1-1-2008

A five-round algebraic property of the advanced encryption standard

Jianyong Huang
University of Wollongong, jyh33@uow.edu.au

Jennifer Seberry
University of Wollongong, jennie@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

A five-round algebraic property of the advanced encryption standard

Abstract

This paper presents a five-round algebraic property of the Advanced Encryption Standard (AES). In the proposed property, we modify twenty bytes from five intermediate values at some fixed locations in five consecutive rounds, and we show that after five rounds of operations, such modifications do not change the intermediate result and finally still produce the same ciphertext. We introduce an algorithm named δ, and the algorithm accepts a plaintext and a key as two inputs and outputs twenty bytes, which are used in the five-round property. We demonstrate that the δ algorithm has 20 variants for AES-128, 28 variants for AES-192 and 36 variants for AES-256. By employing the δ algorithm, we define a modified version of the AES algorithm, the δAES. The δAES calls the δ algorithm to generate twenty bytes, and uses these twenty bytes to modify the AES round keys. The δAES employs the same key scheduling algorithm, constants and round function as the AES. For a plaintext and a key, the AES and the δAES produce the same ciphertext.

Keywords

five, round, algebraic, property, advanced, encryption, standard

Disciplines

Physical Sciences and Mathematics

Publication Details

Huang, J., Seberry, J. R. & Susilo, W. (2008). A five-round algebraic property of the advanced encryption standard. In T. Wu, C. Lei, V. Rijmen & D. Lee (Eds.), Information Security Conference (pp. 316-330). Germany: Springer-Verlag.

A Five-Round Algebraic Property of the Advanced Encryption Standard

Jianyong Huang, Jennifer Seberry and Willy Susilo

Centre for Computer and Information Security Research (CCISR)
School of Computer Science and Software Engineering
University of Wollongong, Australia
{jyh33, jennie, wsusilo}@uow.edu.au

Abstract. This paper presents a five-round algebraic property of the Advanced Encryption Standard (AES). In the proposed property, we modify twenty bytes from five intermediate values at some fixed locations in five consecutive rounds, and we show that after five rounds of operations, such modifications do not change the intermediate result and finally still produce the same ciphertext. We introduce an algorithm named δ, and the algorithm accepts a plaintext and a key as two inputs and outputs twenty bytes, which are used in the five-round property. We demonstrate that the δ algorithm has 20 variants for AES-128, 28 variants for AES-192 and 36 variants for AES-256. By employing the δ algorithm, we define a modified version of the AES algorithm, the δAES. The δAES calls the δ algorithm to generate twenty bytes, and uses these twenty bytes to modify the AES round keys. The δAES employs the same key scheduling algorithm, constants and round function as the AES. For a plaintext and a key, the AES and the δAES produce the same ciphertext.

Keywords: AES, A Five-Round Algebraic Property of the AES, Algorithm δ, Variants of Algorithm δ, Linear Equations, δAES.

1 Introduction

The block cipher Rijndael [1] was selected as the Advanced Encryption Standard (AES) by National Institute of Standards and Technology. Rijndael has a simple and elegant structure, and it was designed carefully to withstand two well-known cryptanalytic attacks: differential cryptanalysis [2] and linear cryptanalysis [3]. Most operations of Rijndael are based on the algebraic Galois field $GF(2^8)$, which can be implemented efficiently in dedicated hardware and in software on a wide range of processors.

Since Rijndael was adopted as a standard [4], there have been many research efforts aiming to evaluate the security of this cipher. A block cipher, named Big Encryption System (BES), was defined in [5], and Rijndael can be embedded into BES. The eXtended Linearization (XL) [6] and the eXtended Sparse Linearization (XSL) [7] techniques are new methods to solve nonlinear algebraic equations. The concept of dual ciphers was introduced in [8], and a collision attack on 7 rounds of Rijndael was proposed in [9]. The most effective attacks on

reduced-round variants of the AES are Square attack which was used to attack the cipher Square [10, 1]. The idea of the Square attack was later employed to improve the cryptanalysis of Rijndael [11], and to attack seven rounds of Rijndael under 192-bit and 256-bit keys [12]. A multiplicative masking method of AES was proposed in [13] and further discussed in [14]. The design of an AES-based stream cipher LEX was described in [15], and the construction of an AES-based message authentication code can be found in [16]. So far, no short-cut attack against the full-round AES has been found.

In this paper, we present a five round property of the AES. We modify twenty bytes from five intermediate values at some fixed locations in five consecutive rounds, and we demonstrate that after five rounds of operations, such modifications do not change the intermediate result and finally still produce the same ciphertext. We introduce an algorithm named δ, and the δ algorithm takes a plaintext and a key as two inputs and outputs twenty bytes, which are used in the five-round property. By employing the δ algorithm, we define a modified version of the AES algorithm, the δAES. The δAES calls the δ algorithm to generate twenty bytes, and uses these twenty bytes to modify the AES round keys. For a plaintext and a key, the AES and the δAES produce the same ciphertext.

This paper is organized as follows: Section 2 provides a short description of the AES. In Section 3, we present the five-round algebraic property of the AES, and introduce the δ algorithm. In Section 4, we define a modified version the AES algorithm, the δAES. Finally, Section 5 concludes this paper. Appendix A and Appendix B provide the process of finding the values of the eight variables which are used in Section 3.

2 Description of the AES

We provide a brief description of the AES, and refer the reader to [4] for a complete description of this cipher. AES is a block cipher with a 128-bit block length and supports key lengths of 128, 192 or 256 bits. For encryption, the input is a plaintext block and a key, and the output is a ciphertext block. The plaintext is first copied to 4×4 array of bytes, which is called the *state*. The bytes of a state is organized in the following format:

a_0	a_4	a_8	a_{12}
a_1	a_5	a_9	a_{13}
a_2	a_6	a_{10}	a_{14}
a_3	a_7	a_{11}	a_{15}

.

where a_i denote the i-th byte of the block. After an initial round key addition, the state array is transformed by performing a round function 10, 12, or 14 times (for 128-bit, 192-bit or 256-bit keys respectively), and the final state is the ciphertext. We denote the AES with 128-bit keys by AES-128, with 192-bit keys by AES-192, and with 256-bit keys by AES-256. Each round of AES consists of the following four transformations (the final round does not include MixColumns):

1. The SubBytes (SB) transformation. It is a non-linear byte substitution that operates independently on each byte of the state using a substitution table.
2. The ShiftRows (SR) transformation. The bytes of the state are cyclically shifted over different numbers of bytes. Row 0 is unchanged, and Row i is shifted to the left i byte cyclicly, $i \in \{1, 2, 3\}$.
3. The MixColumns (MC) transformation. It operates on the state column-by-column, considering each column as a four-term polynomial. The columns are treated as polynomials over $GF(2^8)$ and multiplied modulo $x^4 + 1$ with a fixed polynomial, written as $\{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$.
4. The AddRoundKey (ARK) transformation. A round key is added to the state by a simple bitwise exclusive-or (XOR) operation.

