y_2[12,\cdots,15]\|Y_2[14,15]$.
4. For each $\Delta Z_1[12]\|X_2[12,13,14,15]$, execute the following substeps.
   (a) Compute the state $X_1[12]\|W_0[12]\|\Delta Y_2[12,13,14,15]\|Y_2[12,13,14,15]$.

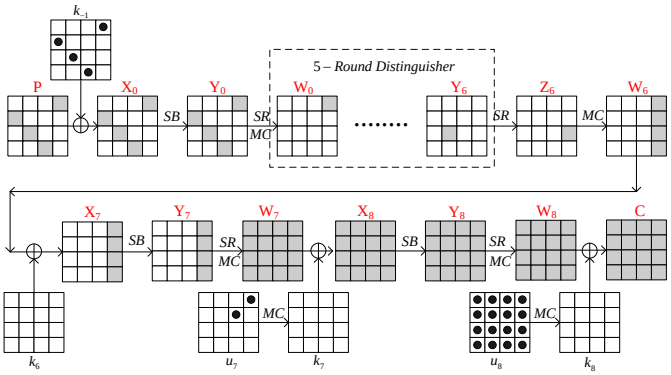

Fig. 5. The attack on 9-round AES-192

(b) Then look up the table T_1 to get about 2^{16} values $\Delta X_5[3,4,9,14] \| X_3 \| W_4[3,4,9,14]$ by the values of $\Delta Y_2[12,13,14,15] \| Y_2[14,15]$. And get the equivalent subkey $u_2[9,12]$.

(c) For each value of $\Delta X_5[3,4,9,14] \| X_3 \| W_4[3,4,9,14]$, we get 2^8 values of $X_5[3,4,9,14]$ by accessing the table T_0. Then compute $k_4[3,4,9,14]$ and $k_5[6]$. Here we get 43-byte variable $W_0[12] \| X_1[12] \| X_2[12,\cdots,15] \| X_3[0,\cdots,15] \| k_3[0,\cdots,15] \| k_4[3,4,9,14] \| k_5[6]$.

(d) Construct the $\delta-$set, and compute the corresponding sequence $Y_6^0[6] \| \cdots \| Y_6^{31}[6]$, and store them in a hash table $\mathcal{H}$.

Online phase. In the online phase, we need to find at least a pair satisfying the truncated differential characteristic, then construct the $\delta-$set, and obtain the first 32 bytes output value of it. Finally, detect whether it belongs to the precomputation table. The attack procedure is described as follows.

1. Encrypt 2^{81} structures of 2^{32} plaintexts, such that $P[1,6,11,12]$ takes all 32-bit values and other bytes are constants. There are 2^{144} pairs totally.
2. For each pair, do the following substeps.
   (a) Guess the difference value $\Delta Y_7[12,13,14,15]$, and compute the subkey u_8 (or k_8, if the last MC operation is omitted). Then deduce $u_7[3,6]$.
   (b) Compute the difference $\Delta X_7[14,15]$, delete the wrong guesses which don't lead to $\Delta Z_6[12,13,15] = 0$, there are about 2^{24} guesses remaining after this step.
   (c) For each remaining guess, deduce subkey $u_7[9,12]$.
   (d) Guess the difference $\Delta W_0[12]$, and compute the subkey $k_{-1}[1,6,11,12]$.
3. For each deduced subkey, select one message of the pair and get the value $W_0[12]$. Then change the value of $W_0[12]$ to be $(0,\cdots,31)$ and compute plaintexts $(P^0,\cdots,P^{31})$. Query their corresponding ciphertexts, and get the

corresponding sequence $Y_6^0[6]\|\cdots\|Y_6^{31}[6]$ by partial decryption. Note that the equivalent subkey $u_6[14]$ is deduced by u_8 in such case.

4. Find the right subkeys by verifying whether the sequence lies in table $\mathcal{H}$. There are about $2^{176}\times 2^{-64}$ subkeys remaining in the end. Then exhaustively search for $u_7[8,10,11,13,14,15]$ to find the real key, which needs about 2^{160} encryptions.

