Security Bound of Biclique Attacks on AES-128

Xiaoli Dong[1] and Jie Chen[2]
(Corresponding author: Xiaoli Dong)
School of Cyberspace Security, Xi'an University of Posts and Telecommunications[1]
Xi'an 710121, China
School of Telecommunication Engineering, Xidian University[2]
Xi'an 710071, China
(Email: dxl_xaut@163.com)
(Received Aug. 24, 2019; Revised and Accepted Dec. 6, 2019; First Online Feb. 3, 2020)

Abstract

For two future possible improvements of AES-128: enhanced subkey diffusion property or increased encryption rounds, this paper evaluates the security bound of R-Round AES-128 ($R > 10$) and 10-Round AES-IND-128 (AES-128 with independent of key schedule) against biclique attack. For attacking R-round AES-128($R > 10$), with the increase of several rounds R, the time complexity increases gradually, but it never reaches $2^{127.86}$, reduced by about 10% compared with brute force. For attacking 10-Round AES-IND-128, a 1-round biclique is firstly constructed, and then the attack is proposed. The time complexity is no more than $2^{127.42}$, reduced by about 33% compared with brute force.

Keywords: AES; Biclique; Block Cipher; Cryptanalysis

1 Introduction

Block ciphers are the central tool in the design of protocols for symmetric encryption [16]. Since Rijndael [4] is the winner of Advanced Encryption Standard (AES) in 2000, it has become one of the most widely used Block Ciphers. AES supports 128-bit block size with three different key lengths of 128, 192 and 256 bits, which are denoted as AES-128, AES-192 and AES-256 respectively.

Due to the popularity of AES, cryptology researchers have increased their focus on its security, such as square attack [5, 10], collision attack [12],impossible differential attack [14, 15], meet-in-the-middle attack [6–9], biclique attack [3], integral attacks [18],polytopic attacks [19], subspace trail cryptanalysis [11], structural-differential attacks [13], yoyo tricks [17],Grassi's Attack [1] and so on.

There are many cryptanalytic results on AES-128. In 1997, the 6-round AES-128 was broken with the square attack by the designer of AES [5]. Since 2000, 7-round AES-128 has been broken successively by a series of attacks as follows. The square attack in [10] requires a data complexity of $2^{127.997}$ and a time complexity of 2^{120}. The collision attack in [12]costs a data complexity of 2^{32} and a time complexity of 2^{128}.The impossible differential attacks were proposed in [14,15] and the fastest needs a time complexity of $2^{117.2}$. The meet-in-the-middle attack was proposed in [7–9] and the fastest needs a time complexity of 2^{99}. In 2011, 10 rounds were successfully broken with a data complexity 2^{88} and a time complexity of $2^{126.18}$ when the biclique attack [3] is applied to AES-128. In 2014, the biclique attack [2] was once again improved with a data complexity of 2^{64} and a time complexity of $2^{126.12}$.

The biclique attack [3] can be divided into two steps: biclique exploration from the independent related key differentials, and then key recovery. In the key recovery phase of the attack, the recomputation technique is usually adopted to reduce the time complexity. There are two parts separate components of recomputation: State recomputation and subkey recomputation. In [3], subkey recomputation is not considered because it is negligible compared to state recomputation. This paper considers both the two parts of recomputation: State recomputation and subkey recomputation, this is because the subkey recomputation cannot be ignored for R-Round AES-128($R > 10$) and 10-Round AES-IND-128 anymore.

The principle of a biclique attack is to test all the keys to discover which is the correct key based on the biclique structure. Therefore, the complexity of the biclique attack on the full rounds of any block cipher will never exceed the complexity of the exhaustive key search. If the encryption rounds are increased for AES-128, or if the diffusion of the key schedule is enhanced, how will the complexity of the biclique attack change?

In this paper,we study the biclique attack for R-Round AES-128($R > 10$) and 10-Round AES-IND-128, where AES-IND-128 denotes AES-128 independent of the key schedule.

1) For attacking R-round AES-128 ($R > 10$),with the increasing number of rounds R, the time complexity increases gradually, yet it will never reach $2^{127.86}$. In the attack,both state and subkey recomputation are consided.Prior to performing the attack, properties of the recomputation of R-round subkeys are discov-

ered.

2) For attacking 10-round AES-IND-128,the time complexity is $2^{127.42}$. AES-IND-128 includes the property: the diffusion property is enhanced so that any subkey byte in any round affects all the subkey bytes in the other rounds in term of the difference. When the diffusion of key schedule is enhanced or the number of encryption rounds is increased, biclique attacks remain effective to AES-128.

The remainder of this paper is organized as follows. The AES block cipher is described in Section 2 and the biclique attack is introduced in Section 3. In Sections 4 and 5, we apply the biclique attacks to R-round AES-128 and 10-round AES-IND-128, respectively. Finally in Section 6, the conclusions are drawn from the results.

2 Description of AES-128

AES-128 [4] encrypts or decrypts data blocks of 128 bits by using keys of 128-bits. The number of rounds is 10, denoted as successively. A 128-bit plaintext and the interme-diate state are treated as byte matrices of size. The round transformation of AES consists of the four basic transformations:

SubBytes (SB): Applying the same 8-bit to 8-bit invertible S-box 16 times in parallel on each byte of the state.

ShiftRows(SR): Cyclically shifting each row (the i'th row is shifted by i bytes to the left).

MixColumns (MC): Multiplication of each column by a 4×4 matrix over the field $GF(2^8)$.

AddRoundKey (ARK): XORing the state and a subkey.

The MC operation is omitted in the last round, and an additional ARK operation using a Whitening SubKey (WSK) is performed before the first round.

The Key Schedule(KS) of AES-128 takes the 128-bit user key and transforms it into 11 subkeys of 128-bits each.The subkey array is denoted by $W_0, W_1, \cdots, W_{43}$, where each word of consists of 32 bits. $K = (W_0, W_1, W_2, W_3)$ is the user supplied keys. $K_r = (W_{4(r+1)}, W_{4(r+1)+1}, W_{4(r+1)+2}, W_{4(r+1)+3})(r = 0, \cdots, 9)$ is updated according to the following rule:
If

$$i \equiv 0 \bmod 4$$

then

$$W_i \equiv W_{i-4} \oplus SB(RotByte(W_{i-1})) \oplus Rcon(i/4)$$

else

$$W_i \equiv W_{i-4} \oplus W_{i-1}.$$

K_r denotes the r-round subkey. $K_{-1} = K$ as a WSK. $RotByte(\cdot)$ denotes the rotation of the word by one byte.

