MixColumns Coefficient Property and Security of the AES with A Secret S-Box

Xin An[1,2], Kai Hu[1,2], and Meiqin Wang[1,2(✉)]

[1] School of Cyber Science and Technology, Shandong University, Qingdao 266237, Shandong, China
{anxin19,hukai}@mail.sdu.edu.cn, mqwang@sdu.edu.cn
[2] Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education, Shandong University, Qingdao 266237, Shandong, China

Abstract. The MixColumns operation is an important component providing diffusion for the AES. The branch number of it ensures that any continuous four rounds of the AES have at least 25 active S-Boxes, which makes the AES secure against the differential and linear cryptanalysis. However, the choices of the coefficients of the MixColumns matrix may undermine the AES security against some novel-type attacks. A particular property of the AES MixColumns matrix coefficient has been noticed in recent papers that *each row or column of the matrix has elements that sum to zero*. Several attacks have been developed taking advantage of the coefficient property.

In this paper we investigate further the influence of the specific coefficient property on the AES security. Our target, which is also one of the targets of the previous works, is a 5-round AES variant with a secret S-Box. We will show how we take advantage of the coefficient property to extract the secret key directly without any assistance of the S-Box information. Compared with the previous similar attacks, the present attacks here are the best in terms of the complexity under the chosen-plaintext scenario.

Keywords: AES · MixColumns · Exchange attack · Key recovery attack · Secret S-Box

1 Introduction

The Advanced Encryption Standard (AES) [7] is designed to achieve good resistance against the differential [3] and linear cryptanalysis [13]. This includes the selection of the S-Box and linear components such as the MixColumns matrix. For the AES, the branch number of its MixColumns matrix is chosen as five then it ensures that any four continuous rounds of differential (linear) characteristics have at least 25 active S-Boxes [7,8]. Considering that the maximum correlation and the maximum difference propagation probability over the AES

A. Nitaj and A. Youssef (Eds.): AFRICACRYPT 2020, LNCS 12174, pp. 114–131, 2020.
https://doi.org/10.1007/978-3-030-51938-4_6

S-Box are 2^{-3} and 2^{-6}, respectively, there are no effective differential or linear characteristics for four or more rounds of the AES.

For the performance reasons, the coefficients of the AES MixColumns are chosen from a group of low-weight numbers. Therefore it is not surprising that there are elements in each row or column that will add up to zero. For example, its first row is $[02, 03, 01, 01]$ thus $01 \oplus 01 = 0$ and $01 \oplus 02 \oplus 03 = 0$. Several attacks have been developed facilitated by this property and show that the property can be a potential weakness [2,9,10,12,15]. For convenience, we conclude it into two types concretely as follows as did in [12],

Property 1. Each row or column of the MixColumns matrix has two elements that sum to zero.

Property 2. Each row or column of the MixColumns matrix has three elements that sum to zero.

At Crypto 2016, Sun et al. noticed Property 1 for the first time and established the first zero-correlation linear hull and the first integral distinguisher for the 5-round AES [15]. The two attacks exploited the existing 4-round corresponding properties and extended them one more round based on the MixColumns coefficient property. We take the 5-round zero-correlation linear hull as an example. As is well-known, the previous zero-correlation linear hull can cover at most 3.5 rounds of the AES (without last MixColumns) [4] which is illustrated in Fig. 1[1].

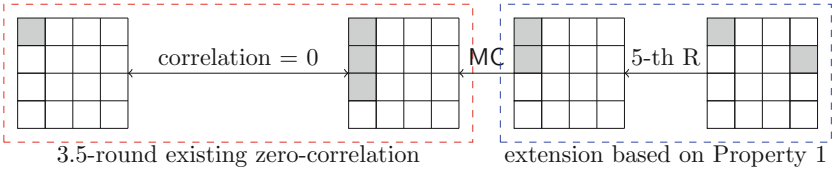

Fig. 1. Extending 3.5-round zero-correlation linear hull for AES to 5 rounds exploiting Property 1

Let the first column of the input mask and the output mask of the MixColumns after the 3.5-round zero-correlation linear hull be Γ_{in} and Γ_{out}, respectively. According to the propagation of the mask over a linear map [4], we have $\Gamma_{in} = M_{AES}^T \Gamma_{out}$, where M_{AES}^T is the transpose of the matrix used by the AES MixColumns. Then if we can ensure that the two active masks of Γ_{out} are equal, we can make certain that Γ_{in} has only three active bytes like Fig. 1. Finally, the zero-correlation linear can be extended to 5 rounds.

Although the two distinguishers in [15] cost the whole codebook, they spawned a sequence of new fundamental results that are based on Property 1 or 2.

[1] In [4], the output mask of the 3.5-round zero-correlation linear hull has only one active byte, but it is easy to check that with 3 active byte in the output mask it is still a zero-correlation linear hull.

Soon after, two following improvements were proposed which aimed to reduce the complexities [6,12]. At FSE 2017, Grassi et al. took Property 1 proposing the first impossible differential distinguisher for the 5-round AES [10]. Later at CT-RSA 2018, the impossible differential distinguisher was further improved by Grassi exploiting Property 2 [9]. In the same paper, he also discussed the attacks on an AES variant with a secret S-Box. By combining the MixColumns coefficient property and the multiple-of-n attack [11], Grassi managed to extract the secret key from the 5-round AES without knowing any information of the S-Box or recovering it in advance as it was done in [16].

The security of the AES variant with a secret S-Box was firstly studied by Tiessen et al. at FSE 2015 [16]. Assuming that the choice of the S-Box is made uniformly at random from all 8-bit S-Boxes and keeping all other components unchanged, the size of the secret information increases from 128 bits to 1812 bits[2] (we focus on the AES-128). Generally speaking, a key-recovery attack requires the details of the S-Box since we have to peel off some key-involved components. Consequently, the authors of [16] needed to recover an equivalent S-Box by the square attack [16] and then found the equivalent secret key. However, the works in [9] showed that it is possible to recover the key information directly without recovering the S-Box in advance if we take advantage of Property 1 or 2 appropriately. At Africacrypt 2019, Bardeh and Rønjom further studied the influence of Property 1 under the adaptive-chosen-ciphertext scenario, which is the newest result in this direction. The AES variant with a secret S-Box has been a popular target for studying the MixColumns coefficient property. In this paper, we also study how to take the MixColumns coefficient property to extract the key information without any knowledge of the S-Box.

1.1 Our Contribution

To explore the influence of the MixColumns coefficient property on the security of the AES, in this paper we propose two new attacks on the 5-round AES variant with a secret S-Box based on Property 1 and 2 respectively. Our attacks are developed upon the newest technique called the exchange attack [1], we manage to transform the 5-round exchange attack to two key-recovery attacks. Compared with those previous attacks based on the MixColumns coefficient property, our 5-round attacks need only $2^{42.6}$ or 2^{46} chosen plaintexts, which are new records under the chosen-plaintext scenario. All the attacks on the 5-round AES related to the MixColumns coefficient property are listed in Table 1 for a convenient comparison.

