2009

A five-round algebraic property of AES and its application to the ALPHA-MAC

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 AES and its application to the ALPHA-MAC

Abstract

We present a five-round algebraic property of the advanced encryption standard (AES), and we show that this algebraic property can be used to analyse the internal structure of ALPHA-MAC whose underlying block cipher is AES. In the proposed property, we modify 20 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. By employing the proposed five-round algebraic property of AES, we provide a method to find second preimages of the ALPHA-MAC based on the assumption that a key or an intermediate value is known. We also show that our idea can also be used to find internal collisions of the ALPHA-MAC under the same assumption.

Disciplines

Physical Sciences and Mathematics

Publication Details

Huang, J., Seberry, J. & Susilo, W. (2009). A five-round algebraic property of AES and its application to the ALPHA-MAC. International Journal of Applied Cryptography, 1 (4), 264-289.

A five-round algebraic property of AES and its application to the ALPHA-MAC

Jianyong Huang*, Jennifer Seberry and Willy Susilo

Centre for Computer and Information Security Research,
School of Computer Science and Software Engineering,
University of Wollongong, Australia
E-mail: jyh33@uow.edu.au
E-mail: jennie@uow.edu.au
E-mail: wsusilo@uow.edu.au
*Corresponding author

Abstract: We present a five-round algebraic property of the advanced encryption standard (AES), and we show that this algebraic property can be used to analyse the internal structure of ALPHA-MAC whose underlying block cipher is AES. In the proposed property, we modify 20 bytes from 5 intermediate values at some fixed locations in 5 consecutive rounds, and we show that after 5 rounds of operations, such modifications do not change the intermediate result and finally, still produce the same ciphertext. By employing the proposed five-round algebraic property of AES, we provide a method to find second preimages of the ALPHA-MAC based on the assumption that a key or an intermediate value is known. We also show that our idea can also be used to find internal collisions of the ALPHA-MAC under the same assumption.

Keywords: AES; advanced encryption standard; algebraic property; ALPHA-MAC; internal collisions; second preimages.

Reference to this paper should be made as follows: Huang, J., Seberry, J. and Susilo, W. (xxxx) 'A five-round algebraic property of AES and its application to the ALPHA-MAC', *Int. J. Applied Cryptography*, Vol. x, No. x, pp.xx–xx.

Biographical notes: Jianyong Huang is a PhD Student at the University of Wollongong. His current research interests include analysis of hash functions and block ciphers.

Jennifer Seberry is the Professor of Computer Science at the University of Wollongong. She graduated PhD in Computation Mathematics from La Trobe University in 1971. She has published extensively in Discrete Mathematics and is renown for her new discoveries on Hadamard Matrices and Statistical Designs. Her studies of the application of discrete mathematics and combinatorial computing via bent functions, S-box design, has led to the design of secure crypto-algorithms and strong hashing algorithms for secure and reliable information transfer in networks. She has over 300 publications and has successfully supervised 28 PhD Theses.

Willy Susilo is an Associate Professor at the School of Computer Science and Software Engineering in the University of Wollongong. He is also the Deputy Director of the ICT Research Institute and the Director of the CCISR research group. His research interests include authentication and digital signature schemes.

1 Introduction

The block cipher Rijndael, invented by Daemen and Rijmen (2001), 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, proposed by Biham and Shamir (1993) and linear cryptanalysis, described by Matsui (1994). 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 by National Institute of Standards and Technology (2001), there have been many research efforts aiming to evaluate the security of this cipher. A block cipher, named big encryption system (BES), was defined by Murphy and Robshaw (2002), and Rijndael can be embedded into BES. The extended linearisation (XL) proposed by Courtois et al. (2000) and the extended sparse linearisation (XSL) provided by Courtois and Pieprzyk (2002) are new methods to solve non-linear algebraic equations. The concept of dual ciphers was introduced by Barkan and Biham (2002), and a collision attack on seven rounds of Rijndael was described by Gilbert and Minier (2000). The most effective attacks on reduced round variants of the AES are square attack which was found by Daemen et al. (1997). The idea of the square attack was later employed by Ferguson et al. (2001) to improve the cryptanalysis of Rijndael, and by Lucks (2000) to attack seven rounds of Rijndael under 192- and 256-bit keys. A multiplicative masking method of AES was proposed by Akkar and Giraud (2001) and further discussed by Golic and Tymen (2002). The design of an AES-based stream cipher LEX was described by Biryukov (2007). A new message authentication code (MAC) construction ALRED and a special instance ALPHA-MAC was designed by Daemen and Rijmen (2005). 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 20 bytes from 5 intermediate values at some fixed locations in 5 consecutive rounds, and we demonstrate that after 5 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 20 bytes, which are used in the 5-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 20 bytes, and uses these 20 bytes to modify the AES round keys. For a plaintext and a key, the AES and the δAES produce the same ciphertext. By employing the proposed algebraic property of the AES, we analyse the internal structure of the ALPHA-MAC. Firstly, we present a method to find second preimages of the ALPHA-MAC by solving eight groups of linear functions based on the assumption that an authentication key or an intermediate value of this MAC is known. Each of these eight groups of linear functions contains two equations. Secondly, we show that the second-preimage finding method can also be used to generate internal collisions. The proposed collision search method can find two five-block messages such that they produce 128-bit collisions under a selected key (or a selected intermediate value).