The key expansion of the AES generates a total of $Nb(Nr + 1)$ words: the algorithm needs an initial set of Nb words, and each of the Nr rounds requires Nb words of key data, where Nb is 4, and Nr is set to 10, 12, or 14 for 128-bit, 192-bit, or 256-bit key sizes respectively. For a 128-bit key K, we denote the round keys by

$$\begin{array}{|c|c|c|c|}\hline K_0^i & K_4^i & K_8^i & K_{12}^i \\ \hline K_1^i & K_5^i & K_9^i & K_{13}^i \\ \hline K_2^i & K_6^i & K_{10}^i & K_{14}^i \\ \hline K_3^i & K_7^i & K_{11}^i & K_{15}^i \\ \hline \end{array},$$

where i is the round number, $i \in \{1, 2, \cdots 10\}$. We note that the round key used in the initial round is the secret key K itself, and the secret key is represented without the superscript i.

3 A Five-Round Property of AES

We present a five-round property of the AES in this section. In the proposed property, we modify twenty bytes from five intermediate values at some fixed locations in five consecutive rounds, and we show that after five rounds of operations, such modifications do not change the intermediate result and finally still produce the same ciphertext. The modifications are carried out by performing four extra XOR operations at the end of each round (i.e., after the ARK transformation), and in total, we perform twenty extra XOR operations in five rounds. We require that each of these five rounds must contain SB, SR, MC and ARK transformations.

We use Figure 1 and Figure 2 to describe this property. The layout of the twenty bytes in the five intermediate values is shown in Figure 2, and the twenty bytes are G'_0, G'_2, G'_8, G'_{10}, M'_0, M'_2, M'_8, M'_{10}, R'_0, R'_2, R'_8, R'_{10}, V'_0, V'_2, V'_8, V'_{10}, Z'_0, Z'_2, Z'_8, and Z'_{10}. In Figure 1, all intermediate values are listed when using the AES algorithm to encrypt a plaintext P under a 128-bit key K , and all bytes of the intermediate values are denoted by plain variables. Correspondingly, Figure 2 enumerates all intermediate values of the AES with 20 extra XOR operations. The twenty-byte modifications take place in Round 1, 2, 3, 4 and 5, and after

ARK transformation in each of these five rounds, we perform XOR operations on Bytes 0, 2, 8 and 10. We show that the twenty-byte modifications do not change the input to Round 6, i.e., both the AES and the AES with 20 extra XOR operations generate the same input to Round 6. In Figure 2, a variable marked by a asterisk indicates that the value at that location has been affected by the twenty-byte modifications, and a plain variable shows that the value at that location is not affected by the twenty-byte modifications. For example, after ARK in Round 1 in Figure 2, Byte G_i is XORed with Byte G'_i, and after SB, we have four modified bytes H^*_i, $i \in \{0, 2, 8, 10\}$, and twelve unchanged bytes: H_1, H_3, H_4, H_5, H_6, H_7, H_9, H_{11}, H_{12}, H_{13}, H_{14}, and H_{15}.

3.1 The δ Algorithm

To decide the values of the twenty bytes: G'_i, M'_i, R'_i, V'_i and Z'_i, $i \in \{0, 2, 8, 10\}$, we introduce an algorithm named δ. For any plaintext P and any key K used in the AES algorithm, the δ algorithm accepts P and K as two inputs, and generates an output which contains twenty bytes $\{G'_i, M'_i, R'_i, V'_i, Z'_i\}$, where G'_i, M'_i, R'_i, V'_i, and Z'_i are bytes, $i \in \{0, 2, 8, 10\}$.

The δ algorithm includes a number of steps:

1. Process the first five rounds of the AES algorithm by taking the plaintext P and the key K as the inputs, i.e., start with the initial round, and process Round 1, 2, 3, 4 and 5 of the AES. Therefore, we know all intermediate values in Figure 1, from initial round to Round 5.
2. Initialize G'_i, M'_i, R'_i, V'_i and Z'_i to zero, $i \in \{0, 2, 8, 10\}$.
3. Choose G'_0, G'_2, G'_8 and G'_{10} freely. The only requirement is that at least one of these four bytes is not equal to zero, namely, G'_0, G'_2, G'_8 and G'_{10} cannot be all zeros. If G'_0, G'_2, G'_8 and G'_{10} are all zeros, the δ algorithm outputs twenty zero bytes. Once G'_0, G'_2, G'_8 and G'_{10} are decided, the remaining 16 bytes will be computed by the procedures described in Section 3.1.1, Appendix A, Appendix B and Section 3.1.2.
4. Decide M'_0, M'_2, M'_8 and M'_{10}.
5. Decide R'_0, R'_2, R'_8 and R'_{10}.
6. Decide V'_0, V'_2, V'_8 and V'_{10}.
7. Decide Z'_0, Z'_2, Z'_8 and Z'_{10}.

Remark 1. There are $2^{32} - 1$ combinations of $\{G'_0, G'_2, G'_8, G'_{10}\}$ because each byte can have 2^8 possible values.

3.1.1 Deciding M'_0, M'_2, M'_8 and M'_{10} After we have decided the values of G'_0, G'_2, G'_8 and G'_{10}, we carry out a four-round computation (of the AES with extra 12 XOR operations), called Routine Computation One, which starts with the initial round and ends with MC in Round 4 (see Figure 2). All intermediate values from the computation of this time are stored in array called Buffer One