Complexity analysis. In the precomputation phase, each value of the $\delta-$set needs about 2-round AES computations. Then the time complexity of the precomputation phase is about $2^{192}\times 2^5\times 2^{-2.2}=2^{194.8}$ 9-round AES encryptions, which also needs about 2^{193} 128-bit words of memory to store all possible sequences. The time complexity of the online phase is dominated by step 3, where the computation of each value in the $\delta-$set needs about 1.5-round AES encryptions. So the time complexity of the online phase is equivalent to $2^{144}\times 2^{32}\times 2^5\times 2^{-2.6}=2^{178.4}$ 9-round encryptions. The attack needs about 2^{113} chosen plaintexts.

Data/time/memory tradeoff. With data/time/memory tradeoff, the adversary only needs to precompute a fraction 2^{-8} of possible sequences, then the time complexity is about $2^{184}\times 2^5\times 2^{-2.2}=2^{186.8}$ 9-round computations. The memory complexity reduces to $2^{193\times 2^{-8}}=2^{185}$. But in the online phase, the adversary will repeat the attack 2^8 times to offset the probability of the failure, that means the attack becomes probabilistic. So the data complexity increases to 2^{121} chosen plaintexts, and the time complexity increase to $2^{178.4}\times 2^8=2^{186.4}$ 9-round encryptions. In total, including the precomputation phase, time complexity is approximately $2^{187.5}$.

3.3 The Attack on 9-round AES-192 from the Third Round

We observe that the memory complexity will be reduced again when the 5-round distinguisher is mounted to rounds 4-9 in order to attack the reduced-round AES-192 from rounds 3 to 11. The same truncated differential characteristic outlined in Fig. 4 is used in the attack, except to move all intermediate states after two rounds. Then the 5-round distinguisher is from the state W_2 to the state Y_8. Similar to the Proposition 1, we consider to encrypt a $\delta-$set $(W_2^0,\cdots,W_2^{255})$ after 5-round AES-192. If a message pair of the δ-set satisfies the expected truncated differential characteristic, then there are about 2^{192} possible values of sequence $Y_8^0\|\cdots\|Y_8^{255}$. Corresponding, the 176-bit subkey $u_4[9,12]\|k_5[0,\cdots,15]\|k_6[3,4,9,14]$ is deduced for each sequence. Here, $u_4[3,6]$ are omitted, which can be deduced from k_5.

However, as described in Fig. 6, we find that $k_6[4]$ are deduced by the values of $k_5[1]$ and $k_6[9]$, which may be contradicted with the known value $k_6[4]$ for each sequence, where the right half in Fig. 6 is the original key schedule of AES-192,

and the left half is its equivalent value $v_i = MC^{-1}(w_i)$. Note that there exist the following relations for the equivalent key v_i if $i \geq 6$.

- If $i \equiv 0 \bmod 6$, then $v[i] = v[i-6] \oplus MC^{-1}(SB(MC(w[i-1]) \lll 8)) \oplus MC^{-1}(Rcon[i/6])$,
- Else $v[i] = v[i-6] \oplus v[i-1]$.

So there are only about 2^{184} possible sequences remaining after eliminating the incorrect states in such case.

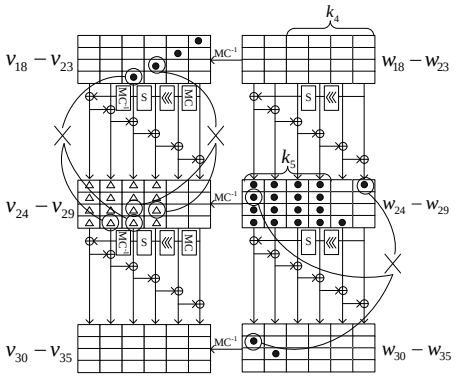

Fig. 6. The key relationship of the attack on AES-192 starting from the third round

The attack procedure is similar to the attack of subsection 3.2. The precomputation table is constructed as follows. For each 128-bit difference ΔX_6, do the following steps.