In this paper, $\#2r$ and $\#(2r+1)$ are addressed as the states before SB and the state after MC in round r, respectively.

Property 1. *For r-round AES-128, the ratio of one SubBytes operation to the full AES is $\sigma = 4/5r$.*

3 Chosen-Ciphertext Biclique Attack

In this section, we introduce the biclique attack proposed by Bogdanov *et al.* [3]. Before the description of the biclique attack, the biclique structure must be denoted: Let f be subcipher that maps an internal state S to the ciphertext $C : f_K(S) = C$.The 3-tuple $[\{C_i\}, \{S_j\}, \{K[i,j]\}]$ is called a d-dimensional biclique, if $C_i = f_{K[i,j]}(S_j)$ $(, i, j \in \{0, 1, \cdots, 2^d - 1\})$.

The biclique attack can be divided into two steps: biclique exploration and key recovery. The following sections of this paper will describe both biclique exploration and key recovery in further detail.

3.1 Biclique Exploration from the Independent Related-Key Differentials

The d-dimensional biclique can be achieved from the independent related-key differentials. There are two stages: key partition and then biclique construction.

3.1.1 Key Partition

2^n keys can be divided into 2^{n-2d} groups $K^{(m)}(m = 0, 1, \cdots, 2^{n-2d} - 1)$, where $K^{(m)}$ is defined to be the m-th key group with 2^{2d} keys $K^{(m)}[i,j](i, j = (0, 1, \cdots, 2^d - 1))$. These keys can be obtained as follows:

- Look for the key differentials Δ_i^K, ∇_j^K, such that $\Delta_i^K \cap \nabla_j^K = \{0\}$.
- With Δ_i^K, ∇_j^K, determine the base key $K^{(m)}[0,0]$ in $K^{(m)}$.

Therefore: $K^{(m)}[i,j] = K^{(m)}[0,0] \oplus \Delta_i^K \oplus \nabla_j^K$.

3.1.2 Biclique Construction

In each $K^{(m)}$, construct the d-dimension biclique as follows:

- Base computation: $S_0^{(m)} \xrightarrow[f]{K^{(m)}[0,0]} C_0^{(m)}$.
- Based on Δ_i^K, ∇_j^K, construct differentials

$$\Delta_i^{(m)} - differentials : 0 \xrightarrow[f]{\Delta_i^K} \Delta_i^{(m)},$$

$$\nabla_j^{(m)} - differentials : \nabla_j^{(m)} \xrightarrow[f]{\nabla_j^K} 0.$$

Such that they do not share the active nonlinear component, where $\Delta_0^K = 0, \Delta_0^{(m)} = 0, \nabla_0^K = 0, \nabla_0^{(m)} = 0$.

Therefore it is denoted:

$$\begin{aligned} S_j^{(m)} &= S_0^{(m)} \oplus \nabla_j^{(m)} \\ C_i^{(m)} &= C_0^{(m)} \oplus \Delta_i^{(m)} \\ K^{(m)}[i,j] &= K^{(m)}[0,0] \oplus \Delta_i^K \oplus \nabla_j^K \end{aligned}$$

and obtain the definition of a d-dimensional biclique $[\{C_i^{(m)}\}, \{S_j^{(m)}\}, \{K^{(m)}[i,j]\}]$.

3.2 Key Recovery under the Chosen-Ciphertext Attack

3.2.1 Key Recovery

For each $K^{(m)}$, based on the d-dimension biclique $[\{C_i^{(m)}\}, \{S_j^{(m)}\}, \{K^{(m)}[i,j]\}]$, the key recovery is as follows:

- Data Collection: The adversary obtains the 2^d plaintexts from the ciphertexts $P_i^{(m)}$ through the decryption oracle: $C_i^{(m)} \xrightarrow[e^{-1}]{decryption \quad oracle} P_i^{(m)}$.
- Key Testing: The block cipher E can be decomposed into $E : P \underset{h}{\rightarrow} V \underset{g}{\rightarrow} S \underset{f}{\rightarrow} C$.For the testing key $K^{(m)}[i,j]$, the adversary checks whether

  $$P_i^{(m)} \xrightarrow[g]{K^{(m)}[i,j]} \vec{v}^{(m)} \stackrel{?}{=} \overleftarrow{v}^{(m)} \xleftarrow[h]{K^{(m)}[i,j]} S_j^{(m)}$$

  If the equation holds, the testing key is the secret key K_{secret}.

The full time complexity of the attack is:

$$C_{full} = 2^{n-2d} C^{(m)} = 2^{n-2d}[C_{biclique}^{(m)} + C_{match}^{(m)} + C_{falsepos}^{(m)}]$$

where $C_{biclique}^{(m)}$ is the complexity of constructing biclique; $C_{match}^{(m)}$ is the complexity of the computation of the internal variable v 2^d times in each direction; $C_{falsepos}^{(m)}$ is the complexity generated by false positives.

3.2.2 The Recomputation Technique

In fact, in order to decrease the time complexity in the key testing, adopt the recomputation technique. In fact, there are two parts of recomputation as follows.

- State Recomputation.
  - Precomputation: The adversary computes and stores 2×2^d computations: $\forall\ i(P_i^{(m)} \xrightarrow[g]{K^{(m)}[i,0]} \vec{v}^{(m)})$ and $\vec{v}^{(m)} \xleftarrow[h]{K^{(m)}[0,j]} S_j^{(m)}$.
  - Recomputation: For particular i and j, the adversary recomputes the states which differ from the stored ones.
- Subkey Recomputation. First, the adversary computes and stores computations $K^{(m)}[i,0]$ and $K^{(m)}[0,j]$. Then, for other $K^{(m)}[i,j]$, it is recomputed only those parts that differ from the stored ones.

  So the full time complexity of the attack is: $C_{full} = 2^{l-2d} C^{(m)} = 2^{l-2d}[C_{biclique}^{(m)} + C_{precomp}^{(m)} + C_{recomp_1}^{(m)} + C_{recomp_2}^{(m)} + C_{falsepos}^{(m)}]$, where $C_{precomp}^{(m)}$ is the complexity of the precomputation in the key recovery; $C_{recomp_1}^{(m)}$ and $C_{recomp_2}^{(m)}$ are the complexities of the state and the subkey recomputation in the matching stage,respectively.