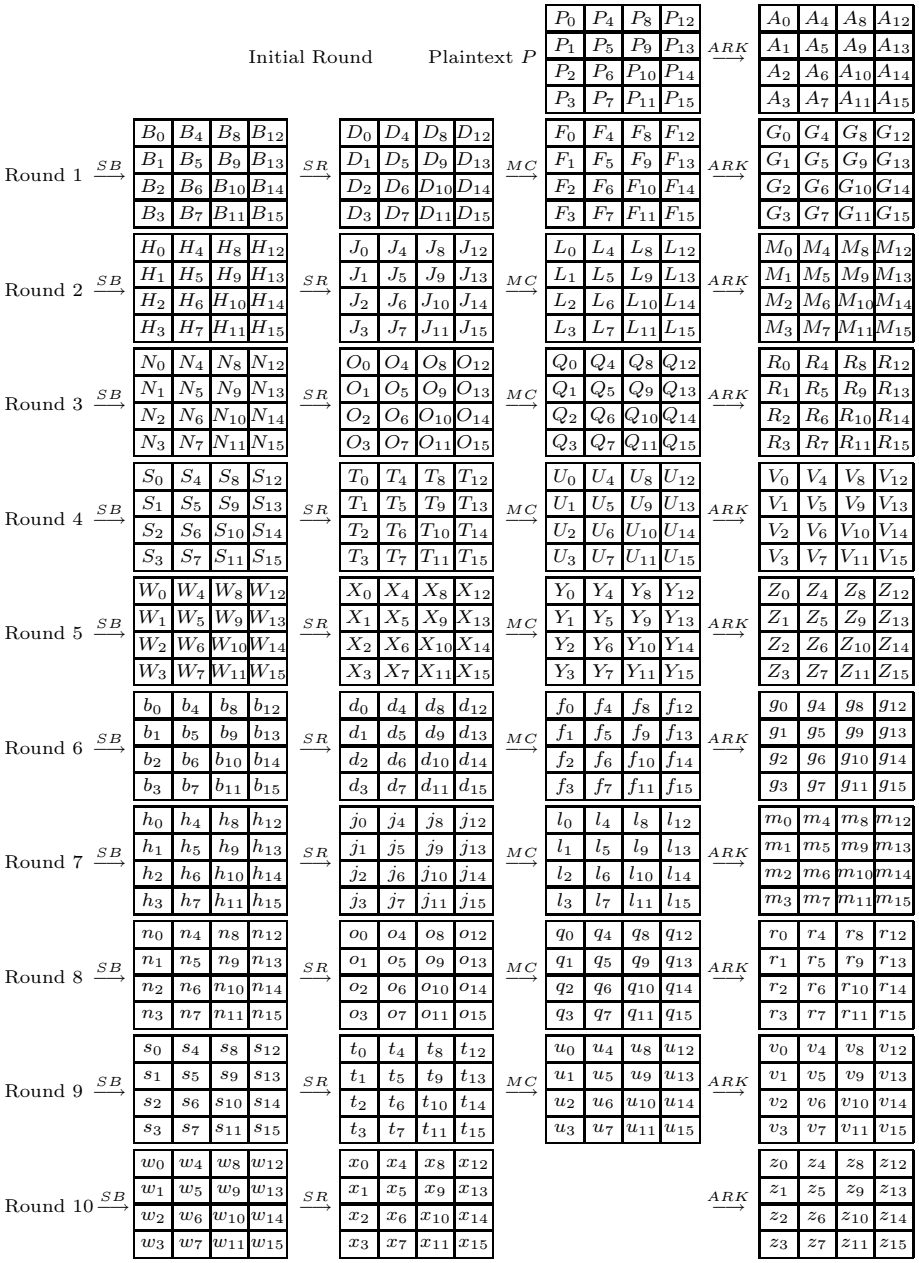

Fig. 1. The Intermediate Values of AES-128

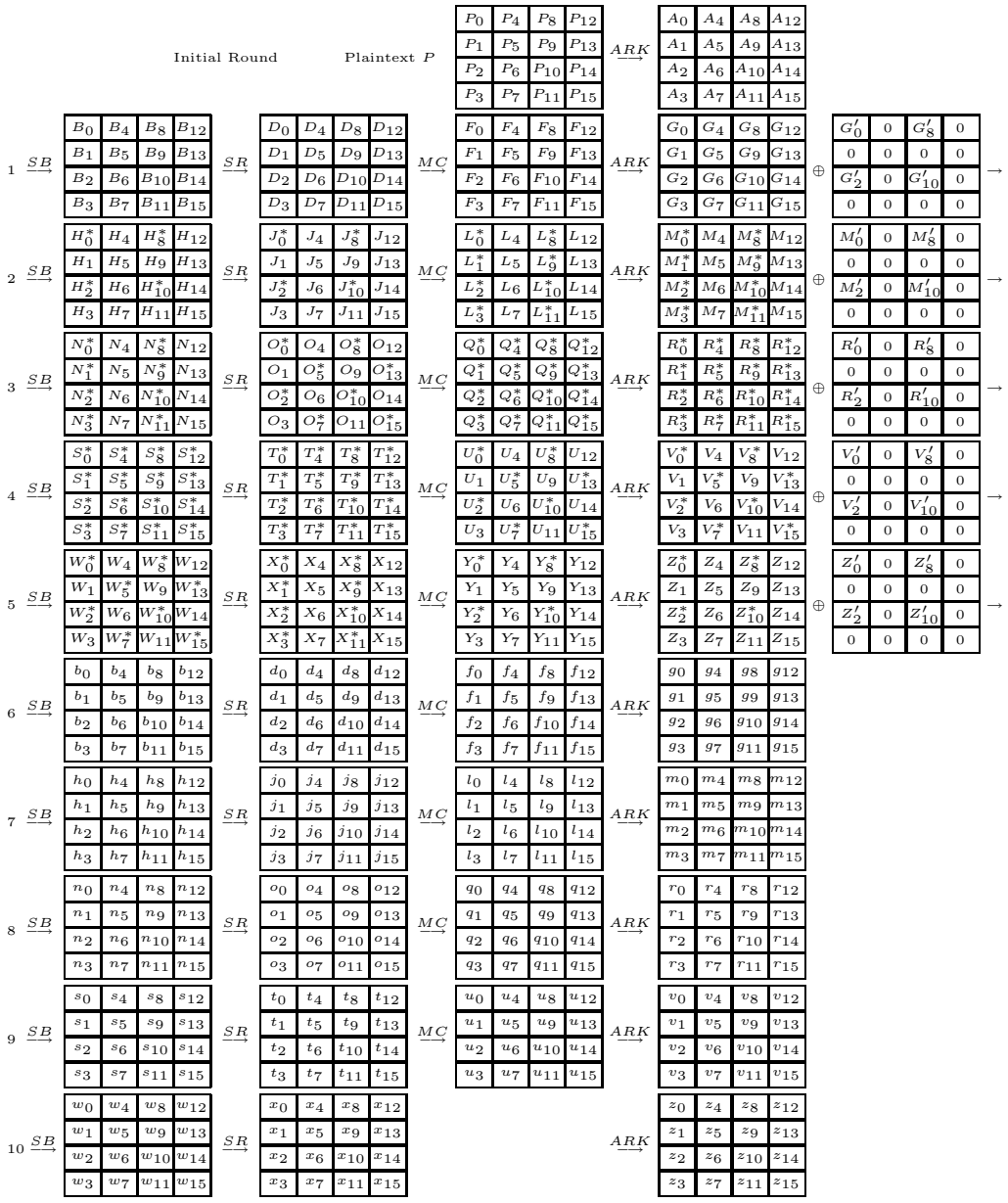

Fig. 2. The Intermediate Values of AES-128 with Extra 20 XOR Operations

(note that Routine Computation One produces 19 intermediate values).