1. Traverse the 40-bit difference $\Delta Y_3[12] \| \Delta Y_4[12,13,14,15]$ to deduce the states $X_4[12,13,14,15] \| X_5 \| W_5 \| u_4[3,6,9,12]$. Store these states in a hash table T_1 with the 24-bit index

$$(Z_5[7] \oplus Z_5[11] \oplus u_4[3]) \| (Z_5[10] \oplus Z_5[14] \oplus u_4[6]) \| W_5[1].$$

2. Traverse the 40-bit difference $\Delta X_8[6] \| \Delta X_7[3,4,9,14]$ to deduce the intermediate states $X_7[3,4,9,14] \| X_6 \| W_6 \| k_6[3,4,9,14]$. Compute the 24-bit value $(MC(X_6)[7] \oplus MC(X_6)[11]) \| (MC(X_6)[10] \oplus MC(X_6)[14]) \| (X_6[1] \oplus k_6[9] \oplus S(k_6[4]) \oplus Rcon[4][1])$. Then access the hash table T_1 with 24-bit value to get the states $X_4[12,13,14,15] \| X_5 \| W_5 \| u_4[3,6,9,12]$. There are about 2^{16} states in table T_1 for each index. In total, we collect 2^{56} correct states which

satisfy

$$\begin{cases} u_4[3] = u_5[7] \oplus u_5[11], \\ u_4[6] = u_5[10] \oplus u_5[14], \\ k_5[1] = k_6[9] \oplus S(k_6[4]) \oplus Rcon[4][1]. \end{cases}$$

3. Construct the $\delta-$set, compute the corresponding sequence $Y_8^0[6] \| \cdots \| Y_8^{31}[6]$, and store them in a hash table.

Complexity analysis. The time complexity of this phase is equivalent to $2^{184} \times 2^5 \times 2^{-2.2} = 2^{186.8}$ 9-round encryptions, the memory complexity is about $2^{184} \times 2 = 2^{185}$ 128-bit storages. The online phase is exactly the same procedure of the attack in Section 3.2, which needs about 2^{113} chosen plaintexts and $2^{178.4}$ 9-round encryptions.

Data/time/memory tradeoff. We precompute a fraction 2^{-4} possible sequences, then the time complexity of the attack in the online phase is about $2^{178.4} \times 2^4 = 2^{182.4}$ 9-round encryptions. The memory complexity decreases to $2^{185} \times 2^{-4} = 2^{181}$ 128-bit, and the data complexity increases to 2^{117} chosen plaintexts. In additional, we need $2^{182.8}$ 9-round encryptions to compute all possible sequences in precomputation phase, then the time complexity including the precomputation is about $2^{183.5}$ 9-round encryptions.

4 Reducing the Memory Complexity with Weak-Key Attacks

It is known that there exists a subkey k' for every sequence in precomputation table. In other view, such a value k' could be regarded as an extensional characteristic of the sequence. In Section 3, we use the property of self-contradictory phenomenon of the k' to reduce the number of possible sequences. In this section, by investigating more properties of this information, we show that the memory complexity could be further reduced without increasing the data and time complexities.

We denote the subkey guessed in the online phase as $\widehat{k}$. It is obvious there exist some linear relations in k' and $\widehat{k}$. Assuming m bits value $\widetilde{k} \subset (k' \cap \widehat{k})$, we first split the precomputation table with the index of $\widetilde{k}$ into 2^m sub-tables. Thus, in the online phase, for each guessed subkey $\widehat{k}$ and its sequence, instead of checking all precomputation table, we only need to detect a sub-table in the line with the index value $\widetilde{k}$. Furthermore, we also split the sequences computed in the online phase to 2^m subsets with the same index $\widetilde{k}$. Then for all sequences belong to a subset, we only need to detect a sub-table, and it is meaningless to check whether they belong to other sub-tables.

Thus, the whole attack could be sorted into 2^m sub-attacks. Each sub-attack contains a sub-table of precomputation, and all of these attacks are independent

each other. Since each sub-attack is worked under a fixed value of m-bit key information, which is also seen as a weak-key attack. Assuming $\mathcal{C}$ is the time (or memory) complexity of the whole attack, then it is evident to see that the time (or memory) complexity for every weak-key attack is $\mathcal{C}/2^m$, but the data and time complexities of the whole attack don't change at all. Nevertheless, if all weak-key attacks are worked in the streaming model, the memory complexity could be reduced by 2^m times since the storages could be reused for each weak-key attack. However, the whole precomputation table could not be reused in such case.