Remark 1. *Subkey recomputation is ignored in [3],but cannot be ignored on attacks on R-Round AES-128 ($R > 10$) and 10-Round AES-128-IND.*

4 Biclique Attack on R-Round AES-128($R > 10$)

The encryption rounds of AES-128 can be improved to exceed 10 rounds, therefore biclique attacks are investigated for R-Round AES-128($R > 10$). For the purpose of this research a recomputation technique was adopted within section III. With the rounds increased, $C_{recomp_2}^{(m)}$ becomes increasingly very critical and complex, so we firstly calculate the recomputation of the subkeys.Then perform the biclique attack on $R-$Round AES-128($R > 10$).

4.1 Recomputation of the Subkeys

The superscript (m) of $K_{(r)}^{(m)}[i,j]$ is not reflected in the following lemmas, so it is omitted and denoted as $K_{(r)}[i,j]$.

If we know r-round subkeys $K_{(r)}[0,0], \Delta_i^K, \nabla_j^K$ and $K_{(r)}[i,j] = K_{(r)}[0,0] \oplus \Delta_i^K \oplus \nabla_j^K$, evaluate recomputation of the $(r-1)$-round subkeys $K_{(r-1)}[i,j]$ by the following lemmas.

Lemma 1. *If the value of r-round subkeys $K_{(r)}[i,j]$ is known, the evaluatation of the recalculation of the $(r-1)$-round subkeys $K_{(r-1)}[i,j]$ is given as follows.*

1) $K_{(r-1)}$ can be expressed as the linear function of $K_{(r)}$ and $\Delta K_{(r)}$, so the recomputation for $K_{(r-1)}$ can be ignored, where $\Delta K_{(r)} = K_{(r)}[i,j] \oplus K_{(r)}[0,0]$.

2) $K_{(r-1)}$ cannot be expressed as the linear function of $K_{(r)}$ and $\Delta K_{(r)}$, so the recomputation for $K_{(r-1)}$ cannot be ignored.

Proof. $K_{(r),h}(h = 0, 1, \cdots, 15)$ denote h-th byte of $K_{(r)}$. If the subkeys in round r are known, the subkeys in round

$r-1$ can be achieved as follows by the key schedule.

$$\begin{aligned}
K_{(r-1),0} &= K_{(r),0} \oplus S(K_{(r),9} \oplus K_{(r),13}) \\
K_{(r-1),1} &= K_{(r),1} \oplus S(K_{(r),10} \oplus K_{(r),14}) \\
K_{(r-1),2} &= K_{(r),2} \oplus S(K_{(r),11} \oplus K_{(r),15}) \\
K_{(r-1),3} &= K_{(r),3} \oplus S(K_{(r),8} \oplus K_{(r),12}) \\
K_{(r-1),4} &= K_{(r),0} \oplus K_{(r),4} \\
K_{(r-1),5} &= K_{(r),1} \oplus K_{(r),5} \\
K_{(r-1),6} &= K_{(r),2} \oplus K_{(r),6} \\
K_{(r-1),7} &= K_{(r),3} \oplus K_{(r),7} \\
K_{(r-1),8} &= K_{(r),4} \oplus K_{(r),8} \\
K_{(r-1),9} &= K_{(r),5} \oplus K_{(r),9} \\
K_{(r-1),10} &= K_{(r),6} \oplus K_{(r),10} \\
K_{(r-1),11} &= K_{(r),7} \oplus K_{(r),11} \\
K_{(r-1),12} &= K_{(r),8} \oplus K_{(r),12} \\
K_{(r-1),13} &= K_{(r),9} \oplus K_{(r),13} \\
K_{(r-1),14} &= K_{(r),10} \oplus K_{(r),14} \\
K_{(r-1),15} &= K_{(r),11} \oplus K_{(r),15}.
\end{aligned}$$

The equivalent but concise expressions are as follows:

$$\begin{aligned}
K_{(r-1),h} &= K_{(r),h} \oplus S(K_{(r),8+(h+1) \bmod 4} \\
&\quad \oplus K_{(r),12+(h+1) \bmod 4}) \ (h = 0, 1, 2, 3) \\
K_{(r-1),h} &= K_{(r),h-4} \oplus K_{(r),h} \ (h = 4, 5, \cdots, 15).
\end{aligned}$$

Remark 2. *For clarity, Rcon(•) is ignored in the key schedule above as it does not affect the final results.*

If the values of r-round subkeys are known: $K_{(r)}[i,j]$, $\Delta K_{(r)} = K_{(r)}[i,j] \oplus K_{(r)}[0,0]$. There are 2 cases in the recalculation of the $r-1$ round subkeys $K_{(r-1)}[i,j]$.

Case 1: For $0 \le h \le 3$, we consider the computation of $K_{(r-1),0}, K_{(r-1),1}, K_{(r-1),2}, K_{(r-1),3}$. As $\Delta K_{(r)} = K_{(r)}[i,j] \oplus K_{(r)}[0,0]$, we have

$$\begin{aligned}
K_{(r),0}[i,j] &= K_{(r),0}[0,0] \oplus \Delta K_{(r),0} \quad (1) \\
K_{(r),8+(h+1) \bmod 4}[i,j] &= K_{(r),8+(h+1) \bmod 4}[0,0] \\
&\quad \oplus \Delta K_{(r),8+(h+1) \bmod 4} \quad (2) \\
K_{(r),12+(h+1) \bmod 4}[i,j] &= K_{(r),12+(h+1) \bmod 4}[0,0] \\
&\quad \oplus \Delta K_{(r),12+(h+1) \bmod 4} \quad (3) \\
K_{(r-1),h}[0,0] &= K_{(r),h}[0,0] \quad (4) \\
&\quad \oplus S(K_{(r),8+(h+1) \bmod 4}[0,0] \\
&\quad \oplus K_{(r),12+(h+1) \bmod 4}[0,0]) \\
K_{(r-1),h}[i,j] &= K_{(r),h}[i,j] \quad (5) \\
&\quad \oplus S(K_{(r),8+(h+1) \bmod 4}[i,j] \\
&\quad \oplus K_{(r),12+(h+1) \bmod 4}[i,j]).
\end{aligned}$$