This paper is organised as follows: Section 2 provides a brief description of the AES algorithm and Section 3 describes a five-round algebraic property of the AES. A modified version of the AES is defined in Section 4. Section 5 shows a description of the ALPHA-MAC construction and Section 6 demonstrates how the proposed five-round property of the AES is used to find second preimages and internal collisions of the ALPAH-MAC. Section 7 concludes this paper. Some examples of the AES and the AES with 20 extra exclusive-or operations are provided in the Appendix.

2 Description of the AES

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 organised in the following format:

α_0	α_4	α_8	α_{12}
α_1	α_5	α_9	α_{13}
α_2	α_6	α_{10}	α_{14}
α_3	α_7	α_{11}	α_{15}

where α_i denote the ith 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-, 192- 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 AddRoundKey (ARK)):

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 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-, 192- or 256-bit key sizes, respectively. For a 128-bit key K, we denote the round keys by

K_0^i	K_4^i	K_8^i	K_{12}^i
K_1^i	K_5^i	K_9^i	K_{13}^i
K_2^i	K_6^i	K_{10}^i	K_{14}^i
K_3^i	K_7^i	K_{11}^i	K_{15}^i

where i is the round number, $i \in \{1, 2, \ldots, 10\}$. 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. The combinations of the key length, block size and number of rounds are listed below:

Key length	*Block size*	*Number of round*
128 bits	4	10
192 bits	4	12
256 bits	4	14

3 A five-round property of AES

We describe a five-round property of the AES in this section. In the proposed property, we modify 20 bytes from 5 intermediate values at some fixed locations in 5 consecutive rounds, and we show that after 5 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 20 extra XOR operations in 5 rounds. We require that each of these 5 rounds must contain SB, SR, MC and ARK transformations.

We use Figures 1–3 to describe this property. The layout of the 20 bytes in the 5 intermediate values is shown in Figure 1, and the 20 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}.

Figure 1 20 bytes

$$\begin{array}{|c|c|c|c|}\hline G'_0 & 0 & G'_8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline G'_2 & 0 & G'_{10} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline M'_0 & 0 & M'_8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline M'_2 & 0 & M'_{10} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline R'_0 & 0 & R'_8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline R'_2 & 0 & R'_{10} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline V'_0 & 0 & V'_8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline V'_2 & 0 & V'_{10} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline Z'_0 & 0 & Z'_8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline Z'_2 & 0 & Z'_{10} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array}$$

Figure 2 The intermediate values of AES-128

Initial Round Plaintext P
$$\begin{array}{|c|c|c|c|}\hline P_0 & P_4 & P_8 & P_{12} \\ \hline P_1 & P_5 & P_9 & P_{13} \\ \hline P_2 & P_6 & P_{10} & P_{14} \\ \hline P_3 & P_7 & P_{11} & P_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline A_0 & A_4 & A_8 & A_{12} \\ \hline A_1 & A_5 & A_9 & A_{13} \\ \hline A_2 & A_6 & A_{10} & A_{14} \\ \hline A_3 & A_7 & A_{11} & A_{15} \\ \hline \end{array}$$

Round 1
$$\xrightarrow{SB} \begin{array}{|c|c|c|c|}\hline B_0 & B_4 & B_8 & B_{12} \\ \hline B_1 & B_5 & B_9 & B_{13} \\ \hline B_2 & B_6 & B_{10} & B_{14} \\ \hline B_3 & B_7 & B_{11} & B_{15} \\ \hline \end{array} \xrightarrow{SR} \begin{array}{|c|c|c|c|}\hline D_0 & D_4 & D_8 & D_{12} \\ \hline D_1 & D_5 & D_9 & D_{13} \\ \hline D_2 & D_6 & D_{10} & D_{14} \\ \hline D_3 & D_7 & D_{11} & D_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline F_0 & F_4 & F_8 & F_{12} \\ \hline F_1 & F_5 & F_9 & F_{13} \\ \hline F_2 & F_6 & F_{10} & F_{14} \\ \hline F_3 & F_7 & F_{11} & F_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline G_0 & G_4 & G_8 & G_{12} \\ \hline G_1 & G_5 & G_9 & G_{13} \\ \hline G_2 & G_6 & G_{10} & G_{14} \\ \hline G_3 & G_7 & G_{11} & G_{15} \\ \hline \end{array}$$