Organization of This Paper

In Sect. 2, we introduce some background knowledge needed in this paper. In Sect. 3 and 4, we present two new attacks exploiting Property 1 and Property 2, respectively. We conclude this paper in Sect. 5.

[2] The number of all the 8-bit S-Boxes is $2^8!$ which is about $log_2^{(2^8!)} \approx 1684$ bits information. Totally, the security information is about $1684 + 128 = 1812$ bits.

Table 1. Attacks on the 5-round AES taking the mixcolumns coefficient property

Attack	Round	Data	Computation	Reference
Integral	5	2^{128} CC	$2^{129.6}$ XOR	[15]
Impossible differential	5	2^{102} CP	2^{107} M $\approx 2^{100.4}$ E*	[10]
Impossible differential	5	$2^{76.4}$ CP	$2^{81.5}$ M $\approx 2^{74.9}$ E	[9]
Integral	5	2^{96} CP	2^{96} E	[12]
Multiple-of-n	5	$2^{53.6}$ CP	$2^{55.6}$ M $\approx 2^{48.86}$ E	[9]
Zero difference	5	$2^{29.19}$ CP+2^{32}ACC	2^{31} XOR	[2]
Exchange	**5**	$\mathbf{2^{42.6}}$ **CP**	$\mathbf{2^{42.6}}$ E	**Sect.** 3
Exchange	**5**	$\mathbf{2^{46}}$ **CP**	$\mathbf{2^{46}}$ E	**Sect.** 4

CC: chosen ciphertexts, CP: chosen plaintexts, ACC: adaptive chosen ciphertexts
M: memory access, XOR: XOR operation, E: 5-round AES encryption
*: In [9,10], the authors used the scale that 100 times of memory access are approximately equivalent to 1 times of 5-round AES. In this paper, we use the same scale.

2 Preliminary

2.1 Description of the AES

The AES (Advanced Encryption Standard) [7] is an iterated block cipher with the substitution-permutation network (SPN). It has three versions with the key size 128, 192, 256 bits and the number of rounds is 10, 12, 14, respectively. The length of the block cipher is 128-bit and it will be initialized as a 4×4 matrix of bytes as values in the finite field $\mathbb{F}_{2^8}$ defined over the the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ (AES finite field). The round function of the AES, except the last one, applies four operations to every state matrix:

- SubBytes (SB) - each of the 16 bytes in the state matrix is replaced by another value getting from an 8-bit S-Box. In our attack the adversary does not know the exact information about the S-Box.
- ShiftRows (SR) - the i-th ($0 \leq i \leq 3$) row of the state matrix is rotated to the left by i position(s).
- MixColumns (MC) - each column of the state matrix is multiplied by an MDS matric M_{AES} from the left over the AES finite field. The invertible matrix M_{AES} is shown as follows, each byte of matrix is presented as hexadecimal.

$$M_{AES} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \tag{1}$$

- AddRoundKey(AK) - the state of the AES is XORed with the 128-bit round key.

In the first round an additional AK will be applied to the plaintext ahead the SB operation. And in the last round the MixColumns operation is omitted for convenient decryption. In this paper, we focus on the 5-round AES variant

where we consider the five full rounds of the AES keeping the last MC only for convenient description.

The AES Variant with A Secret S-Box. The target of this paper is an AES variant with a secret S-Box, i.e., the S-Box is replaced by a secret one and other structure and components are as the same as the original AES.

2.2 Notations

Let x denote a plaintext, a ciphertext, an intermediate state or a key. Then $x_{i,j}$ with $i, j \in \{0, 1, 2, 3\}$ denotes the byte located at the intersection of the i-th row and the j-th column. The secret key is usually denoted by k. We denote one round of the AES by R and denote r full rounds of the AES by R^r[3]. In this paper, we will also adopt the notations of the subspaces for the AES proposed initially in [10]. For a pair (x, x'), its dual pair $(\hat{x}, \hat{x}')$ is generated by exchanging the first diagonal between x and x'. We call a pair and its dual pair, i.e., $(x, x', \hat{x}, \hat{x}')$ a pair-of-pair. For a matrix or a vector v, we denote its transpose by v^T.

Subspaces of the AES. The subspace trial of the AES works with vectors and vector spaces over $\mathbb{F}_{2^8}^{4\times 4}$. We denote the unit vectors of $\mathbb{F}_{2^8}^{4\times 4}$ by $e_{0,0}, e_{0,1}, ..., e_{3,3}$ where $e_{i,j}$ has a single 1 in the intersection of the i-th row and the j-th column.

Definition 1 (Column Space [10]). *The column space C_i are defined as $C_i = \langle e_{0,i}, e_{1,i}, e_{2,i}, e_{3,i} \rangle$.*

Definition 2 (Diagonal and Inverse-Diagonal Space [10]). *The diagonal spaces D_i and inverse-diagonal spaces ID_i are defined as $D_i = \mathsf{SR}^{-1}(C_i)$ and $ID_i = \mathsf{SR}(C_i)$.*

Definition 3 (Mixed Space [10]). *The i-th mixed spaces M_i are defined as $M_i = \mathsf{MC}(ID_i)$.*

Definition 4 ([10]). *For $I \subseteq \{0, 1, 2, 3\}$ where $0 < |I| \leq 3$, let C_I, D_I, ID_I and M_I defined as*

$$C_I = \bigoplus_{i \in I} C_i, D_I = \bigoplus_{i \in I} D_i, ID_I = \bigoplus_{i \in I} ID_i, M_I = \bigoplus_{i \in I} M_i.$$

We refer readers to [10] for more details.

Next we introduce a useful one round subspace trail.

Lemma 1 ([10]). *For any coset $D_I \oplus a$ there exists a unique $b \in C_I^{\perp}$ such that after one round $R(D_I \oplus a)$ belongs to a coset of column space, i.e., $R(D_I \oplus a) = C_I \oplus b$. In other words, if $x \oplus x' \in D_I$, then $R(x) \oplus R(x') \in C_I$.*

[3] For the unity of description, we do not omit the last MC of R^r when we metion R^r.

2.3 Exchange Attack

The exchange attack is a new distinguisher proposed at Asiacrypt 2019 which can be used to attack the 5- and 6-round AES [1]. Since this paper only use the distinguishing attack on the 5-round AES, we only introduce some basic ideas about its application to the 5-round AES.

For a pair of states, if we exchange their first diagonals between the two values and get its dual pair, it is equivalent to swap the corresponding column after one round encryption. Furthermore, in some special cases, to exchange a column is equivalent to exchange a diagnoal. For example, if the difference of the state pair behaves like the rightmost state in Fig. 2, exchanging its first column is equivalent to exchange its first diagonal, because only the byte at the intersection of the first column and the first diagnoal is active.