$$
\begin{aligned}
&\texttt{Routine Computation One}\\
&\qquad\texttt{Initial round}: \xrightarrow{ARK}\\
&\qquad\texttt{Round 1}: \xrightarrow{SB}\ \xrightarrow{SR}\ \xrightarrow{MC}\ \xrightarrow{ARK}\ \xrightarrow{\oplus}\\
&\qquad\texttt{Round 2}: \xrightarrow{SB}\ \xrightarrow{SR}\ \xrightarrow{MC}\ \xrightarrow{ARK}\ \xrightarrow{\oplus}\\
&\qquad\texttt{Round 3}: \xrightarrow{SB}\ \xrightarrow{SR}\ \xrightarrow{MC}\ \xrightarrow{ARK}\ \xrightarrow{\oplus}\\
&\qquad\texttt{Round 4}: \xrightarrow{SB}\ \xrightarrow{SR}\ \xrightarrow{MC}.
\end{aligned}
$$

We denote the input and output of MC in Round 4 by

$$
\begin{bmatrix} T_0^* & T_4^* & T_8^* & T_{12}^* \\ T_1^* & T_5^* & T_9^* & T_{13}^* \\ T_2^* & T_6^* & T_{10}^* & T_{14}^* \\ T_3^* & T_7^* & T_{11}^* & T_{15}^* \end{bmatrix} \xrightarrow{MC} \begin{bmatrix} U_0^* & U_4^* & U_8^* & U_{12}^* \\ U_1^* & U_5^* & U_9^* & U_{13}^* \\ U_2^* & U_6^* & U_{10}^* & U_{14}^* \\ U_3^* & U_7^* & U_{11}^* & U_{15}^* \end{bmatrix}.
$$

Next, we will show that there is an algebraic relation between Bytes $\{M_0', M_2', M_8', M_{10}'\}$ and Bytes $\{U_4^*, U_6^*, U_{12}^*, U_{14}^*\}$. Based on this relationship, we can change the values of $\{U_4^*, U_6^*, U_{12}^*, U_{14}^*\}$ to the values of $\{U_4, U_6, U_{12}, U_{14}\}$ by setting the values of $\{M_0', M_2', M_8', M_{10}'\}$. After we have decided the values of $\{M_0', M_2', M_8', M_{10}'\}$, we aim to have an intermediate value after MC in Round 4 in the format of

$$
\begin{bmatrix} U_0^* & U_4 & U_8^* & U_{12} \\ U_1^* & U_5^* & U_9^* & U_{13}^* \\ U_2^* & U_6 & U_{10}^* & U_{14} \\ U_3^* & U_7^* & U_{11}^* & U_{15}^* \end{bmatrix}.
$$

The steps of deciding $\{M_0', M_2', M_8', M_{10}'\}$ are listed as follows:

$$\{M_0', M_2', M_8', M_{10}'\} \leftarrow \{N_0^*, N_2^*, N_8^*, N_{10}^*\} \leftarrow \{O_0^*, O_2^*, O_8^*, O_{10}^*\} \leftarrow \{Q_1^*, Q_3^*, Q_9^*, Q_{11}^*\}$$

$$\leftarrow \{R_1^*, R_3^*, R_9^*, R_{11}^*\} \leftarrow \{S_1^*, S_3^*, S_9^*, S_{11}^*\} \leftarrow \{T_5^*, T_7^*, T_{13}^*, T_{15}^*\} \leftarrow \{U_4, U_6, U_{12}, U_{14}\}.$$

After we change the values of $\{U_4^*, U_6^*, U_{12}^*, U_{14}^*\}$ to the values of $\{U_4, U_6, U_{12}, U_{14}\}$, the input and output of MC in Round 4 become

$$
\begin{bmatrix} T_0^* & T_4^* & T_8^* & T_{12}^* \\ T_1^* & T_5^* & T_9^* & T_{13}^* \\ T_2^* & T_6^* & T_{10}^* & T_{14}^* \\ T_3^* & T_7^* & T_{11}^* & T_{15}^* \end{bmatrix} \xrightarrow{MC} \begin{bmatrix} U_0^* & U_4 & U_8^* & U_{12} \\ U_1^* & U_5^* & U_9^* & U_{13}^* \\ U_2^* & U_6 & U_{10}^* & U_{14} \\ U_3^* & U_7^* & U_{11}^* & U_{15}^* \end{bmatrix}.
$$

Our next target is to modify the values of $\{T_5^*, T_7^*, T_{13}^*, T_{15}^*\}$ according to the values of $\{U_4, U_6, U_{12}, U_{14}\}$. From the MC transformation, we have the following

formula:

$$\begin{bmatrix} U_0^* & U_4 & U_8^* & U_{12} \\ U_1^* & U_5^* & U_9^* & U_{13}^* \\ U_2^* & U_6 & U_{10}^* & U_{14} \\ U_3^* & U_7^* & U_{11}^* & U_{15}^* \end{bmatrix} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} T_0^* & T_4^* & T_8^* & T_{12}^* \\ T_1^* & T_5^* & T_9^* & T_{13}^* \\ T_2^* & T_6^* & T_{10}^* & T_{14}^* \\ T_3^* & T_7^* & T_{11}^* & T_{15}^* \end{bmatrix}.$$

To find out the values of $\{T_5^*, T_7^*, T_{13}^*, T_{15}^*\}$, we need to solve the following two groups of linear functions, which are marked by (1) and (2).