By Equations (1)-(5), we have

$$\begin{aligned}
K_{(r-1),h}[i,j] &\overset{(5)}{=} K_{(r),h}[i,j] \\
&\quad \oplus S(K_{(r),8+(h+1) \bmod 4}[i,j] \\
&\quad \oplus K_{(r),12+(h+1) \bmod 4}[i,j]) \\
&\overset{(2)(3)}{=} K_{(r),h}[i,j] \oplus S(K_{(r),8+(h+1) \bmod 4}[0,0] \\
&\quad \oplus \Delta K_{(r),8+(h+1) \bmod 4} \\
&\quad \oplus K_{(r),12+(h+1) \bmod 4}[0,0] \\
&\quad \oplus \Delta K_{(r),12+(h+1) \bmod 4}) \\
&\overset{(1)}{=} K_{(r),h}[0,0] \oplus \Delta K_{(r),h} \\
&\quad \oplus S(K_{(r),8+(h+1) \bmod 4}[0,0] \\
&\quad \oplus \Delta K_{(r),8+(h+1) \bmod 4} \\
&\quad \oplus K_{(r),12+(h+1) \bmod 4}[0,0] \\
&\quad \oplus \Delta K_{(r),12+(h+1) \bmod 4}) \\
&\overset{(4)}{=} K_{(r-1),h}[0,0] \oplus \Delta K_{(r),h} \\
&\quad \oplus S(K_{(r),8+(h+1) \bmod 4}[0,0] \\
&\quad \oplus K_{(r),12+(h+1) \bmod 4}[0,0]) \\
&\quad \oplus S(K_{(r),8+(h+1) \bmod 4}[0,0] \\
&\quad \oplus \Delta K_{(r),8+(h+1) \bmod 4} \\
&\quad \oplus K_{(r),12+(h+1) \bmod 4}[0,0] \\
&\quad \oplus \Delta K_{(r),12+(h+1) \bmod 4}).
\end{aligned}$$

Therefore,

$$\begin{aligned}
K_{(r-1),h}[i,j] &= K_{(r-1),h}[0,0] \oplus \Delta K_{(r),h} \\
&\quad \oplus S(K_{(r),8+(h+1) \bmod 4}[0,0] \\
&\quad \oplus K_{(r),12+(h+1) \bmod 4}[0,0]) \\
&\quad \oplus S(K_{(r),8+(h+1) \bmod 4}[0,0] \\
&\quad \oplus \Delta K_{(r),8+(h+1) \bmod 4} \\
&\quad \oplus K_{(r),12+(h+1) \bmod 4}[0,0] \\
&\quad \oplus \Delta K_{(r),12+(h+1) \bmod 4}). \quad (6)
\end{aligned}$$

Therefore, there are 2 Conditions 1), 2) from Equation (6).

1) $\Delta K_{(r),8+(h+1) \bmod 4} \oplus \Delta K_{(r),12+(h+1) \bmod 4} = 0 \Rightarrow K_{(r-1),h}[i,j] = K_{(r-1),h}[0,0] \oplus \Delta K_{(r),h}$. $K_{(r-1)}[i,j]$ can be expressed as the linear function of $K_{(r-1)}[0,0]$ and $\Delta K_{(r)}$.
2) $\Delta K_{(r),8+(h+1) \bmod 4} \oplus \Delta K_{(r),12+(h+1) \bmod 4} \neq 0 \Rightarrow K_{(r-1),h}[i,j] \neq K_{(r-1),h}[0,0] \oplus \Delta K_{(r),h}$. $K_{(r-1)}[i,j]$ cannot be expressed as the linear function of $K_{(r-1)}[0,0]$ and $\Delta K_{(r)}$.

Case 2: For $4 \le h \le 15$,we consider the computation of $K_{(r-1),4}, K_{(r-1),5}, \cdots K_{(r-1),15}$. Because $\Delta K_{(r)} = K_{(r)}[i,j] \oplus K_{(r)}[0,0]$, we have

$$\begin{aligned}
K_{(r),h-4}[i,j] &= K_{(r),h-4}[0,0] \oplus \Delta K_{(r),h-4} \quad (7) \\
K_{(r),h}[i,j] &= K_{(r),h}[0,0] \oplus \Delta K_{(r),h}. \quad (8)
\end{aligned}$$

By the schedule above when $h = 4, \cdots, 15$, the equations are:

$$\begin{aligned}
K_{(r-1),h}[i,j] &= K_{(r),h-4}[i,j] \oplus K_{(r),h}[i,j] \quad (9) \\
K_{(r-1),h}[0,0] &= K_{(r),h-4}[0,0] \oplus K_{(r),h}[0,0]. \\
&\quad (10)
\end{aligned}$$

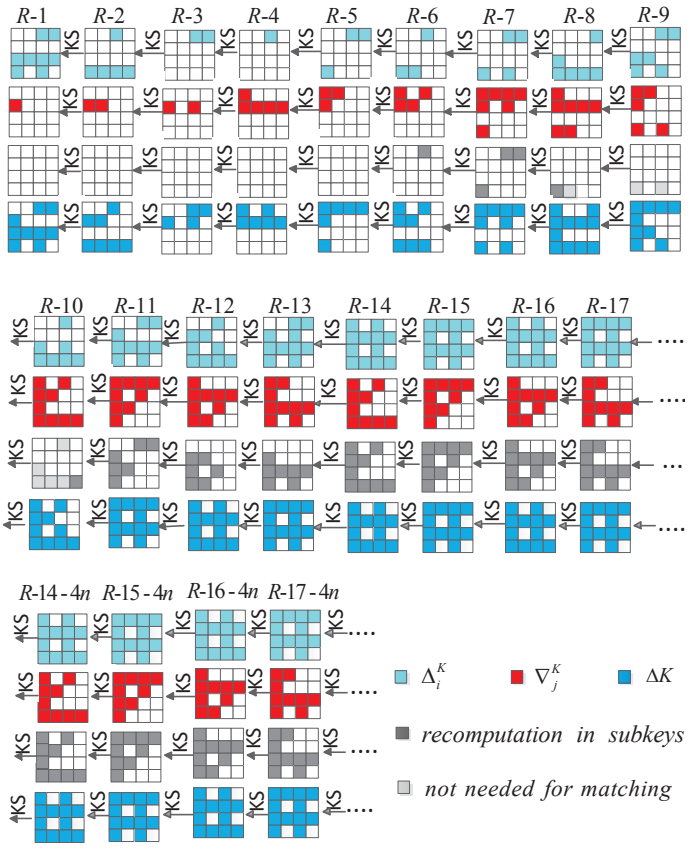

Figure 1: Computation in the subkeys form round -1 to round R-1

By Equations (7)-(10), we have:

$$\begin{aligned} K_{(r-1),h}[i,j] &\overset{(9)}{=} K_{(r),h-4}[i,j] \oplus K_{(r),h}[i,j] \\ &\overset{(7)(8)}{=} K_{(r),h-4}[0,0] \oplus \Delta K_{(r),h-4} \\ &\quad \oplus K_{(r),h}[0,0] \oplus \Delta K_{(r),h} \\ &\overset{(10)}{=} K_{(r-1),h-4}[0,0] \oplus \Delta K_{(r),h-4} \\ &\quad \oplus \Delta K_{(r),h}. \end{aligned} \tag{11}$$

From Equation (11), there is

$$\begin{aligned} K_{(r-1),h}[i,j] &= K_{(r-1),h-4}[0,0] \oplus \Delta K_{(r),h-4} \\ &\quad \oplus \Delta K_{(r),h}. \end{aligned}$$

So $K_{(r-1)}[i,j]$ can be expressed as the linear function of $K_{(r-1)}[0,0]$ and $\Delta K_{(r)}$.