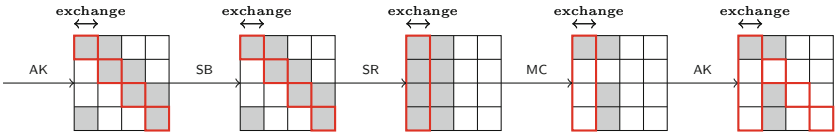

Fig. 2. Swapping the first column is equivalent to swap the first diagonal.

In [1], the authors modified a theorem from [14], which states an exchange-difference relation over 4 rounds of the AES.

Theorem 1 (4-round Exchange-Difference Relation [14]). *Let* $x, x' \in \mathbb{F}_{2^8}^{4\times 4}$*, exchange some diagonals between* x *and* x' *and get* $\hat{x}, \hat{x}'$*, then for* $J \subseteq \{0,1,2,3\}$ *and* $0 < |J| \leq 3$*,*

$$Pr(R^4(\hat{x}) \oplus R^4(\hat{x}') \in M_J | R^4(x) \oplus R^4(x') \in M_J) = 1.$$

According to the exchange attack illustrated in Fig. 2 [1], we choose a pair of plaintext $x, x' \in D_J \oplus a$ where $J = \{0,1\}$, and exchange the first diagonal to get its dual pair $\hat{x}, \hat{x}' \in C_I \oplus a$. With some probability $x \oplus x'$ and $\hat{x} \oplus \hat{x}'$ may satisfy a special difference pattern making that it is equivalent to exchange some diagonals of $(R(x), R(x'))$ to get $(R(\hat{x}), R(\hat{x}'))$. Then it meets the starting condition of Theorem 1, we can get a 5-round exchange-equivalent relation for the AES.

3 Improved Key-Recovery Attack Based on Property 1

In this section, we show how to combine Property 1 with the exchange attack to establish an improved key-recovery attack on the 5-round AES with a secret S-Box. The basic idea of this attack is to extend the 4-round exchange-difference relation (Theorem 1) to 5 rounds. In the attack, we first choose two plaintexts

p, p' from a subspace $S_0 = a \oplus D_I$ where $I = \{0, 1\}$, and expect that $R(p), R(p')$ will be in a specific subspace $S_1 = b \oplus C_I$ as follows,

$$S_1 \triangleq \left\{ b \oplus \begin{bmatrix} x_1 & x_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & x_3 & 0 & 0 \\ 0 & x_4 & 0 & 0 \end{bmatrix} \middle| x_1, x_2, x_3, x_4, b \in \mathbb{F}_{2^8} \right\}. \tag{2}$$

For two randomly drawn plaintexts $p, p' \in S_0$, the probability that $R(p) \oplus R(p') \in S_1$ is 2^{-32}. However, taking Property 1 into consideration and choosing p, p' carefully according to some secret key information, we can vary the probability of $R(p) \oplus R(p') \in S_1$ between the wrong and right key guess.

Once $R(p) \oplus R(p') \in S_1$, we can exchange the first diagonal between p and p' and get its dual pair $(\hat{p}, \hat{p}')$, thus $(R(p), R(p'))$ and $(R(\hat{p}), R(\hat{p}'))$ are two pairs satisfying the starting condition of Theorem 1. Hence, $R^5(p) \oplus R^5(p')$ and $R^5(\hat{p}) \oplus R^5(\hat{p}')$ will be always in the same M_J for certain $J \subseteq \{0, 1, 2, 3\}$ at the same time. For sake of convenience, in this section we call such pair-of-pair $(p, p', \hat{p}, \hat{p}')$ a **right pair-of-pair**.

Details. Based on Property 1, if the four input bytes of MC have two zero-difference values and the difference of the remaining two bytes are equal, the output vector will have one zero-difference byte with probability 1. Without loss of generality, we assume the input difference is $[a, 0, 0, a]^T$, then

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a \\ 0 \\ 0 \\ a \end{bmatrix} = \begin{bmatrix} 3a \\ 0 \\ 2a \\ a \end{bmatrix}. \tag{3}$$

It can be seen that the second value of the output difference must be zero. Then if the second column of the input difference of MC is really the patten such as $[a, 0, 0, a]^T$ where $a \in \mathbb{F}_{2^8} \backslash \{0\}$, the probability that $R(p) \oplus R(p') \in S_1$ (Eq. 2) will be 2^{-24} rather than 2^{-32}. For this reason, we define the set $A_{z,\delta}$ as follows,

$$A_{z,\delta} \triangleq \left\{ a \oplus \begin{bmatrix} y_0 & z & 0 & 0 \\ 0 & y_1 & 0 & 0 \\ 0 & 0 & y_2 & 0 \\ z \oplus \delta & 0 & 0 & y_3 \end{bmatrix} \middle| \forall y_0, y_1, y_2, y_3, a \in F_{2^8} \right\} \text{ where } z, \delta \in F_{2^8}, \tag{4}$$

and then choose two different plaintexts $p \in A_{z_0,\delta}$ and $p' \in A_{z_1,\delta}$ where $z_0 \neq z_1$.

Let the two secret key bytes which are XORed with $p_{0,1}$ (Resp. $p'_{0,1}$) and $p_{3,0}$ (Resp. $p'_{3,0}$) be $k_{0,1}$ and $k_{3,0}$, respectively. After $\mathsf{f} \triangleq \mathsf{SR} \circ \mathsf{SB} \circ \mathsf{AK}$ operation, the second column of $\mathsf{f}(p) \oplus \mathsf{f}(p')$ is

$$(\mathsf{f}(p) \oplus \mathsf{f}(p'))_{C_1} = \begin{bmatrix} \text{S-Box}(z_0 \oplus k_{0,1}) \oplus \text{S-Box}(z_1 \oplus k_{0,1}) \\ 0 \\ 0 \\ \text{S-Box}(z_0 \oplus \delta \oplus k_{3,0}) \oplus \text{S-Box}(z_1 \oplus \delta \oplus k_{3,0}) \end{bmatrix}.$$

To meet the condition shown in Eq. 3, Eq. 5 should be met,

$$\text{S-Box}(z_0 \oplus k_{0,1}) \oplus \text{S-Box}(z_1 \oplus k_{0,1}) = \text{S-Box}(z_0 \oplus \delta \oplus k_{3,0}) \oplus \text{S-Box}(z_1 \oplus \delta \oplus k_{3,0}) \quad (5)$$

Since the S-Box is a secret permutation, Eq. 5 has only two solutions, i.e.,

$$\delta = k_{0,1} \oplus k_{3,0} \text{ or } \delta = z_0 \oplus z_1 \oplus k_{0,1} \oplus k_{3,0}.$$