4.1 Reducing the Memory Complexity for Attacks on 9-round AES-192

We take into account the application of this method to the attack on 9-round AES-192, where k' is 176-bit subkey $u_2[9,12]\|k_3[0,\cdots,15]\|k_4[3,4,9,14]$, and $\widehat{k}$ is the 176-bit subkey $u_8[0,\cdots,15]\|u_7[9,12]\|k_{-1}[1,6,11,12]$. It is not difficult to see that the set $k' \cup \widehat{k}$ contains the whole information of 192-bit master key, and the set $k' \cap \widehat{k}$ contains 160-bit information. We use 8-bit information $k_{-1}[6]$ as the index to split the attack to 2^8 weak-key attacks, where

$$k_{-1}[6] = SB(k_3[1] \oplus k_3[5]) \oplus k_3[10] \oplus k_3[14] \oplus Rcon[2][2].$$

The attack procedure is very simple. We first split 2^{128} possible value of k_3 to 2^8 subsets with the index value $k_{-1}[6]$. Then for each weak-key attack with a fixed value $k_{-1}[6]$, do as follows.

1. For the corresponding subset of k_3, do as described in section 3.2 to construct the sub-table $\mathcal{H}'$.
2. For 2^{113} plaintexts, guess 24-bit subkey $k_{-1}[1,11,12]$, and collect all pairs satisfying $\Delta W_0[13,14,15] = 0$.
3. For every pair guess the difference $\Delta Y_6[6]$.
4. Guess $\Delta Y_7[14,15]$ to deduce the subkey $u_7[3,6]\|u_8[0,1,4,7,10,11,13,14]$, and only keep the value which satisfies $u_7[3] = u_8[7] \oplus u_8[11]$ and $u_7[6] = u_8[10] \oplus u_8[14]$. There is one value of $\Delta Y_7[14,15]$ along with its subkey remain on average for every $\Delta Y_6[6]$.
5. Guess $\Delta Y_7[12,13]$ to deduce the subkey $u_7[9,12]\|u_8[2,3,5,6,8,9,12,15]$.
6. Construct the $\delta-$set, compute the corresponding sequence, and check whether they belongs to $\mathcal{H}'$.

Complexity analysis. For each weak-key attack, the time complexity of step 1 is about $2^{184} \times 2^5 \times 2^{-2.2} = 2^{186.8}$, the memory complexity is about 2^{185} 128-bit. For step 2 to step 6, the time complexity is about $2^{168} \times 2^5 \times 2^{-2.6} = 2^{170.4}$. By data/time/memory tradeoff, we precompute a fraction 2^{-8} possible sequences,

the time complexity is about $2^{179.5}$ 9-round encryptions, the memory complexity could be reduced to 2^{177} 128-bit spaces. In total, the complexity is still $2^{179.5} \times 2^8$ 9-round encryptions, approximately.

Reducing the time complexity by a half. By investigating the information of $k' \cap \widehat{k}$, we learn that the 64-bit information is identified out of 160-bit information, that is $k_{-1}[6] \| k_{-1}[11] \| u_1[12] \| u_3[1] \| k_4[4,5,6] \| k_5[11]$. Then each sequence of the distinguisher is represented by the first 16 bytes of the $\delta-$set along with above 64-bit information and 176-bit k'. Hence, for each sequence computed in the online phase, we get k' by the 192-bit index $Y_6^0[6] \| \cdots \| Y_6^{15}[6] \| k_{-1}[6,11] \| u_1[12] \| u_3[1] \| k_4[4,5,6] \| k_5[11]$, and sieve the right key by verifying the consistency of k' and $\widehat{k}$. The probability of this filter is about $\frac{2^{64}}{2^{160}} = 2^{-96}$. Thus, the time complexity of the attack is reduced by a half, but the memory complexity is increased to $2^{192} \times 2^{1.5} = 2^{193.5}$, which is used to store 368-bit information $Y_6^0[6] \| \cdots \| Y_6^{15}[6] \| k_{-1}[6,11] \| u_1[12] \| u_3[1] \| k_4[4,5,6] \| k_5[11] \| k'$ in such case. Combined with data/time/memory tradeoff and weak-key method, the time complexity of the attack is about $2^{186.5}$, the memory complexity is about $2^{177.5}$.