$$\begin{cases} \begin{bmatrix} 02\ 03\ 01\ 01 \end{bmatrix} \begin{bmatrix} T_4^* \\ T_5^* \\ T_6^* \\ T_7^* \end{bmatrix} = U_4 \\ \begin{bmatrix} 01\ 01\ 02\ 03 \end{bmatrix} \begin{bmatrix} T_4^* \\ T_5^* \\ T_6^* \\ T_7^* \end{bmatrix} = U_6 \end{cases} \quad (1) \qquad \begin{cases} \begin{bmatrix} 02\ 03\ 01\ 01 \end{bmatrix} \begin{bmatrix} T_{12}^* \\ T_{13}^* \\ T_{14}^* \\ T_{15}^* \end{bmatrix} = U_{12} \\ \begin{bmatrix} 01\ 01\ 02\ 03 \end{bmatrix} \begin{bmatrix} T_{12}^* \\ T_{13}^* \\ T_{14}^* \\ T_{15}^* \end{bmatrix} = U_{14} \end{cases} \quad (2)$$

In (1), there are two linear equations with two undecided variables T_5^* and T_7^*, and thus we can solve (1) to obtain the values of T_5^* and T_7^*. Similarly, there are two linear equations in (2) with two undecided variables T_{13}^* and T_{15}^*, and therefore we can solve (2) to get the values of T_{13}^* and T_{15}^*. After having T_5^*, T_7^*, T_{13}^* and T_{15}^*, perform SR^{-1} (inverse ShiftRows) and SB^{-1} (inverse SubBytes), and we have the values of R_1^*, R_3^*, R_9^* and R_{11}^* after ARK in Round 3. Apply the ARK transformation to R_1^*, R_3^*, R_9^* and R_{11}^*, and we have the values of Q_1^*, Q_3^*, Q_9^* and Q_{11}^*. Our next task is to modify the values of O_0^*, O_2^*, O_8^* and O_{10}^*. In Round 3, the input and output of MC are as follows:

$$\begin{bmatrix} Q_0^* & Q_4^* & Q_8^* & Q_{12}^* \\ Q_1^* & Q_5^* & Q_9^* & Q_{13}^* \\ Q_2^* & Q_6^* & Q_{10}^* & Q_{14}^* \\ Q_3^* & Q_7^* & Q_{11}^* & Q_{15}^* \end{bmatrix} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} O_0^* & O_4 & O_8^* & O_{12} \\ O_1 & O_5^* & O_9 & O_{13}^* \\ O_2^* & O_6 & O_{10}^* & O_{14} \\ O_3 & O_7^* & O_{11} & O_{15}^* \end{bmatrix}.$$

We can form two groups of linear equations, which are named (3) and (4), and solve them to decide O_0^*, O_2^*, O_8^* and O_{10}^*. There are two linear equations in (3) with two undetermined variables O_0^* and O_2^*, and we can solve them to determine the values of O_0^* and O_2^*. Also, there are two linear equations in (4) with two undecided variables O_8^* and O_{10}^*, and we can get O_8^* and O_{10}^* by solving (4).

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} O_0^* \\ O_1 \\ O_2^* \\ O_3 \end{bmatrix} = Q_1^* \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} O_0^* \\ O_1 \\ O_2^* \\ O_3 \end{bmatrix} = Q_3^* \end{cases} \quad (3) \qquad \begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} O_8^* \\ O_9 \\ O_{10}^* \\ O_{11} \end{bmatrix} = Q_9^* \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} O_8^* \\ O_9 \\ O_{10}^* \\ O_{11} \end{bmatrix} = Q_{11}^* \end{cases} \quad (4)$$

Once knowing the values of O_0^*, O_2^*, O_8^* and O_{10}^*, we perform SR^{-1} and thus we get Bytes N_0^*, N_2^*, N_8^* and N_{10}^* after SB in Round 3. Finally, Bytes M_0', M_2', M_8' and M_{10}' are decided by the following computations (note that M_0^*, M_2^*, M_8^* and M_{10}^* are obtained from Buffer One):

$$M_0' = M_0^* \oplus SB^{-1}(N_0^*), \quad M_2' = M_2^* \oplus SB^{-1}(N_2^*),$$

$$M_8' = M_8^* \oplus SB^{-1}(N_8^*), \quad M_{10}' = M_{10}^* \oplus SB^{-1}(N_{10}^*).$$

At this stage, we have decided the values of $\{G_i', M_i'\}$, and $\{R_i', V_i', Z_i'\}$ are not yet decided (note: they are still initialized to zero), $i \in \{0, 2, 8, 10\}$.

The process of deciding R_0', R_2', R_8' and R_{10}' and the routine of finding V_0', V_2', V_8' and V_{10}' are similar to the steps of determining M_0', M_2', M_8' and M_{10}', and they are described in Appendix A and Appendix B respectively.

3.1.2 Deciding Z_0', Z_2', Z_8' and Z_{10}' Perform Routine Computation Two second time, and the intermediate value after MC in Round 5 is

$$\begin{bmatrix} Y_0^* & Y_4 & Y_8^* & Y_{12} \\ Y_1 & Y_5 & Y_9 & Y_{13} \\ Y_2^* & Y_6 & Y_{10}^* & Y_{14} \\ Y_3 & Y_7 & Y_{11} & Y_{15} \end{bmatrix}.$$

Apply ARK to the intermediate value above, we have

$$\begin{bmatrix} Z_0^* & Z_4 & Z_8^* & Z_{12} \\ Z_1 & Z_5 & Z_9 & Z_{13} \\ Z_2^* & Z_6 & Z_{10}^* & Z_{14} \\ Z_3 & Z_7 & Z_{11} & Z_{15} \end{bmatrix}.$$

Bytes Z_0', Z_2', Z_8' and Z_{10}' are computed as follows: (note that Z_0, Z_2, Z_8 and Z_{10} are obtained from the computation in which the AES algorithm is used to encrypt the plaintext P under the key K (see Round 5 in Figure 1)):

$$Z_0' = Z_0^* \oplus Z_0, \quad Z_2' = Z_2^* \oplus Z_2,$$

$$Z'_8 = Z^*_8 \oplus Z_8, \quad Z'_{10} = Z^*_{10} \oplus Z10.$$

Finally, we have decided all values of $\{G'_i, M'_i, R'_i, V'_i, Z'_i\}$, $i \in \{0, 2, 8, 10\}$. Now, we carry out a five-round computation of the AES with extra 20 XOR operations, called Routine Computation Three, by using Bytes G'_0, G'_2, G'_8, G'_{10}, M'_0, M'_2, M'_8, M'_{10}, R'_0, R'_2, R'_8, R'_{10}, V'_0, V'_2, V'_8, V'_{10}, Z'_0, Z'_2, Z'_8, and Z'_{10}, and we will get the same input to Round 6 as the AES algorithm.