□

Corollary 1. *The column 1,2,3 of $(r-1)$-round subkeys $K_{(r-1)}[i,j]$ are linear expressions of r-round subkeys $K_{(r-1)}[0,0]$ and $\Delta K_{(r)}$.*

Proof. Based on Lemma 1 in Equation (11). □

Corollary 2. *Only column 0 of $(r-1)$-round subkeys is a nonlinear expression of r-round subkeys $K_{(r-1)}[0,0], K_{(r)}[0,0]$ and $\Delta K_{(r)}$.*

Proof. Based on Lemma 1 in Equation (6). □

Corollary 3. *With the subkey recomputation, for $r \in \{-1, 0, \cdots, R-1\})$, if we know $K_{(r)}[0,0], \Delta K_{(r)}$ and $K_{(r-1)}[0,0]$, the recomputation for $(r-1)$-round subkeys $K_{(r-1)}[i,j]$, only appears in byte $0,1,2,3$, and they are only related to the last two columns of r-round subkeys $K_{(r)}[0,0]$ and $\Delta K_{(r)}$.*

Proof. By Corollary 2 and Lemma 1 in Equation (6). □

Lemma 2. *If*

$$\Delta_i^K = \begin{bmatrix} 0 & 0 & i & i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \nabla_j^K = \begin{bmatrix} 0 & 0 & 0 & 0 \\ j & 0 & j & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$K^{(m)}[0,0] \in \begin{bmatrix} N & N & N & 0 \\ 0 & N & N & N \\ N & N & N & N \\ N & N & N & N \end{bmatrix}$$

in round $R-3$, computation $K^{(m)}[i,j]$ in the subkeys form round -1 to round $R-1$ is depicted in Figure 1 ($i,j \in GF(2^8)$, N denotes non-zero byte and 0 denotes zero byte).

Proof. The first, the second and fourth lines of Figure 1 above can be derived by the computer experiments based on the key schedule. The third line of Figure 1 can be derived by Lemma 1 and Corollarys 1, 2 and 3.

1) In the first line of Figure 1, Δ_i^K is depicted, and $K^{(m)}[i,0] = K^{(m)}[0,0] \oplus \Delta_i^K$.

2) In the second line of Figure 1, ∇_j^K is depicted, and $K^{(m)}[0,j] = K^{(m)}[0,0] \oplus \nabla_j^K$.

3) In the third line of Figure 1, the recomputation in the subkeys of $K^{(m)}[i,j]$ is depicted, where white cells need no recomputation because they are only related to Δ_i^K or ∇_j^K; Dark gray cells need recomputa-tion based on the Lemma 1 and Corollarys 1, 2 and 3; light gray cells are not required for matching.

4) In the fourth line of Figure 1, $\Delta K = K^{(m)}[i,j] \oplus K^{(m)}[0,0]$ is depicted based on the first line and the second line, where white cells denote zero difference.

□

Remark 3. *Our key partition in Round $R-3$ is the same as key partition in Round 8 [3], the computation of the subkeys in Figure 1 from $R-11$ to R is the same as Figure 9 [3] from -1 to 10.*

Lemma 3. *In Figure 1, the current equations are:*

1) $Pos_u(1) = Pos_{u-4v}(1)(u \le R-14, v = 0,1,\cdots)$ in Δ_i^K and ∇_j^K,where $Pos_u(1)$ denotes the position of non-zero differential in Round u.

2) $Pos_u(2) = Pos_{u-4v}(2)$ in recomputation of subkeys in Round u,where $Pos_u(2)$ denotes the position of recomputation in Round u.

Proof.

1) In the first, second and the fourth lines of Figure 1,it can be seen, the regularity from Round $R-4$ to Round -1, the position of non-zero differential byte in Δ_i^K, ∇_j^K and ΔK repeats every 4 rounds.

2) Similarly, in the third line of Figure 1, it can be shown (2) holds.

□

Theorem 1. *With the subkey recomputation technique, for $r \in \{-1,0,\cdots,R-1\}$), if we know r-round subkeys $K_{(r)}[0,0], \Delta K_{(r)}$, $K_{(r)}[i,j] = K_{(r)}[0,0] \oplus \Delta K_{(r)}$ and $(r-1)$-round subkeys $K_{(r-1)}[0,0]$, there are $(2R-16)$ S-boxes to be recomputed at most for the $(r-1)$-round subkeys $K_{(r-1)}[i,j]$ (Figure 1).*

Proof. We assume that we know r-round subkeys $K_{(r)}[0,0], \Delta K_{(r)}$ and $K_{(r)}[i,j] = K_{(r)}[0,0] \oplus \Delta K_{(r)}$.

When r is from round $R-1$ to round $R-8$,it is obviously there are all 2 S-boxes required to be recomputed for $K_{(r-1)}[i,j]$.

When $r \in \{-1,\cdots,R-9\}$, in the fourth line in Figure 1 based on Lemmas 2 and 3.

$$\Delta K_{r-1} = \begin{pmatrix} N & 0 & N & 0 \\ N & N & N & N \\ N & 0 & N & 0 \\ N & N & N & N \end{pmatrix} \text{ or } \begin{pmatrix} N & N & N & N \\ N & 0 & N & 0 \\ N & N & N & N \\ N & 0 & N & 0 \end{pmatrix}$$