The attack starting from the third round. For the attack starting from the third round, the 16-bit shared information $k_1[6,11]$ could be used as the index to convert the attack to 2^{16} weak-key attacks, where $k_1[6] = k_5[2] \oplus k_5[6] \oplus k_5[14]$ and $k_1[11] = k_5[7] \oplus k_5[11] \oplus k_6[3]$. The attack procedure is similar to above attack, in use of the data/time/memory tradeoff, the memory complexity of the attack is reduced to 2^{165} 128-bit spaces. If we use the information $k' \cap \widehat{k}$ instead of partial value of the sequence, the time complexity would be reduced to $2^{182.5}$, and the memory complexity is about $2^{165.5}$.

4.2 Reducing the Memory Complexity for the Attack on AES-256

This attack is based on the 5-round distinguisher, where the active byte of the $\delta-$set is defined in $W_0[3]$, and the output value is located in $Y_6[7]$.

Proposition 2. *If one encrypts the first 32 values* $(W_0^0, \cdots, W_0^{31})$ *of the* $\delta-$*set through 5-round AES-256, assuming a pair of the* $\delta-$*set satisfying the expected truncated differential characteristic, then the sequence* $Y_6^0[6] \| \cdots \| Y_6^{31}[6]$ *along with a 192-bit subkey* $u_2[1,4,11,14] \| k_3[0, \cdots, 15] \| k_4[3,4,9,14]$ *takes about* 2^{208} *values.*

According to the generic attack, we need to precompute all possible values of sequence $Y_6^0[6] \| \cdots \| Y_6^{31}[6]$. Then in the online phase, we collect 2^{144} pairs, and find a pair satisfying the differential path for each 192-bit subkey $k_{-1}[0,5,10,15] \| k_8 \| u_7[2,5,8,15]$. After that, construct the $\delta-$set, compute the sequence $Y_6^0[6] \| \cdots \| Y_6^{31}[6]$, and check whether it belongs to precomputation table. Finally, detect the consistency of $u_2[1,4,11,14] \| k_3[0, \cdots, 15] \| k_4[3,4,9,14]$ and $k_{-1}[0,5,10,15] \| k_8 \|$

$u_7[2,5,8,15]$ to retrieve the correct key. Then the time complexity in precomputation phase is about $2^{208} \times 2^5 \times 2^{-2.2} = 2^{210.8}$. The memory complexity is about $2^{209.9}$ 128-bit words of memory, where we need to store 448-bit information. In online phase, the time complexity is about $2^{192} \times 2^5 \times 2^{-2.6} = 2^{194.4}$. The data complexity is about 2^{113} chosen plaintexts. By data/time/memory tradeoff, we precompute a fraction 2^{-8} possible sequences, then the data, time and memory complexities are 2^{121}, $2^{203.5}$ and $2^{201.9}$, respectively. Here, we consider to use the 32-bit information $k_{-1}[10,15] \| k_4[9,14]$ to convert the attack to 2^{32} weak-key attacks, then the memory complexity reduces to $2^{169.9}$. Note that the subkey $k_{-1}[10,15]$ and $k_4[9,14]$ are linear dependent on k_3 and k_8, respectively.

5 Conclusion

In this paper, we take advantage of some subkey relations in the truncated differential to reduce the memory complexity of meet-in-the-middle attack, which is the bottleneck of this kind of attack. For 9-round AES-192, the 16-bit subkey conditions are obtained in the construction of the δ-set sequence. Based on this, we propose the 9-round attack on AES-192. In particular, when the 9-round attack starts from the third round of AES, the time complexity is $2^{182.4}$ 9-round encryption, the data complexity is 2^{117} chosen plaintexts, and the memory complexity is 2^{181} blocks. Moreover, combining the key relations between the inline phase and online phase, we introduce an interesting method to decompose the whole attack into a series of weak-key attacks, which helps to reduce the memory complexity of the attack without increasing the data and time complexities. To the best of our knowledge, these attacks are the most efficient results in single-key model for 9-round AES-192 and AES-256.