Routine Computation Three

$$\begin{aligned}
\texttt{Initial round}&: \xrightarrow{ARK}\\
\texttt{Round 1}&: \xrightarrow{SB} \xrightarrow{SR} \xrightarrow{MC} \xrightarrow{ARK} \xrightarrow{\oplus}\\
\texttt{Round 2}&: \xrightarrow{SB} \xrightarrow{SR} \xrightarrow{MC} \xrightarrow{ARK} \xrightarrow{\oplus}\\
\texttt{Round 3}&: \xrightarrow{SB} \xrightarrow{SR} \xrightarrow{MC} \xrightarrow{ARK} \xrightarrow{\oplus}\\
\texttt{Round 4}&: \xrightarrow{SB} \xrightarrow{SR} \xrightarrow{MC} \xrightarrow{ARK} \xrightarrow{\oplus}\\
\texttt{Round 5}&: \xrightarrow{SB} \xrightarrow{SR} \xrightarrow{MC} \xrightarrow{ARK} \xrightarrow{\oplus}.
\end{aligned}$$

Remark 2. The most important part of the δ algorithm is solving those eight groups of linear equations: (1), (2), (3), (4), (5), (6), (7) and (8). There is one question needs to be answered. The question is: are these eight groups of linear equations independent? The answer to this question is choosing different values of Bytes G'_0, G'_2, G'_8, G'_{10} if we face such situations. Among the twenty bytes: G'_0, G'_2, G'_8, G'_{10}, M'_0, M'_2, M'_8, M'_{10}, R'_0, R'_2, R'_8, R'_{10}, V'_0, V'_2, V'_8, V'_{10}, Z'_0, Z'_2, Z'_8, and Z'_{10}, we can select the values of G'_0, G'_2, G'_8 and G'_{10} freely. As we showed in Remark 1, there are $2^{32} - 1$ combinations of these four bytes, and correspondingly, we can have $2^{32} - 1$ intermediate values in Figure 2, starting with SB in Round 2 and ending with ARK in Round 10. If we meet any dependent equations, we can overcome this problem by choosing different values of Bytes G'_0, G'_2, G'_8 and G'_{10}. Therefore, this question will not cause any trouble. So far, we have not met any dependent equations in our large-sample experiments.

Remark 3. From Remark 1, we note that there is more than one combination of the twenty output bytes of Algorithm δ for a given pair of (P, K).

Remark 4. For distinct plaintext and cipher key pairs (P, K), Algorithm δ needs to perform individual computations to decide the values of the twenty bytes.

3.2 Variants of Algorithm δ

We show that there are other variants of the δ algorithm. In section 3.1, the locations of the twenty bytes are $\{0, 2, 8, 10\}$, and there are three other combinations, which are $\{4, 6, 12, 14\}$, $\{1, 3, 9, 11\}$ and $\{5, 7, 13, 15\}$. In Figure 2, $\{G'_i, M'_i, R'_i, V'_i, Z'_i\}$ operate in Round $\{1, 2, 3, 4, 5\}$, and they can also operate in

Round $\{2,3,4,5,6\}$, $\{3,4,5,6,7\}$, $\{4,5,6,7,8\}$, or Round $\{5,6,7,8,9\}$. Therefore, there are 4 different combinations for the byte locations, and there are 5 different combinations for the round numbers in AES-128. In total, there are 20 ($= 4 \times 5$) variants of the δ algorithm for AES-128. The δ algorithm has 28 ($= 4 \times 7$) variants for AES-192, and 36 ($= 4 \times 9$) variants for AES-256.

4 The Modified Version of the AES: δAES

By employing the δ algorithm, we propose a modified version of the AES, which is named δAES. The major difference between the AES and the δAES is that the δAES uses modified AES round keys. In Figure 2 in Section 3, we apply 20 extra XOR operations to the intermediate values after ARK in Round 1, 2, 3, 4 and 5 by using Bytes $\{G'_i, M'_i, R'_i, V'_i, Z'_i\}, i \in \{0,2,8,10\}$. The construction of the δAES comes from the fact that we can use Bytes $\{G'_i, M'_i, R'_i, V'_i, Z'_i\}$ to XOR with AES Round Key 1, 2, 3, 4 and 5 (instead of with the intermediate values after ARK), and we still get the same result, $i \in \{0,2,8,10\}$. There are twenty-byte differences between the AES round keys and the δAES round keys. The δAES employs the same key scheduling algorithm, constants and round function (i.e., SubBytes, ShiftRows, MixColumns and AddRoundKey) as the AES.

The construction of the δAES is adding two procedures, which are calling the δ algorithm and modifying the AES round keys, to the AES algorithm.

1. Suppose for a plaintext P and a cipher key K, the AES algorithm produces a ciphertext C, written as $C = AES_K(P)$.
2. By accepting P and K as two inputs, use the δ algorithm to generate twenty output bytes:

$$\{G'_i, M'_i, R'_i, V'_i, Z'_i\}, \qquad i \in \{0,2,8,10\}^1.$$

3. Apply the AES key scheduling algorithm to K and get the round keys.
4. Use $\{G'_i, M'_i, R'_i, V'_i, Z'_i\}$ to XOR with the corresponding AES round keys and get the round keys for the δAES, $i \in \{0,2,8,10\}$.
5. After carrying out the transformations above, the δAES uses the same round function (i.e., SubBytes, ShiftRows, MixColumns and AddRoundKey) to process the plaintext P with modified AES round keys, and finally the δAES also generates the same ciphertext C, denoted by $C = \delta AES_K(P)$.

Compared with the AES algorithm, the δAES needs to do some extra transformations, i.e., calling the δ algorithm and modifying the AES round keys. Moreover, for distinct plaintext and cipher key pairs (P, K), the δAES needs to carry out individual computations to get Bytes $\{G'_i, M'_i, R'_i, V'_i, Z'_i\}, \in \{0,2,8,10\}$.

[1] For simplicity, we use only one variant of the δ algorithm here. Other variants of the δ algorithm also work.

5 Conclusions

We described a five-round algebraic property of the AES algorithm. In the presented property, we modify twenty bytes from five intermediate values at some fixed locations in five consecutive rounds by carrying out twenty extra XOR operations, and we show that after five rounds of processing, such modifications do not change the intermediate result and finally still produce the same ciphertext. We defined an algorithm named δ, and the δ algorithm takes a plaintext and a cipher key as two inputs and outputs twenty bytes, which are used in the five-round property. By employing the δ algorithm, we proposed a modified version of the AES algorithm, the δAES. The δAES uses the δ algorithm to generate twenty output bytes, which are used to modify the AES round keys. For a plaintext and a key, the AES and the δAES produce the same ciphertext.