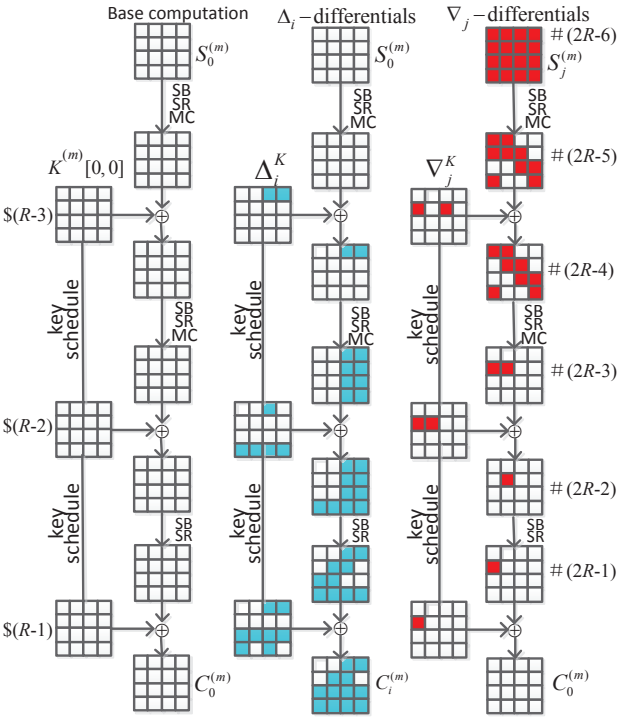

Figure 2: Biclique for R-round AES-128(R >10) [3]

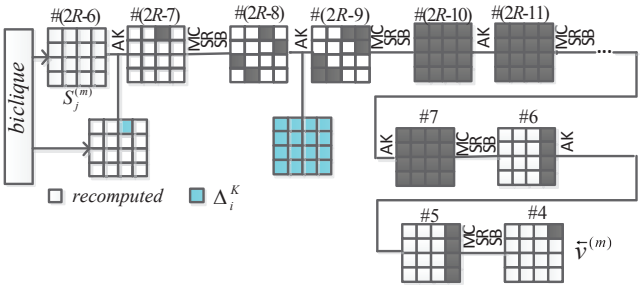

Figure 3: Recomputation in the forward direction for R-round AES-128(R>10)

where N denotes non-zero byte and 0 denotes zero byte. Then by the key schedule,

$$\Delta K_r = \begin{pmatrix} N & N & D_1 & D_2 \\ N & N & N & N \\ N & N & D_3 & D_4 \\ N & N & N & N \end{pmatrix} \text{ or } \begin{pmatrix} N & N & N & N \\ N & N & D_5 & D_6 \\ N & N & N & N \\ N & N & D_7 & D_8 \end{pmatrix}$$

It can therefore be demonstrated that: $D_1 = D_2, D_3 = D_4, D_5 = D_6, D_7 = D_8$. Based on Corollary 2 (*i.e.* Lemma 1 in Equation (4.6)), 2 S-boxes required to be recomputed at most in each round r.

So, there are $2(R-9)+2$ S-boxes be recomput-uted at most from round -1 to $R-1$ for $K_{(r-1)}[i,j]$.

Therefore, for the subkeys $K_r^{(m)}[i,j](r \in \{-1,0,\cdots,R-1\})$, there are $2R-16$ S-boxes to be recomputed at most. □

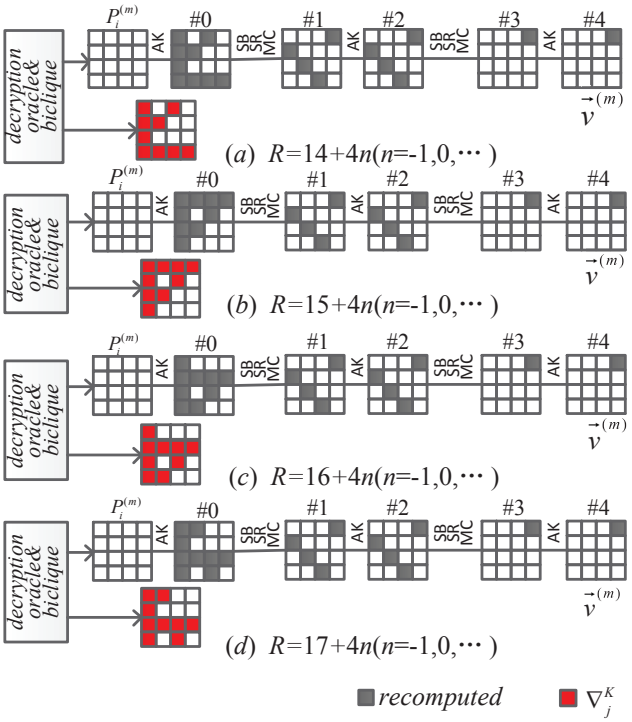

Figure 4: Recomputation in the forward direction for R-round AES-128(R>10)

4.2 Biclique Attack

Theorem 2. *For attacking R rounds of AES-128($R \geq 10$), the complexity of biclique attack*

$$C_{full} \leq 2^{121}(-\frac{752.4}{R} + 2^{6.86}).$$

Proof. For R-Round AES-128, biclique attack can be described in the following two steps: 3-round biclique construction and key recovery.

1) 3-round biclique construction. The 3-round biclique of dimension 8 $[\{C_i^{(m)}\}, \{S_j^{(m)}\}, \{K^{(m)}[i,j]\}]$ is constructed for AES-128 as shown in Figure 2,similar with Figure 4 [3].

2) Key recovery. We recover the secret key with recomputation technique.

Precompute $P_i^{(m)} \xrightarrow[g]{K^{(m)}[i,0]} \vec{v}_i^{(m)}$ and $\vec{v}_j^{(m)} \xleftarrow[h]{K^{(m)}[0,j]} S_j^{(m)}$. Then, store intermediate states and subkeys, where $\vec{v}_i^{(m)}$ and $\overleftarrow{v}_j^{(m)}$ are state #4.

- The amount of state recomputation in both directions is evaluated, where $S_j^{(m)}$ is the input of the round $R-3$.

  Backward direction: Recompute $\overleftarrow{v}^{(m)} \xleftarrow[h]{K^{(m)}[i,j]} S_j^{(m)}$ which is different from $\overleftarrow{v}_j^{(m)} \xleftarrow[h]{K^{(m)}[0,j]} S_j^{(m)}$. As shown in Figure 3, $41 + 16(R-10)$ S-boxes should be recomputed.

  Forward direction: Recompute $P_i^{(m)} \xrightarrow[g]{K^{(m)}[i,j]} \vec{v}^{(m)}$ which is different from $P_i^{(m)} \xrightarrow[g]{K^{(m)}[i,0]} \vec{v}_i^{(m)}$, therefore at most 13 S-boxes should be recomputed because the whitening subkeys of $K^{(m)}[i,j]$ and $K^{(m)}[i,0]$ differ in at most 9 bytes as shown in the line 2 of Figure 1. As shown in Figure4, for whitening subkeysK_{-1}, there are 4 kinds of difference between $K^{(m)}[i,j]$ and $K^{(m)}[i,0]$ when $R = 14+4n, R = 15+4n, R = 16+4n, R = 17+4n(n = -1, 0, \cdots)$.At most 13 S-boxes should be recomputed because the whitening subkeys of $K^{(m)}[i,j]$ and $K^{(m)}[i,0]$ differ in at most 9 bytes in four cases.