Round 2
$$\xrightarrow{SB} \begin{array}{|c|c|c|c|}\hline H_0 & H_4 & H_8 & H_{12} \\ \hline H_1 & H_5 & H_9 & H_{13} \\ \hline H_2 & H_6 & H_{10} & H_{14} \\ \hline H_3 & H_7 & H_{11} & H_{15} \\ \hline \end{array} \xrightarrow{SR} \begin{array}{|c|c|c|c|}\hline J_0 & J_4 & J_8 & J_{12} \\ \hline J_1 & J_5 & J_9 & J_{13} \\ \hline J_2 & J_6 & J_{10} & J_{14} \\ \hline J_3 & J_7 & J_{11} & J_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline L_0 & L_4 & L_8 & L_{12} \\ \hline L_1 & L_5 & L_9 & L_{13} \\ \hline L_2 & L_6 & L_{10} & L_{14} \\ \hline L_3 & L_7 & L_{11} & L_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline M_0 & M_4 & M_8 & M_{12} \\ \hline M_1 & M_5 & M_9 & M_{13} \\ \hline M_2 & M_6 & M_{10} & M_{14} \\ \hline M_3 & M_7 & M_{11} & M_{15} \\ \hline \end{array}$$

Round 3
$$\xrightarrow{SB} \begin{array}{|c|c|c|c|}\hline N_0 & N_4 & N_8 & N_{12} \\ \hline N_1 & N_5 & N_9 & N_{13} \\ \hline N_2 & N_6 & N_{10} & N_{14} \\ \hline N_3 & N_7 & N_{11} & N_{15} \\ \hline \end{array} \xrightarrow{SR} \begin{array}{|c|c|c|c|}\hline O_0 & O_4 & O_8 & O_{12} \\ \hline O_1 & O_5 & O_9 & O_{13} \\ \hline O_2 & O_6 & O_{10} & O_{14} \\ \hline O_3 & O_7 & O_{11} & O_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline Q_0 & Q_4 & Q_8 & Q_{12} \\ \hline Q_1 & Q_5 & Q_9 & Q_{13} \\ \hline Q_2 & Q_6 & Q_{10} & Q_{14} \\ \hline Q_3 & Q_7 & Q_{11} & Q_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline R_0 & R_4 & R_8 & R_{12} \\ \hline R_1 & R_5 & R_9 & R_{13} \\ \hline R_2 & R_6 & R_{10} & R_{14} \\ \hline R_3 & R_7 & R_{11} & R_{15} \\ \hline \end{array}$$

Round 4
$$\xrightarrow{SB} \begin{array}{|c|c|c|c|}\hline S_0 & S_4 & S_8 & S_{12} \\ \hline S_1 & S_5 & S_9 & S_{13} \\ \hline S_2 & S_6 & S_{10} & S_{14} \\ \hline S_3 & S_7 & S_{11} & S_{15} \\ \hline \end{array} \xrightarrow{SR} \begin{array}{|c|c|c|c|}\hline T_0 & T_4 & T_8 & T_{12} \\ \hline T_1 & T_5 & T_9 & T_{13} \\ \hline T_2 & T_6 & T_{10} & T_{14} \\ \hline T_3 & T_7 & T_{11} & T_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline U_0 & U_4 & U_8 & U_{12} \\ \hline U_1 & U_5 & U_9 & U_{13} \\ \hline U_2 & U_6 & U_{10} & U_{14} \\ \hline U_3 & U_7 & U_{11} & U_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline V_0 & V_4 & V_8 & V_{12} \\ \hline V_1 & V_5 & V_9 & V_{13} \\ \hline V_2 & V_6 & V_{10} & V_{14} \\ \hline V_3 & V_7 & V_{11} & V_{15} \\ \hline \end{array}$$