References

1. Daemen, J., Rijmen, V.: AES Proposal: Rijndael, AES Round 1 Technical Evaluation CD-1: Documentation, National Institute of Standards and Technology. (1998)
2. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer, Heidelberg (1993)
3. Matsui, M.: Linear Cryptoanalysis Method for DES Cipher. In: Advances in Cryptology - EUROCRYPT 1993, LNCS, vol. 765, pp. 386-397. Springer, Heidelberg (1994)
4. NIST: Federal Information Processing Standards (FIPS) 197: Advanced Encryption Standard (AES). National Institute of Standards and Technology (26 November 2001)
5. Murphy, S., Robshaw, M.: Essential Algebraic Structure within the AES. In: Advances in Cryptology - CRYPTO 2002, LNCS, vol. 2442, pp. 1-16. Springer, Heidelberg (2002)
6. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In: Advances in Cryptology - EUROCRYPT 2000, LNCS, vol. 1807, pp. 392-407. Springer, Heidelberg (2000)
7. Courtois, N., Pieprzyk, J.: Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In: Advances in Cryptology - ASIACRYPT 2002, LNCS, vol. 2501, pp. 267-287. Springer, Heidelberg (2002)
8. Barkan, E., Biham, E.: In How Many Ways Can You Write Rijndael? In: Advances in Cryptology - ASIACRYPT 2002, LNCS, vol. 2501, pp. 160-175. Springer, Heidelberg (2002)
9. Gilbert, H., Minier, M.: A Collision Attack on 7 Rounds of Rijndael. In: The Third Advanced Encryption Standard Candidate Conference, pp. 230-241. (2000)
10. Daemen, J., Knudsen, L., Rijmen, V.: The Block Cipher Square. In: Fast Software Encryption 1997, LNCS, vol. 1267, pp. 149-165. Springer, Heidelberg (1997)
11. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved Cryptanalysis of Rijndael. In: Fast Software Encryption 2000, LNCS, vol. 1978, pp. 213-230. Springer, Heidelberg (2001)

12. Lucks, S.: Attacking Seven Rounds of Rijndael under 192-bit and 256-bit Keys. In: The Third Advanced Encryption Standard Candidate Conference, pp. 215-229. (2000)
13. Akkar, M.L., Giraud, C.: An Implementation of DES and AES, Secure against Some Attacks. In: Cryptographic Hardware and Embedded Systems 2001, LNCS, vol. 2162, pp. 309-318. Springer, Heidelberg (2001)
14. Golic, J., Tymen, C.: Multiplicative Masking and Power Analysis of AES. In: Cryptographic Hardware and Embedded Systems 2002, LNCS, vol. 2523, pp. 198-212. Springer, Heidelberg (2003)
15. Biryukov, A.: The Design of a Stream Cipher LEX. In: Selected Areas in Cryptography 2006, LNCS, vol. 4356, pp. 67-75. Springer, Heidelberg (2006)
16. Daemen, J., Rijmen, V.: A New MAC Construction ALRED and a Specific Instance ALPHA-MAC. In: Fast Software Encryption 2005, LNCS, vol. 3557, pp. 1-17. Springer, Heidelberg (2005)

A Deciding R'_0, R'_2, R'_8 and R'_{10}

Perform Routine Computation One second time, and all intermediate values from the computation of this time are stored in an array called Buffer Two. The intermediate value after MC in Round 4 is

$$\begin{bmatrix} U_0^* & U_4 & U_8^* & U_{12} \\ U_1^* & U_5^* & U_9^* & U_{13}^* \\ U_2^* & U_6 & U_{10}^* & U_{14} \\ U_3^* & U_7^* & U_{11}^* & U_{15}^* \end{bmatrix}.$$

We will demonstrate that there is an algebraic relation between Bytes $\{R'_0, R'_2, R'_8, R'_{10}\}$ and Bytes $\{U_1^*, U_3^*, U_9^*, U_{11}^*\}$. By employing this relationship, we are able to change the values of $\{U_1^*, U_3^*, U_9^*, U_{11}^*\}$ to the values of $\{U_1, U_3, U_9, U_{11}\}$ by choosing the values of $\{R'_0, R'_2, R'_8, R'_{10}\}$. The moves of determining the values of $\{R'_0, R'_2, R'_8, R'_{10}\}$ are shown below:

$$\{R'_0, R'_2, R'_8, R'_{10}\} \leftarrow \{S_0^*, S_2^*, S_8^*, S_{10}^*\} \leftarrow \{T_0^*, T_2^*, T_8^*, T_{10}^*\} \leftarrow \{U_1, U_3, U_9, U_{11}\}.$$

After we replace the values of $\{U_1^*, U_3^*, U_9^*, U_{11}^*\}$ with the values of $\{U_1, U_3, U_9, U_{11}\}$, the input and output of MC in Round 4 are

$$\begin{bmatrix} T_0^* & T_4^* & T_8^* & T_{12}^* \\ T_1^* & T_5^* & T_9^* & T_{13}^* \\ T_2^* & T_6^* & T_{10}^* & T_{14}^* \\ T_3^* & T_7^* & T_{11}^* & T_{15}^* \end{bmatrix} \xrightarrow{MC} \begin{bmatrix} U_0^* & U_4 & U_8^* & U_{12} \\ U_1 & U_5^* & U_9 & U_{13}^* \\ U_2^* & U_6 & U_{10}^* & U_{14} \\ U_3 & U_7^* & U_{11} & U_{15}^* \end{bmatrix}.$$

We need to modify the values of $\{T_0^*, T_2^*, T_8^*, T_{10}^*\}$ according to the values of $\{U_1, U_3, U_9, U_{11}\}$. We can form two groups of linear equations, which are named (5) and (6). There are two undecided variables T_0^* and T_2^* in (5), and we can solve (5) to get the values of T_0^* and T_2^*. In (6), there are two undetermined variables T_8^* and T_{10}^*, and we can find out the values of T_8^* and T_{10}^* by solving (6).