If we let δ run through all values in $\mathbb{F}_{2^8}$, we can guarantee that there are at least two values of δ leading that Eq. 5 holds. For sake of simplicity, we call the two δ **right** δ and other values **wrong** δ. For right δ, the probability that $R(p) \oplus R(p') \in S_1$ will be 2^{-24}. For wrong δ, the probability is still 2^{-32}. Combinining with Theorem 1, we conclude the following proposition,

Proposition 1. *Let $p \in A_{z_0,\delta}$ and $p' \in A_{z_1,\delta}$. $(\hat{p}, \hat{p}')$ is the dual pair of (p, p'). If δ is right, for certain M_J with $|J| = 3$,*

$$Pr(R^5(p) \oplus R^5(p') \in M_J \wedge R^5(\hat{p}) \oplus R^5(\hat{p}') \in M_J) \approx 2^{-54}.$$

While for wrong δ,

$$Pr(R^5(p) \oplus R^5(p') \in M_J \wedge R^5(\hat{p}) \oplus R^5(\hat{p}') \in M_J) \approx 2^{-62}.$$

Proof. If two pairs satisfy the starting condition of Theorem 1, they will be in the same M_J at the same time after 4 rounds of encryption. Let $|J| = 3$, the probability for the two pairs being a right pair-of-pair is 2^{-30} since we have four choices of J.

For wrong δ, the starting condition of Theorem 1 is statisfied with probability 2^{-32}. Then, the probability for the two pairs being a right pair-of-pair is about 2^{-62}, which is consistent with the random case. While for right δ, the starting condition is met with probability 2^{-24}, so the probability for the two pairs being a right pair-of-pair is 2^{-54}. □

Finding δ Candidates. We can take advantage of Proposition 1 to find the right δ that implies $k_{0,1} \oplus k_{3,0}$. The process for finding δ is illustrated in Algorithm 1. For each candidate $\delta \in \mathbb{F}_{2^8}$, we find collision pairs and check whether there is at least one collision pair satisfying that its dual pair is also a collision pair. We explain briefly some crucial lines in Algorithm 1.

Line 4. For $A_{z_0,\delta}$ and $A_{z_1,\delta}$, we require that the i-th plaintexts in $A_{z_0,\delta}$ and $A_{z_1,\delta}$ should have the same value in the first diagonal. In this way, $(c^i_{z_0}, c^j_{z_1})$ must be the dual pair of $(c^i_{z_1}, c^j_{z_0})$. We can prepare a subset of D_0 with size 2^N and use it to generate the two sets $A_{z_0,\delta}$ and $A_{z_1,\delta}$ where $z_0 \neq z_1$.

Line 14. Since we have stored all the ciphertexts in tables, we only need to store the indexes of ciphertexts into the two hash tables. If the i-th lines of $\mathcal{T}_{z_0}$ and $\mathcal{T}_{z_1}$ are not empty simultaneously, we find a collision pair pointed by the corresponding indexes.

Algorithm 1. Finding δ Candidates (Property 1)

procedure CORE(z_0, z_1, r, c) ▷ Return a set containing the possible right δ
  for Each $\delta \in \mathbb{F}_{2^8}$ **do**
    Initialize 2 sequence tables $\mathcal{C}_{z_0}$, $\mathcal{C}_{z_1}$, 1 table Δ
    Prepare two sets $A_{z_0,\delta}, A_{z_1,\delta}$ with 2^{29} plaintexts ▷ Make sure $A_{z_0,\delta}[i]_{D_0} = A_{z_1,\delta}[i]_{D_0}$, according to Equation 4
    for $i = 0; i < 2^{29}; i = i + 1$ **do**
      for $j = 0; j < 2; j = j + 1$ **do**
        $c^i_{z_j} \leftarrow R^5(p^i_{z_j})$ ▷ $p^i_{z_j}$ is the i-th plaintext in $A_{z_j,\delta}$
        $\mathcal{C}_{z_j}[i] \leftarrow c^i_{z_j}$ ▷ Store $c^i_{z_j}$
      end for
    end for
    for $k = 0; k < 4; k = k + 1$ **do** ▷ For each M_k space, search for collisions
      Initialize 2 hash tables $\mathcal{T}_{z_0}, \mathcal{T}_{z_1}$
      for $i = 0; i < 2^{29}; i = i + 1$ **do**
        for $j = 0; j < 2; j = j + 1$ **do**
          $\mathcal{T}_{z_j}[\mathsf{MC}^{-1}(c^i_{z_j})_{ID_k}] \leftarrow \mathsf{index}(c^i_{z_j})$ ▷ $\mathsf{index}(c^i_{z_j}) = i$
        end for
      end for
      for $i = 0; i < 2^{32}; i = i + 1$ **do** ▷ For each line of $\mathcal{T}_{z_0}$ and $\mathcal{T}_{z_1}$
        if there is a collision pair with indexes (i_0, i_1) and $i_0 \neq i_1$ **then**
          $c^{i_1}_{z_0} \leftarrow C_{z_0}[i_1]$, $c^{i_0}_{z_1} \leftarrow C_{z_1}[i_0]$ ▷ $(c^{i_0}_{z_0}, c^{i_1}_{z_1})$ and $(c^{i_1}_{z_0}, c^{i_0}_{z_1})$ are dual pairs
          if $c^{i_1}_{z_0} \oplus c^{i_0}_{z_1} \in M_k$ **then** ▷ $(c^{i_1}_{z_0}, c^{i_0}_{z_1})$ is also collided
            $\Delta \leftarrow \delta$
          end if
        end if
      end for
    end for
  end for
  return Δ
end procedure

Algorithm 2. Remove wrong δ

procedure REMOVE(Δ, z_0, z_1)
  for $\delta \in \Delta$ **do**
    if $\delta \oplus z_1 \oplus z_2 \notin \Delta$ **then**
      Remove δ from Δ
    end if
  end for
  return Δ
end procedure

Line 20. $(c^i_{z_0}, c^j_{z_1})$ and $(c^i_{z_1}, c^j_{z_0})$ are dual pairs, then we need to check if $c^i_{z_1} \oplus c^j_{z_0} \in M_k$.

Determine the Size of $A_{z_0,\delta}$ And $A_{z_1,\delta}$. For $A_{z_0,\delta}$ and $A_{z_1,\delta}$ with 2^N elements, we can obtain 2^{2N} pairs (p, p') by choosing $p \in A_{z_0,\delta}$ and $p' \in A_{z_1,\delta}$. By exchanging the first diagonal, we get 2^{2N-1} pair-of-pairs such as $(p, p', \hat{p}, \hat{p}')$.