- The amount of subkey recomputation is evaluated.

  With the round increased, the recomputation of the subkeys cannot be ignored. First, the adversary computes and stores computations $K^{(m)}[i,0]$ and $K^{(m)}[0,j]$. Then, for other $K^{(m)}[i,j]$, he recomputes only those parts that differ from the stored ones.

  According to the Theorem 1, for the subkeys $K_r^{(m)}[i,j](r \in \{-1, 0, \cdots, R-1\})$, there are $2R-16$ S-boxes to be recomputed at most.

  In summary, for each $K^{(m)}$, $41 + 16(R-10) + 9 + (2R-16) = 18R-126$ S-box should be recomputed at most. The full time complexity of attacking R-round AES-128 is

$$C_{full} \leq 2^{112}\{2^7+2^7+2^{16}\times[18R-126]\times\frac{1}{16}\times\frac{4}{5R}+2^8\}$$
$$= 2^{121}(-\tfrac{752.4}{R} + 2^{6.86}) < 2^{127.86}$$

Theorem 2 shows: For attacking R rounds of AES-128 ($R \geq 10$), with the increase of number of rounds R, the complexity of biclique attack increases gradually, but it can never reach $2^{127.86}$. □

5 Biclique Attack on 10-Round AES-IND-128

AES-IND-128 denotes AES-128 independent of the key schedule. The AES key scheme can also be improved, so we need consider the AES-IND-128. In this paper, we assume the key schedule of AES-IND-128 satisfies: The diffusion property is enhanced so that any subkey byte in any round affects all the subkey bytes in the other rounds in terms of the difference.

Theorem 3. *The time complexity of the biclique attack on 10-round AES-IND-128 is* $2^{127.42}$.

Proof. For 10-round AES-IND-128, the biclique attack can be described in the following two steps: biclique construction and key recovery.

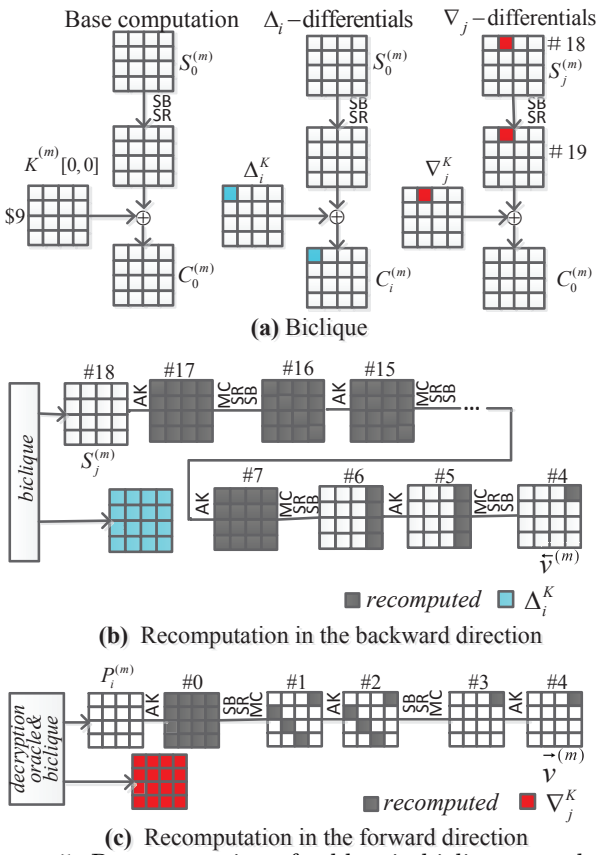

Figure 5: Recomputation of subkey in biclique attacks on 10-round AES-128-IND

1) 1-Round Biclique of Dimension 8. For 10-round AES-IND-128, it may fail to construct the 3-round or even 2-round biclique due to the diffusion properties of the internal rounds. Fortunately, it is feasible to construct 1-round biclique. In fact, what we should do is to find $\Delta_i^K \cap \Delta_j^K = \{0\}$ in round-9 subkey space.

- Key Partition. With the strategy of the key partition inside the biclique, define the key groups with respect to round-9 subkey space and enumerate the 2^{112} groups of 2^{16} keys: $K^{(m)}[i,j] = K^{(m)}[0,0] \oplus \Delta_i^K \oplus \nabla_j^K$ $(i,j = 0,1,\cdots,2^8-1; m = 0,1,\cdots 2^{112}-1)$, where

$$\Delta_i^K = \begin{bmatrix} i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\nabla_j^K = \begin{bmatrix} 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$K^{(m)}[0,0] \in \begin{bmatrix} 0 & 0 & N & N \\ N & N & N & N \\ N & N & N & N \\ N & N & N & N \end{bmatrix}$$

- 1-Round Biclique Construction. For each $K^{(m)}$, construct the biclique (Figure 5(a)). Fix $C_0^{(m)} = 0$ and computes $S_0^{(m)} = f^{-1}_{K^{(m)}[0,0]}(C_0^{(m)})$. Then, construct $\Delta_i^{(m)}$−differentials and $\nabla_j^{(m)}$−differentials. Finally, get the 1-round biclique of dimension 8.

  Because Δ_i-differentials influence the 1 bytes of the ciphertext, all the ciphertexts share the same values except bytes C_0.Therefore, the data complexity does not exceed 2^8.

2) Key Recovery.

- The amount of state recomputation in both directions is evaluated.

  Backward Direction: Because of the high diffusion of the AES-IND-128 key schedule, the subkeys 9 of $K^{(m)}[i,j]$ and $K^{(m)}[0,j]$ differ in all 16 bytes. As shown in Figure 5(b), 85 S-boxes should be recomputed.

  Forward Direction: The Whitening Subkey of $K^{(m)}[i,j]$ and $K^{(m)}[i,0]$ differ in all 16 bytes. As demonstrated in Figure 5(c), 20 S-boxes should be recomputed.

- The amount of subkey recomputation is evaluated.

  The diffusion of key schedule is enhanced: Any subkey byte in any round affects all the subkey bytes in the other rounds in terms of the difference. In the following, it is assumed that the nonlinear transform of the subkey generation does not change, $W_i \equiv W_{i-4} \oplus SB(RotByte(W_{i-1})) \oplus Rcon(i/4)(i \equiv 0 \bmod 4)$, S-boxes in subkey recomputation occurs in the first column of each round.