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} T_0^* \\ T_1^* \\ T_2^* \\ T_3^* \end{bmatrix} = U_1 \\ \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} T_0^* \\ T_1^* \\ T_2^* \\ T_3^* \end{bmatrix} = U_3 \end{cases} \quad (5) \qquad \begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} T_8^* \\ T_9^* \\ T_{10}^* \\ T_{11}^* \end{bmatrix} = U_9 \\ \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} T_8^* \\ T_9^* \\ T_{10}^* \\ T_{11}^* \end{bmatrix} = U_{11} \end{cases} \quad (6)$$

After knowing the values of $\{T_0^*, T_2^*, T_8^*, T_{10}^*\}$, we perform SR^{-1} and have four corresponding values $\{S_0^*, S_2^*, S_8^*, S_{10}^*\}$ after SB in Round 4. Bytes R_0', R_2', R_8' and R_{10}' are computed as follows: (note that R_0^*, R_2^*, R_8^* and R_{10}^* are obtained from Buffer Two):

$$R_0' = R_0^* \oplus SB^{-1}(S_0^*), \quad R_2' = R_2^* \oplus SB^{-1}(S_2^*),$$

$$R_8' = R_8^* \oplus SB^{-1}(S_8^*), \quad R_{10}' = R_{10}^* \oplus SB^{-1}(S_{10}^*).$$

At this moment, we have decided the values of $\{G_i', M_i', R_i'\}$, and $\{V_i', Z_i'\}$ are not determined and they are still equal to their initial values, $i \in \{0, 2, 8, 10\}$.

B Deciding V_0', V_2', V_8' and V_{10}'

After having the values of R_0', R_2', R_8' and R_{10}', we carry out a five-round computation of the AES with 16 extra XOR operations, called Routine Computation Two, which begins with the initial round and ends with MC in Round 5 (See Figure 2). All intermediate values from the computation of this time are stored in an array named Buffer Three (note that Routine Computation Two generates 24 intermediate values).

Routine Computation Two

$$\begin{aligned} \texttt{Initial round}&: \xrightarrow{ARK} \\ \texttt{Round 1}&: \xrightarrow{SB} \xrightarrow{SR} \xrightarrow{MC} \xrightarrow{ARK} \xrightarrow{\oplus} \\ \texttt{Round 2}&: \xrightarrow{SB} \xrightarrow{SR} \xrightarrow{MC} \xrightarrow{ARK} \xrightarrow{\oplus} \\ \texttt{Round 3}&: \xrightarrow{SB} \xrightarrow{SR} \xrightarrow{MC} \xrightarrow{ARK} \xrightarrow{\oplus} \\ \texttt{Round 4}&: \xrightarrow{SB} \xrightarrow{SR} \xrightarrow{MC} \xrightarrow{ARK} \xrightarrow{\oplus} \\ \texttt{Round 5}&: \xrightarrow{SB} \xrightarrow{SR} \xrightarrow{MC}. \end{aligned}$$

After MC in Round 5, we will have an intermediate value in the following format:

$$\begin{bmatrix} Y_0^* & Y_4 & Y_8^* & Y_{12} \\ Y_1^* & Y_5 & Y_9^* & Y_{13} \\ Y_2^* & Y_6 & Y_{10}^* & Y_{14} \\ Y_3^* & Y_7 & Y_{11}^* & Y_{15} \end{bmatrix}.$$

There is an algebraic relation between Bytes $\{V_0', V_2', V_8', V_{10}'\}$ and Bytes $\{Y_1^*, Y_3^*, Y_9^*, Y_{11}^*\}$, and we can change the values of $\{Y_1^*, Y_3^*, Y_9^*, Y_{11}^*\}$ to the values of $\{Y_1, Y_3, Y_9, Y_{11}\}$ by setting the values of $\{V_0', V_2', V_8', V_{10}'\}$. The steps of determining the values of $\{V_0', V_2', V_8', V_{10}'\}$ are shown below:

$$\{V_0', V_2', V_8', V_{10}'\} \leftarrow \{W_0^*, W_2^*, W_8^*, W_{10}^*\} \leftarrow \{X_0^*, X_2^*, X_8^*, X_{10}^*\} \leftarrow \{Y_1, Y_3, Y_9, Y_{11}\}.$$

We replace Bytes $\{Y_1^*, Y_3^*, Y_9^*, Y_{11}^*\}$ with Bytes $\{Y_1, Y_3, Y_9, Y_{11}\}$, and the input and output of MC in Round 5 are

$$\begin{bmatrix} X_0^* & X_4 & X_8^* & X_{12} \\ X_1^* & X_5 & X_9^* & X_{13} \\ X_2^* & X_6 & X_{10}^* & X_{14} \\ X_3^* & X_7 & X_{11}^* & X_{15} \end{bmatrix} \xrightarrow{MC} \begin{bmatrix} Y_0^* & Y_4 & Y_8^* & Y_{12} \\ Y_1 & Y_5 & Y_9 & Y_{13} \\ Y_2^* & Y_6 & Y_{10}^* & Y_{14} \\ Y_3 & Y_7 & Y_{11} & Y_{15} \end{bmatrix}.$$

We form two groups of linear functions, marked by (7) and (8). There are two undecided variables X_0^* and X_2^* in (7), and we can solve (7) to get the values of X_0^* and X_2^*. In (8), there are two undecided variables X_8^* and X_{10}^*, and we can obtain the values of X_8^* and X_{10}^* by solving (8).

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} X_0^* \\ X_1^* \\ X_2^* \\ X_3^* \end{bmatrix} = Y_1 \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} X_0^* \\ X_1^* \\ X_2^* \\ X_3^* \end{bmatrix} = Y_3 \end{cases} \quad (7) \qquad \begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} X_8^* \\ X_9^* \\ X_{10}^* \\ X_{11}^* \end{bmatrix} = Y_9 \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} X_8^* \\ X_9^* \\ X_{10}^* \\ X_{11}^* \end{bmatrix} = Y_{11} \end{cases} \quad (8)$$

After deciding the values of $\{X_0^*, X_2^*, X_8^*, X_{10}^*\}$, we perform SR^{-1} and have four corresponding values $\{W_0^*, W_2^*, W_8^*, W_{10}^*\}$ after SB in Round 5. Bytes V_0', V_2', V_8' and V_{10}' are computed as follows: (note that V_0^*, V_2^*, V_8^* and V_{10}^* are obtained from Buffer Three):

$$V_0' = V_0^* \oplus SB^{-1}(W_0^*), \quad V_2' = V_2^* \oplus SB^{-1}(W_2^*),$$
$$V_8' = V_8^* \oplus SB^{-1}(W_8^*), \quad V_{10}' = V_{10}^* \oplus SB^{-1}(W_{10}^*).$$

At this stage, we have decided the values of $\{G_i', M_i', R_i', V_i'\}$, and Z_i' is not determined and it is equal to the initial value, $i \in \{0, 2, 8, 10\}$.