For 5-round AES, these 2^{2N-1} pair-of-pairs can be regarded as 2^{2N-1} Bernoulli trials, and the number of right pair-of-pairs should obey Binomial distribution $\mathcal{B}(2^{2N-1}, 2^{-54})$ when δ is right. Otherwise, it will obey $\mathcal{B}(2^{2N-1}, 2^{-62})$. Let N_r and N_w be the number of right pair-of-pairs for right and wrong δ, respectively.
For right δ,

$$Pr(N_r \geq 1) = 1 - Pr(N_r = 0) = 1 - (1 - 2^{-54})^{2^{2N-1}} \approx 1 - exp(-2^{2N-1-54}).$$

For wrong δ,

$$Pr(N_w \geq 1) = 1 - Pr(N_w = 0) = 1 - (1 - 2^{-62})^{2^{2N-1}} \approx 1 - exp(-2^{2N-1-62}).$$

When we take $N = 29$, $Pr(N_r \geq 1) \approx 0.9997$ while $Pr(N_w \geq 1) \approx 0.0308$, which means we can distinguish the right δ from the wrong δ.

Determining the Exact $k_{0,1} \oplus k_{3,0}$. Either of the right δ including $\delta = k_{0,1} \oplus k_{3,0}$ and $\delta = k_{0,1} \oplus k_{3,0} \oplus z_0 \oplus z_1$ will bring at least one right pair-of-pair with probability about 0.9997. Therefore, they will be both returned by Algorithm 1 with probability $0.9997^2 \approx 0.9994$. At the same time, the probability for a wrong δ being recommended is 0.0308. For all the $2^8 - 2$ wrong δ, on average there will be $(2^8 - 2) \times 0.0308 \approx 8$ wrong δ which are also recommended. All the δ candidates are inserted into a set Δ, which is returned by Algorithm 1 finally.

To remove the wrong δ from Δ, we XOR $z_0 \oplus z_1$ with each value in Δ. For right δ, $\delta \oplus z_0 \oplus z_1$ should be also in Δ in a high probability (0.9994) while for wrong δ, the probability is about 2^{-8}. The method of removing wrong δ is shown in Algorithm 2.

Now the set Δ contains only $k_{0,1} \oplus k_{3,0}$ and $k_{0,1} \oplus k_{3,0} \oplus z_0 \oplus z_1$. To determine the exact right key byte, we have to call Algorithm 1 and Algorithm 2 again with (z_2, z_3) where $z_2 \oplus z_3 \neq z_0 \oplus z_1$. With $\Delta' = \{k_{0,1} \oplus k_{3,0}, k_{0,1} \oplus k_{3,0} \oplus z_2 \oplus z_3\}$ returned, we can easily determine the right $k_{0,1} \oplus k_{3,0}$ by comparing Δ and Δ'. Therefore we recover one byte key information with $0.9994^2 \approx 0.9988$ success probability. The process is illustrated in Algorithm 3.

The procedure $\mathsf{RecoverKeyByte}(r, c)$ (Algorithm 3) can be used to recover $k_{r,c} \oplus k_{r+1,c+1}$[4]. Since the equal bytes in MC matrix are all adjacent, for the i-th diagonal of the key state, we can recover $k_{0,i} \oplus k_{1,i+1}$, $k_{1,i+1} \oplus k_{2,i+2}$, $k_{2,i+2} \oplus k_{3,i+3}$ and $k_{3,i+3} \oplus k_{0,i}$. However, from any three out of the four values we can derive the remaining one, which means we can recover three bytes of useful key information for one diagonal. For the four diagonals of key state, we can recover 12 bytes of key information, i.e. we can get the secret key up to 2^{32} variants.

[4] In this paper, the addition of indexes are modulo 4.

Algorithm 3. Recover the real key $k_{r,c} \oplus k_{r+1,c+1}$ (Property 1)

```
1: procedure RecoverKeyByte(r, c)        ▷ Recover k_{r,c} ⊕ k_{r+1,c+1} with 99.88% probability
2:     Allocate z_0, z_1, z_2, z_3 s.t. z_0 ⊕ z_1 ≠ z_2 ⊕ z_3
3:     Δ_0 ← Core(z_0, z_1, r, c)
4:     if |Δ_0| == 0 then
5:         return ⊥
6:     else
7:         Δ_0 ← Remove( Δ_0, z_0, z_1)
8:     end if
9:     Δ_1 ← Core(z_2, z_3, r, c)
10:    if |Δ_1| == 0 then
11:        return ⊥
12:    else
13:        Δ_1 ← Remove( Δ_1, z_0, z_1)
14:    end if
15:    if Δ_0, Δ_1 have the same value then
16:        return δ ← (Δ_0, Δ_1)        ▷ Right k_{r,c} ⊕ k_{r+1,c+1} must lie in both set
17:    else
18:        return ⊥
19:    end if
20: end procedure
```

Data Complexity. From Algorithm 1, for every $\delta \in \mathbb{F}_{2^8}$ we use four sets $A_{z_i,\delta}$ for $i = 0, 1, 2, 3$ each with 2^{29} plaintexts. Therefore we need $2^{29} \times 2^8 \times 4 = 2^{39}$ chosen plaintexts to recover one byte key. In order to recover 12 key bytes, the total data complexity is $2^{39} \times 12 \approx 2^{42.6}$ chosen plaintexts.

Computation Complexity. Firstly, we evaluate the complexity of Algorithm 1. For each possible $\delta \in \mathbb{F}_{2^8}$ we encrypt two sets $A_{z_0,\delta}$ and $A_{z_1,\delta}$ each with 2^{29} plaintexts, this operation needs $2^{29} \times 2 = 2^{30}$ 5-round encryptions. After obtaining 2^{30} ciphertexts, we insert them into C_{z_0} and C_{z_1} with 2^{30} table-lookups. To insert all the ciphertexts to T_{z_0} and T_{z_1}, we need 2^{30} table-lookups again. Then we compare each line of T_{z_0} and T_{z_1} to find collision pairs, it requires $2 \times 2^{32} = 2^{33}$ table-lookups. For the two sets $A_{z_0,\delta}$ and $A_{z_1,\delta}$ each with 2^{29} chosen plaintexts, on average we can obtain $2^{29} \times 2^{29} \times 2^{-32} = 2^{26}$ collision pairs. Once we find a collision pair $(c^i_{z_0}, c^j_{z_1})$, we need a time of XOR to check whether $(c^i_{z_1}, c^j_{z_0})$ is collided. These memory operations above need about 2^{33} table-lookups. Considering we have four possible M_k, the whole memory operations cost 2^{35} table-lookups. We use the convention that 100 times of table look-ups are equivalent to one time 5-round encryption. Hence, encrypting the plaintexts is dominant in the time complexity, which requires 2^{30} 5-round encryptions for each δ.

To determine the exact one byte information of key (Algorithm 3), the time complexity is $2^8 \times 2 \times 2^{30} = 2^{39}$ 5-round encryptions. Recovering 12 bytes key requires $2^{39} \times 12 \approx 2^{42.6}$ times of 5-round encryption.

Memory Complexity. We allocate 2 sequence tables with size 2^{29} and 2 hash tables with size 2^{32}. Since these tables can be reused, the total memory complexity is about $2^{32} \times 2 + 2^{29} \times 2 \approx 2^{33}$ 128-bit blocks.