Round 5
$$\xrightarrow{SB} \begin{array}{|c|c|c|c|}\hline W_0 & W_4 & W_8 & W_{12} \\ \hline W_1 & W_5 & W_9 & W_{13} \\ \hline W_2 & W_6 & W_{10} & W_{14} \\ \hline W_3 & W_7 & W_{11} & W_{15} \\ \hline \end{array} \xrightarrow{SR} \begin{array}{|c|c|c|c|}\hline X_0 & X_4 & X_8 & X_{12} \\ \hline X_1 & X_5 & X_9 & X_{13} \\ \hline X_2 & X_6 & X_{10} & X_{14} \\ \hline X_3 & X_7 & X_{11} & X_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline Y_0 & Y_4 & Y_8 & Y_{12} \\ \hline Y_1 & Y_5 & Y_9 & Y_{13} \\ \hline Y_2 & Y_6 & Y_{10} & Y_{14} \\ \hline Y_3 & Y_7 & Y_{11} & Y_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline Z_0 & Z_4 & Z_8 & Z_{12} \\ \hline Z_1 & Z_5 & Z_9 & Z_{13} \\ \hline Z_2 & Z_6 & Z_{10} & Z_{14} \\ \hline Z_3 & Z_7 & Z_{11} & Z_{15} \\ \hline \end{array}$$

Round 6
$$\xrightarrow{SB} \begin{array}{|c|c|c|c|}\hline b_0 & b_4 & b_8 & b_{12} \\ \hline b_1 & b_5 & b_9 & b_{13} \\ \hline b_2 & b_6 & b_{10} & b_{14} \\ \hline b_3 & b_7 & b_{11} & b_{15} \\ \hline \end{array} \xrightarrow{SR} \begin{array}{|c|c|c|c|}\hline d_0 & d_4 & d_8 & d_{12} \\ \hline d_1 & d_5 & d_9 & d_{13} \\ \hline d_2 & d_6 & d_{10} & d_{14} \\ \hline d_3 & d_7 & d_{11} & d_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline f_0 & f_4 & f_8 & f_{12} \\ \hline f_1 & f_5 & f_9 & f_{13} \\ \hline f_2 & f_6 & f_{10} & f_{14} \\ \hline f_3 & f_7 & f_{11} & f_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline g_0 & g_4 & g_8 & g_{12} \\ \hline g_1 & g_5 & g_9 & g_{13} \\ \hline g_2 & g_6 & g_{10} & g_{14} \\ \hline g_3 & g_7 & g_{11} & g_{15} \\ \hline \end{array}$$