Acknowledgments.

We would like to thank anonymous reviewers for their very helpful comments on the paper. This work is supported by 973 Program (No.2013CB834205), and the National Natural Science Foundation of China (No. 61133013, 61373142 and 61272035).

References

1. Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, and Adi Shamir. Key Recovery Attacks of Practical Complexity on AES-256 Variants with up to 10 Rounds. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 299–319. Springer, 2010.

2. Alex Biryukov and Dmitry Khovratovich. Related-Key Cryptanalysis of the Full AES-192 and AES-256. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.
3. Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and Related-Key Attack on the Full AES-256. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 231–249. Springer, 2009.
4. Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique Cryptanalysis of the Full AES. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2011.
5. Joan Daemen and Vincent Rijmen. Aes proposal: Rijndael. In *First Advanced Encryption Standard (AES) Conference*, 1998.
6. Hüseyin Demirci and Ali Aydin Selçuk. A Meet-in-the-Middle Attack on 8-Round AES. In Kaisa Nyberg, editor, *FSE 2008*, volume 5086 of *Lecture Notes in Computer Science*, pages 116–126. Springer, 2008.
7. Hüseyin Demirci, Ihsan Taskin, Mustafa Çoban, and Adnan Baysal. Improved Meet-in-the-Middle Attacks on AES. In Bimal K. Roy and Nicolas Sendrier, editors, *INDOCRYPT 2009*, volume 5922 of *Lecture Notes in Computer Science*, pages 144–156. Springer, 2009.
8. Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean. Exahusting Demirici-Selçuk Meet-in-the-Middle Attacks against Reduced-Round AES. In *FSE 2013 (to appear)*, 2013.
9. Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean. Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 371–387. Springer, 2013.
10. Whitfield Diffie and Martin E. Hellman. Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10:74–84, 1977.
11. Orr Dunkelman, Nathan Keller, and Adi Shamir. Improved Single-Key Attacks on 8-Round AES-192 and AES-256. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 158–176. Springer, 2010.
12. Pierre-Alain Fouque, Jérémy Jean, and Thomas Peyrin. Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128. In Ran Canetti and Juan A. Garay, editors, *CRYPTO*, volume 8042 of *Lecture Notes in Computer Science*, pages 183–203. Springer, 2013.
13. Henri Gilbert and Marine Minier. A Collision Attack on 7 Rounds of Rijndael. In *AES Candidate Conference*, pages 230–241, 2000.
14. Henri Gilbert and Thomas Peyrin. Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations. In Seokhie Hong and Tetsu Iwata, editors, *Fast Software Encryption 2010*, volume 6147 of *Lecture Notes in Computer Science*, pages 365–383. Springer, 2010.
15. Jiqiang Lu, Orr Dunkelman, Nathan Keller, and Jongsung Kim. New Impossible Differential Attacks on AES. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *Progress in Cryptology - INDOCRYPT 2008*, volume 5365 of *Lecture Notes in Computer Science*, pages 279–293. Springer, 2008.

16. Hamid Mala, Mohammad Dakhilalian, Vincent Rijmen, and Mahmoud Modarres-Hashemi. Improved impossible differential cryptanalysis of 7-round AES-128. In Guang Gong and Kishan Chand Gupta, editors, *INDOCRYPT 2010*, pages 282–291. Springer, 2010.
17. National Institute of Standards and Technology. ADVANCED ENCRYPTION STANDARD. In *FIPS PUB 197, Federal Information Processing Standards Publication*, 2001.
18. Yongzhuang Wei, Jiqiang Lu, and Yupu Hu. Meet-in-the-Middle Attack on 8 Rounds of the AES Block Cipher under 192 Key Bits. In Feng Bao and Jian Weng, editors, *ISPEC 2011*, volume 6672 of *Lecture Notes in Computer Science*, pages 222–232. Springer, 2011.
19. Wentao Zhang, Wenling Wu, and Dengguo Feng. New Results on Impossible Differential Cryptanalysis of Reduced AES. In Kil-Hyun Nam and Gwangsoo Rhee, editors, *ICISC 2007*, volume 4817 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2007.