Practical Verification. Using C/C++ implementation, we practically verified our key-recovery attack on a small-scale variant of the AES as presented in [5]. The block size of the small-scale AES is 64 bits, and each word is a 4-bit nibble in the state matrix. We simply recover one byte of the secret key XOR in our experiment. The experimental result supports our theory.[5]

4 Improved Key-Recovery Attack Based on Property 2

Similar to the exchange attack based on Property 1, we can also combine Property 2 of MC matrix with exchange attack to realize the key recovery attack with a secret S-Box. To exploit Property 2, we focus on another subspace S_1' that two plaintexts $p, p' \in D_I, I = \{0, 1\}$ should fall into after the first round encryption.

$$S_1' \triangleq \left\{ b \oplus \begin{bmatrix} a_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a_3 & 0 & 0 \\ a_2 & a_4 & 0 & 0 \end{bmatrix} \middle| a_1, a_2, a_3, a_4, b \in \mathbb{F}_{2^8} \right\}. \tag{6}$$

If we exchange the first diagonal between p and p', it is equivalent to exchange the first column between $R(p)$ and $R(p')$. Since $R(p), R(p') \in S_1'$, it is also equivalent to exchange the first and the fourth diagonals between $R(p)$ and $R(p')$.

Details. Property 2 of MC says that three elements in each row can be XORed to zero. If the input difference of the four bytes of MC has three equal values and the remaining one value is zero, the output difference will have two zero-difference byte with probability 1. Without loss of generality, we assume the input difference is $[a, a, a, 0]^T$, then

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a \\ a \\ a \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2a \\ 3a \end{bmatrix} \tag{7}$$

It can be seen that there are two zero-difference values in the output difference with probability 1. Then if the input difference of MC is really the pattern such as $[a, a, a, 0]^T$ for any $a \in \mathbb{F}_{2^8} \backslash \{0\}$. To achieve it, we define the set A_{w,δ_1,δ_2} as follows,

$$A_{w,\delta_1,\delta_2} \triangleq \left\{ a \oplus \begin{bmatrix} y_1 & w & 0 & 0 \\ 0 & y_2 & w \oplus \delta_1 & 0 \\ 0 & 0 & y_3 & w \oplus \delta_2 \\ 0 & 0 & 0 & y_4 \end{bmatrix} \middle| \forall y_0, y_1, y_2, y_3 \in \mathbb{F}_{2^8} \right\} \tag{8}$$

where $w, \delta_1, \delta_2 \in \mathbb{F}_{2^8}$.

[5] https://github.com/anxin19/5-round-AES-keyrecoveryattack.git.

We choose two different plaintexts $p \in A_{w_0,\delta_1,\delta_2}, p' \in A_{w_1,\delta_1,\delta_2}$. Let the key bytes XORed with $p_{0,1}, p_{1,2}, p_{2,3}$ (Resp. $p'_{0,1}, p'_{1,2}, p'_{2,3}$) are $k_{0,1}, k_{1,2}, k_{2,3}$, respectively. After the operation $\mathsf{f} = \mathsf{SR} \circ \mathsf{SB} \circ \mathsf{AK}$, the difference between the second column of $\mathsf{f}(p)$ and $\mathsf{f}(p')$ is

$$\mathsf{f}(p)_{C_1} \oplus \mathsf{f}(p')_{C_1} = \begin{bmatrix} \text{S-Box}(w_0 \oplus k_{0,1}) \oplus \text{S-Box}(w_1 \oplus k_{0,1}) \\ \text{S-Box}(w_0 \oplus \delta_1 \oplus k_{1,2}) \oplus \text{S-Box}(w_1 \oplus \delta_1 \oplus k_{1,2}) \\ \text{S-Box}(w_0 \oplus \delta_2 \oplus k_{2,3}) \oplus \text{S-Box}(w_1 \oplus \delta_2 \oplus k_{2,3}) \\ 0 \end{bmatrix} \tag{9}$$

To meet the condition shown in Eq. 7, the following equation should be satisfied (denote S-Box$(\cdot)$ by S$(\cdot)$ for short),

$$\begin{cases} \mathrm{S}(w_0 \oplus k_{0,1}) \oplus \mathrm{S}(w_1 \oplus k_{0,1}) = \mathrm{S}(w_0 \oplus \delta_1 \oplus k_{1,2}) \oplus \mathrm{S}(w_1 \oplus \delta_1 \oplus k_{1,2}) \\ \mathrm{S}(w_0 \oplus k_{0,1}) \oplus \mathrm{S}(w_1 \oplus k_{0,1}) = \mathrm{S}(w_0 \oplus \delta_2 \oplus k_{2,3}) \oplus \mathrm{S}(w_1 \oplus \delta_2 \oplus k_{2,3}) \end{cases} \tag{10}$$

Since the S-Box is a secret permutation, there can be only four kinds of solutions,

$$\begin{aligned} &(\delta_1, \delta_2) = (k_{0,1} \oplus k_{1,2}, k_{0,1} \oplus k_{2,3}) \text{ or} \\ &(\delta_1, \delta_2) = (k_{0,1} \oplus k_{1,2}, w_0 \oplus w_1 \oplus k_{0,1} \oplus k_{2,3}) \text{ or} \\ &(\delta_1, \delta_2) = (w_0 \oplus w_1 \oplus k_{0,1} \oplus k_{1,2}, k_{0,1} \oplus k_{2,3}) \text{ or} \\ &(\delta_1, \delta_2) = (w_0 \oplus w_1 \oplus k_{0,1} \oplus k_{1,2}, w_0 \oplus w_1 \oplus k_{0,1} \oplus k_{2,3}) \end{aligned} \tag{11}$$

Similar with the attack in Sect. 3, we let (δ_1, δ_2) run through all possible values in $F_{2^8} \times F_{2^8}$. There will be at least four values of (δ_1, δ_2) that make Eq. 10 hold. We call the four (δ_1, δ_2) in Eq. 11 **right** (δ_1, δ_2) and the other values **wrong** (δ_1, δ_2). For right (δ_1, δ_2), the probability of $R(p^1) \oplus R(p^2) \in S'_1$ is 2^{-16} while for wrong (δ_1, δ_2) the probability is still 2^{-32}. Combining with Theorem 1, we conclude the following proposition.

Proposition 2. *Let $p \in A_{w_0,\delta_1,\delta_2}$ and $p' \in A_{w_1,\delta_1,\delta_2}$. $(\hat{p}, \hat{p}')$ is generated by exchanging the first diagonal between p and p'. If (δ_1, δ_2) is right, for certain M_J with $|J| = 3$,*

$$Pr(R^5(p) \oplus R^5(p') \in M_J \wedge R^5(\hat{p}) \oplus R^5(\hat{p}') \in M_J) \approx 2^{-46},$$

while for wrong (δ_1, δ_2),

$$Pr(R^5(p) \oplus R^5(p') \in M_J \wedge R^5(\hat{p}) \oplus R^5(\hat{p}') \in M_J) \approx 2^{-62}.$$

The proof of Proposition 2 is similar to the Proposition 1, we omit it here.