  Recomputations of the subkey in each round is located within TABLE 1.It is evident from the data within Table 1 that the following is true:

  Backward Direction: 4 S-boxes should be recomputed in round 3,4,5,6,7,8, respectively.

  Forward Direction: There are 4 S-boxes, 1S-box should be recomputed in round -1,0 respectively.

  So, for the subkeys $K_r^{(m)}[i,j](r \in \{-1,0,\cdots,R-1\})$, when Δ_i^K, ∇_j^K are computed,there are 29 S-boxes to be recomputed at most. Complexities: For each $K^{(m)}$,105+29=134 S-box should be recomputed, so

$$C_{recomp}^{(m)} = 2^{16} \times 134 \times 16^{-1} \times 12.5^{-1} \approx 2^{15.42}$$

  The time complexity of attacking10-round AES-IND-128 is

$$C_{full} = 2^{112}(2^7 + 2^7 + 2^{15.07} + 2^8) = 2^{127.42}$$

□

Theorem 3 shows: It can be demonstrated that although the emphasis was focused on the improvement of the AES-128 key schedule, the full time complexity of biclique attack on 10-round AES-128 cannot exceed $2^{127.42}$.

Table 1: Recomputation of subkey in biclique attacks on 10-round AES-128-IND

Round	Recomputation in subkey
-1	4
0	1
1	0
2	0
3	4
4	4
5	4
6	4
7	4
8	4

6 Conclusions

In this paper, the application of chosen ciphertext biclique attacks to AES-128 have been performed. Attacks on 10-Round AES-IND-128 and R-Round AES-128 ($R > 10$) are considered. 10-Round AES-IND-128 and R-Round AES-128 ($R > 10$) are more secure than 10-Round AES-IND-128 and 10-Round AES-128 in terms of the biclique attacks. Yet, it is evident that when the diffusion of key schedule is enhanced or the number of encryption rounds is increased, the biclique attacks remain effective to AES-128. So in order to make the biclique attack approximate exhaustive key attack in theory, we need not only enhance the diffusion of key schedule, but also increase the number of encryption rounds.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No.61772418), Natural Science Basic Research Plan in Shaanxi Province of China (No.2017JQ6010) and National Cryptography Development Fund(No.MMJJ20180219)

References

[1] A. Bar-On, O. Dunkelman, N. Keller, E. Ronen, A. Shamir, "Improved key recovery attacks on reduced-round AES with practical data and memory complexities," in *Advances in Cryptology*, pp 187-212, 2018.

[2] A. Bogdanov, D. Chang, M. Ghosh, S. K. Sanadhya, "Bicliques with minimal data and time complexity for AES," in *International Conference on Information Security and Cryptology*, pp. 160-174, 2014.

[3] A. Bogdanov, D. Khovratovich, C. Rechberger, "Biclique cryptanalysis of the full AES," in *Advances in Cryptology*, pp. 344-371, 2011.

[4] J. Daemen, V. Rijmen, "The design of Rijndael," *Information Security and Cryptography*, 2002. (`https://autonome-antifa.org/IMG/pdf/Rijndael.pdf`)

[5] J. Daemen, L. Knudsen, V. Rijmen, "The block cipher SQUARE," in *International Workshop on Fast Software Encryption*, pp. 149-165, 1997.

[6] H. Demiric, A. Selcuk, "A meet in the middle attack on 8-round AES," in *International Workshop on Fast Software Encryption*, pp.116-126, 2008.

[7] P. Derbez, P. A. Fouque, J. Jean, "Improved key recovery attacks on reduced-round AES in the single-key setting," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 371-387, 2013.

[8] P. Derbez, P. A. Fouque, "Exhausting demirci-seluk meet-in-the-middle attacks against reduced-round AES," in *Computer Science*, pp. 541-560, 2013.

[9] O. Dunkelman, N. Keller, A. Shamir, "Improved single-key attacks on 8-round AES," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 158-176, 2010.

[10] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, D. Whiting, "Improved cryptanalysis of Rijndael," in *International Workshop on Fast Software Encryption*, pp. 213-230, 2002.

[11] L. Grassi, C. Rechberger, S. Ronjom, "Subspace trail cryptanalysis and its applications to AES," in *IACR Transactions on Symmetric Cryptology*, vol. 2, pp. 192-225, 2017.

[12] H. Gilbert, M. Minier, "A collision attack on 7 rounds of Rijndael," in *AES Candidate Conference*, 2000. (`https://pdfs.semanticscholar.org/7405/c463a0d8477396c3a60408fb3ead0917bfb4.pdf`)

[13] L. Grassi, C. Rechberger, S. Ronjom,"A new structural-differential property of 5-round AES," in *Advances in Cryptology*, pp.289-317, 2017.

[14] J. Lu, O. Dunkelman, N. Keller, J. Kim, "New impossible differential attacks on AES," in *International Conference on Cryptology in India*, pp. 279-293, 2008.

[15] H. Mala, M. Dakhilalian, V. Rijmen, M. Modarres-Hashemi, "Improved impossible differential cryptanalysis of 7-round AES-128," in *International Conference on Cryptology in India*, pp. 282-291, 2010.

[16] A. Mirsaid, T. Gulom, "The encryption algorithm AES-RFWKIDEA32-1 based on network RFWKIDEA32-1," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 1-11, 2016.

[17] S. Ronjom, N. Bardeh, T. Helleseth, "Yoyo tricks with AES," in *Advances in Cryptology*, pp. 217-243, 2017.

[18] B. Sun, M. Liu, J. Guo, L. Qu, V. Rijmen, "New insights on AES-like SPN ciphers," in *Advances in Cryptology*, pp.605-624, 2016.

[19] T. Tiessen, "Polytopic cryptanalysis," in *Advancesin Cryptology*, pp. 214-239, 2016.

Biography

Dong Xiaoli is now a lecturer at Xian university of posts and telecommunications,China. She received the B.S. degree and the M.S. degree in mathematics major from Xi'an University of Technology in 2005 and 2008, and the Ph.D. degrees in cryptography science from Xidian University in 2011, respectively. Her main research fields include block cipher in cryptography, applied mathematics, and optimization algorithm in digital processing.

Chen Jie is now an associate professor at School of Telecommunication Engineering, Xidian University,China. She received the M.S. and Ph.D.degrees in cryptology from Xidian University in2005 and 2007, respectively. Her research fields include cryptography and network security.