Finding (δ_1, δ_2) Candidates. We can also take advantage of Proposition 2 to find the right (δ_1, δ_2) which implies the key byte information $k_{0,1} \oplus k_{1,2}$ and $k_{0,1} \oplus k_{2,3}$. The process for finding (δ_1, δ_2) candidates is similar to Algorithm 1 except we need to guess two key byte difference. The process is illustrated in Algorithm 4.

Determine the Size of $A_{w_0,\delta_1,\delta_2}$ And $A_{w_1,\delta_1,\delta_2}$. If the size of $A_{w_0,\delta_1,\delta_2}$ and $A_{w_1,\delta_1,\delta_2}$ are both 2^M, we can obtain 2^{2M} pairs of (p, p') by choosing $p \in A_{w_0,\delta_1,\delta_2}$ and $p' \in A_{w_1,\delta_1,\delta_2}$. By exchanging the first diagonal, we can get totally 2^{2M-1} pair-of-pairs such as $(p, p', \hat{p}, \hat{p}')$. If $R^5(p) \oplus R^5(p') \in M_J$ and $R^5(\hat{p}) \oplus R^5(\hat{p}') \in M_J$ for $|J| = 3$ hold at the same time, then we call such $(p, p', \hat{p}, \hat{p}')$ a **right** pair-of-pair. Consider the number of right pair-of-pairs,
For right (δ_1, δ_2),

$$Pr(M_r \geq 1) = 1 - Pr(M_r = 0) = 1 - (1 - 2^{-46})^{2^{2M-1}} \approx 1 - exp(-2^{2M-1-46}).$$

For wrong (δ_1, δ_2),

$$Pr(M_w \geq 1) = 1 - Pr(M_w = 0) = 1 - (1 - 2^{-62})^{2^{2M-1}} \approx 1 - exp(-2^{2M-1-62}).$$

When we take $M = 25$, $Pr(M_r \geq 1) \approx 0.9997$ while $Pr(M_w \geq 1) \approx 0.0001$ which means we can distinguish the right (δ_1, δ_2) from the wrong ones.

Determining $k_{0,1} \oplus k_{1,2}$ and $k_{0,1} \oplus k_{2,3}$. In this attack, we also have a probability $1 - (1 - 0.0001)^{2^{16}-4} \approx 0.9986$ nearly close to 1 to return at least one wrong (δ_1, δ_2). On average, approximately $(2^{16} - 4) \times 0.0001 \approx 7$ wrong (δ_1, δ_2) will be returned. To remove the wrong (δ_1, δ_2) from Δ, we XOR $w_0 \oplus w_1$ with the two components of each value in Δ and check whether the result is in Δ or not as Algorithm 5. To determine the exact $(k_{0,1} \oplus k_{1,2}, k_{0,1} \oplus k_{2,3})$, we need to use additional two sets $A_{w_2,\delta_1,\delta_2}$ $A_{w_3,\delta_1,\delta_2}$ where $(w_0, w_1) \neq (w_2, w_3)$ with 2^{25} plaintexts and do the same. Finally, the probability that we succeed to recover the two key bytes with probability $0.9997^{4\times2} \approx 0.9976$. The process is illustrated in Algorithm 6 .

After we recover two key bytes information, we can take the same strategy to recover another different key byte information in the same diagonal. At last we can recover 12 key byte difference, i.e., we can get the entire secret key up to 2^{32} variants.

Data Complexity. According to Algorithm 4, for each (δ_1, δ_2) we use two sets $A_{w_0,\delta_1,\delta_2}$ and $A_{w_1,\delta_1,\delta_2}$ each with 2^{25} plaintexts. Additional two sets $A_{w_2,\delta_1,\delta_2}$ and $A_{w_3,\delta_1,\delta_2}$ are also required to find the exact two key byte information. Therefore, totally we need $2^{25} \times 2^{16} \times 2 \times 2 = 2^{43}$ chosen plaintexts to recover two key bytes. To find the 12 bytes key information, the total data complexity is about $2^{43} \times 8 = 2^{46}$.

Computation Complexity. Encrypting two sets with 2^{25} plaintexts we need $2^{25} \times 2 = 2^{26}$ 5-round encryption which is the donimant in the complexity of Algorithm 4. The total time complexity is about $2^{26} \times 2^{16} \times 2 \times 8 = 2^{46}$ 5-round encryption.

Memory Complexity. We allocate two sequence tables with size 2^{25} to store the two ciphertext sets and additionally 2 hash tables with size 2^{32}. The memory complexity is finally 2^{33} 128-bit blocks.

Algorithm 4. Finding (δ_1, δ_2) Candidates (Property 2)

procedure CORE$'(w_0, w_1, r, c)$ ▷ Return a set containing possible (δ_1, δ_2)
  for Each $(\delta_1, \delta_2) \in \mathbb{F}_{2^8} \times \mathbb{F}_{2^8}$ **do**
    Initialize 2 sequence tables $\mathcal{C}_{w_0}$ and $\mathcal{C}_{w_1}$, 1 table Δ
    Prepare two sets $A_{w_0,\delta_1,\delta_2}, A_{w_1,\delta_1,\delta_2}$ with 2^{25} plaintexts each as Eq. 8
    for $i = 0; i < 2^{25}; i = i + 1$ **do**
      for $j = 0; j < 2; j = j + 1$ **do**
        $c^i_{w_j} \leftarrow R^5(p^i_{w_j})$
        $\mathcal{C}_{w_j}[i] \leftarrow c^i_{w_j}$ ▷ Push back $c^i_{w_j}$ into sequence table
      end for
    end for
    for $k = 0; k < 4; k = k + 1$ **do**
      Initialize 2 hash tables $\mathcal{T}_{w_0}, \mathcal{T}_{w_1}$
      for $i = 0; i < 2^{25}; i = i + 1$ **do**
        for $j = 0; j < 2; j = j + 1$ **do**
          $\mathcal{T}_{w_j}[\mathsf{MC}^{-1}(c^i_{w_j})_{ID_k}] \leftarrow \mathsf{index}(c^i_{w_j})$ ▷ Insert the index of $c^i_{w_j}$ into hash table
        end for
      end for
      for $i = 0; i < 2^{32}; i = i + 1$ **do**
        if there is a collision pair with indexes (i_0, i_1) and $i_0 \neq i_1$ **then**
          $c^{i_1}_{w_0} \leftarrow C_{w_0}[i_1]$, $c^{i_0}_{w_1} \leftarrow C_{w_1}[i_0]$ ▷ $(c^{i_0}_{w_0}, c^{i_1}_{w_1})$ and $(c^{i_1}_{w_0}, c^{i_0}_{w_1})$ are dual pairs
          if $c^{i_1}_{w_0} \oplus c^{i_0}_{w_1} \in M_k$ **then** ▷ $(c^{i_1}_{w_0}, c^{i_0}_{w_1})$ is also collided
            $\Delta \leftarrow \delta$
          end if
        end if
      end for
    end for
  end for
  return Δ
end procedure

Algorithm 5. Remove wrong (δ_1, δ_2)

procedure REMOVE$'(\ \Delta, w_0, w_1)$
  for each $(\delta_1, \delta_2) \in \Delta$ **do**
    if $(\delta_1 \oplus w_0 \oplus w_1, \delta_2 \oplus w_0 \oplus w_1) \notin \Delta$ **then**
      Remove$'$ (δ_1, δ_2) from Δ
    end if
  end for
  return Δ
end procedure

Algorithm 6. Recover $k_{r,c} \oplus k_{r+1,c+1}$ and $k_{r,c} \oplus k_{r+2,c+2}$ (Property 2)

procedure RecoverKeyByte$'(r, c, t)$ ▷ Recover $k_{r,c} \oplus k_{r+1,c+1}$ and $k_{r,c} \oplus k_{r+2,c+2}$ with 99.76% success probability
  Allocate w_0, w_1, w_2, w_3 s.t. $w_0 \oplus w_1 \neq w_2 \oplus w_3$
  $\Delta_0 \leftarrow$ Core$'(w_0, w_1, r, c)$
  if $|\Delta_0| == 0$ **then**
    return $\perp$ ▷ Fail
  else
    $\Delta_0' \leftarrow$ Remove$'(\Delta_0, w_0 \oplus w_1)$
  end if
  $\Delta_1 \leftarrow$ Core$'(w_2, w_3, r, c)$
  if $|\Delta_1| == 0$ **then**
    return $\perp$
  else
    $\Delta_1' \leftarrow$ Remove$'(\Delta_1, w_2 \oplus w_3)$
  end if
  if Δ_0', Δ_1' have the same value **then**
    return $(\delta_1, \delta_2) \leftarrow (\Delta_0', \Delta_1')$ ▷ Right $k_{r,c} \oplus k_{r+1,c+1}$ and $k_{r,c} \oplus k_{r+2,c+2}$ must lie in both sets
  else
    return $\perp$
  end if
end procedure

5 Conclusion

In this paper, we explore the impact of the MC coefficient property on the security of the AES variant with a secret S-Box. We provide two attacks based on Property 1 and Property 2 respectively and achieve the best record in terms of the complexity under chosen-plaintext scenario. Such attacks remind us to notice the choice of MC matrix for AES-like ciphers.

To our best knowledge, no previous attacks on the AES have taken advantage of other properties except the branch number of the MC matrix. It means that we may substitute any other MDS matrix free of Property 1 or 2^6 for the AES MC matrix without hazarding its security against other attacks. In [9], Grassi showed that about only 6.87% among all the MDS matrices have the two kinds of properties. Nevertheless, the choice of MC is still a difficult work since we should consider the performance of the cipher. The MC matrix of AES is already qualified for its pretty low weight, thus it is an interesting open question how to choose a proper MDS matrix without the particular coefficient property and achieve the same or even higher efficiency simultaneously.

Acknowledgement. We thank the anonymous reviewers for their valuable comments. This work is supported by the National Key Research and Development Project No. 2018YFA0704702, Major Scientific and Technological Innovation Project of Shandong

[6] Its inverse matrix should not have Property 1 or 2.

Province, China under Grant No. 2019JZZY010133, National Natural Science Foundation of China (NSFC) under Grant No. 61572293, 61502276 and 61692276.

References

1. Bardeh, N.G., Rønjom, S.: The exchange attack: how to distinguish six rounds of AES with 288.2 chosen plaintexts. In: Galbraith, S., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 347–370. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34618-8_12
2. Bardeh, N.G., Rønjom, S.: Practical attacks on reduced-round AES. In: Buchmann, J., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2019. LNCS, vol. 11627, pp. 297–310. SPringer, Cham (2019). https://doi.org/10.1007/978-3-030-23696-0_15
3. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: Menezes, A.J., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-38424-3_1
4. Bogdanov, A., Rijmen, V.: Linear hulls with correlation zero and linear cryptanalysis of block ciphers. Des. Codes Crypt. **70**(3), 369–383 (2012). https://doi.org/10.1007/s10623-012-9697-z
5. Cid, C., Murphy, S., Robshaw, M.J.B.: Small scale variants of the AES. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LnCS, vol. 3557, pp. 145–162. Springer, Heidelberg (2005). https://doi.org/10.1007/11502760_10
6. Cui, T., Sun, L., Chen, H., Wang, M.: Statistical integral distinguisher with multi-structure and its application on AES. In: Pieprzyk, J., Suriadi, S. (eds.) ACISP 2017, Part I. LNCS, vol. 10342, pp. 402–420. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60055-0_21
7. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography. Springer, Heidelberg (2002). https://doi.org/10.1007/978-3-662-04722-4
8. Daemen, J., Rijmen, V.: Security of a wide trail design. In: Menezes, A., Sarkar, P. (eds.) INDOCRYPT 2002. LNCS, vol. 2551, pp. 1–11. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36231-2_1
9. Grassi, L.: Mixcolumns properties and attacks on (round-reduced) AES with a single secret S-box. In: Smart, N. (ed.) CT-RSA 2018. LNCS, vol. 10808, pp. 243–263. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76953-0_13
10. Grassi, L., Rechberger, C., Rønjom, S.: Subspace trail cryptanalysis and its applications to AES. IACR Trans. Symmetric Cryptol. **2016**(2), 192–225 (2016)
11. Grassi, L., Rechberger, C., Rønjom, S.: A new structural-differential property of 5-round AES. In: Coron, J.S., Nielsen, J. (eds.) EUROCRYPT 2017, Part II. LNCS, vol. 10211, pp. 289–317. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56614-6_10
12. Hu, K., Cui, T., Gao, C., Wang, M.: Towards key-dependent integral and impossible differential distinguishers on 5-round AES. In: Cid, C., Jacobson Jr., M. (eds.) SAC 2018. LNCS, vol. 11349, pp. 139–162. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-10970-7_7
13. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48285-7_33
14. Rønjom, S., Bardeh, N.G., Helleseth, T.: Yoyo tricks with AES. ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 217–243. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_8

15. Sun, B., Liu, M., Guo, J., Qu, L., Rijmen, V.: New insights on AES-Like SPN ciphers. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 605–624. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_22
16. Tiessen, T., Knudsen, L.R., Kölbl, S., Lauridsen, M.M.: Security of the AES with a secret S-box. In: Leander, G. (ed.) FSE 2015. LNCS, vol. 9054, pp. 175–189. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48116-5_9