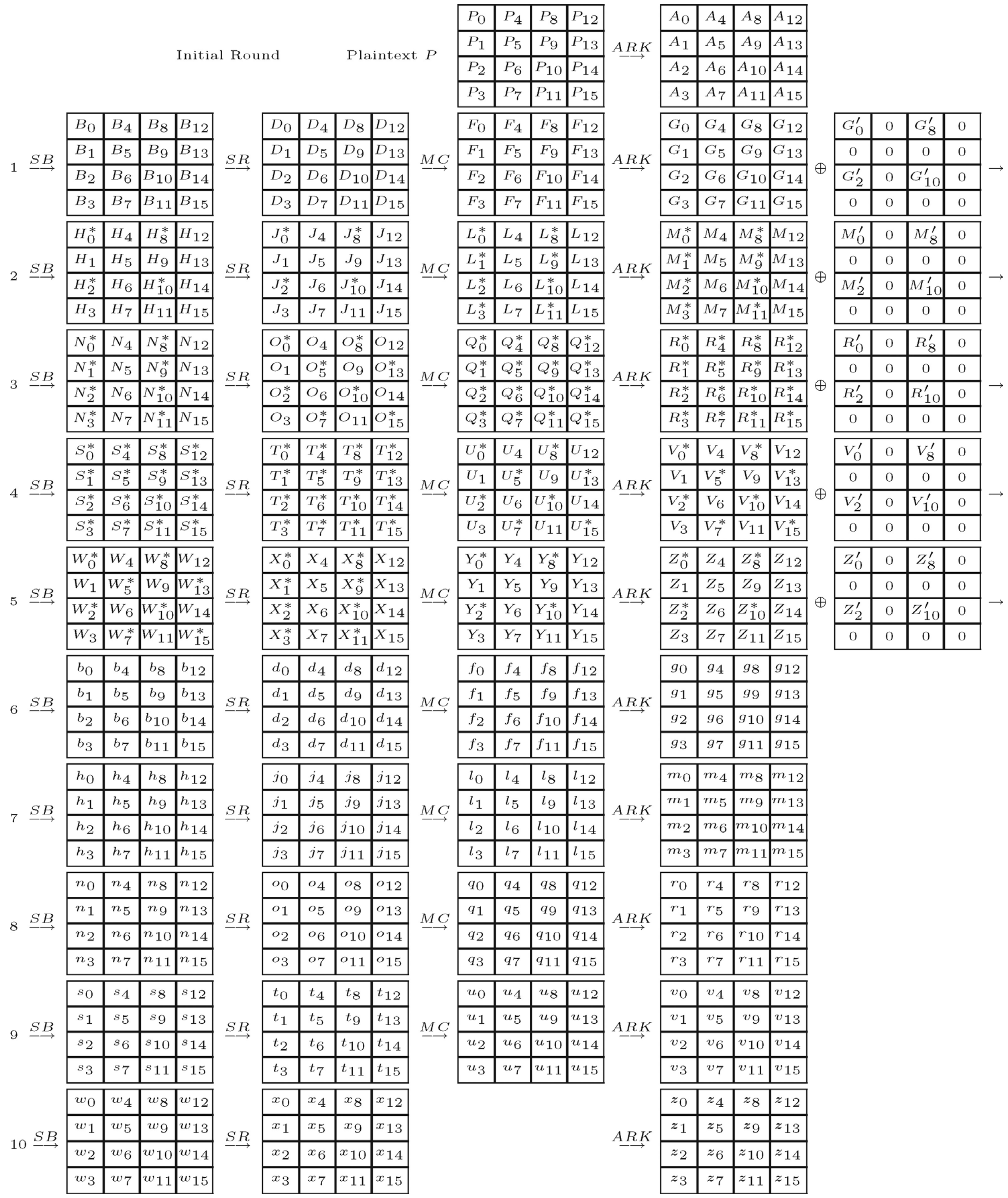
Round 7
$$\xrightarrow{SB} \begin{array}{|c|c|c|c|}\hline h_0 & h_4 & h_8 & h_{12} \\ \hline h_1 & h_5 & h_9 & h_{13} \\ \hline h_2 & h_6 & h_{10} & h_{14} \\ \hline h_3 & h_7 & h_{11} & h_{15} \\ \hline \end{array} \xrightarrow{SR} \begin{array}{|c|c|c|c|}\hline j_0 & j_4 & j_8 & j_{12} \\ \hline j_1 & j_5 & j_9 & j_{13} \\ \hline j_2 & j_6 & j_{10} & j_{14} \\ \hline j_3 & j_7 & j_{11} & j_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline l_0 & l_4 & l_8 & l_{12} \\ \hline l_1 & l_5 & l_9 & l_{13} \\ \hline l_2 & l_6 & l_{10} & l_{14} \\ \hline l_3 & l_7 & l_{11} & l_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline m_0 & m_4 & m_8 & m_{12} \\ \hline m_1 & m_5 & m_9 & m_{13} \\ \hline m_2 & m_6 & m_{10} & m_{14} \\ \hline m_3 & m_7 & m_{11} & m_{15} \\ \hline \end{array}$$

Round 8
$$\xrightarrow{SB} \begin{array}{|c|c|c|c|}\hline n_0 & n_4 & n_8 & n_{12} \\ \hline n_1 & n_5 & n_9 & n_{13} \\ \hline n_2 & n_6 & n_{10} & n_{14} \\ \hline n_3 & n_7 & n_{11} & n_{15} \\ \hline \end{array} \xrightarrow{SR} \begin{array}{|c|c|c|c|}\hline o_0 & o_4 & o_8 & o_{12} \\ \hline o_1 & o_5 & o_9 & o_{13} \\ \hline o_2 & o_6 & o_{10} & o_{14} \\ \hline o_3 & o_7 & o_{11} & o_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline q_0 & q_4 & q_8 & q_{12} \\ \hline q_1 & q_5 & q_9 & q_{13} \\ \hline q_2 & q_6 & q_{10} & q_{14} \\ \hline q_3 & q_7 & q_{11} & q_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline r_0 & r_4 & r_8 & r_{12} \\ \hline r_1 & r_5 & r_9 & r_{13} \\ \hline r_2 & r_6 & r_{10} & r_{14} \\ \hline r_3 & r_7 & r_{11} & r_{15} \\ \hline \end{array}$$

Round 9
$$\xrightarrow{SB} \begin{array}{|c|c|c|c|}\hline s_0 & s_4 & s_8 & s_{12} \\ \hline s_1 & s_5 & s_9 & s_{13} \\ \hline s_2 & s_6 & s_{10} & s_{14} \\ \hline s_3 & s_7 & s_{11} & s_{15} \\ \hline \end{array} \xrightarrow{SR} \begin{array}{|c|c|c|c|}\hline t_0 & t_4 & t_8 & t_{12} \\ \hline t_1 & t_5 & t_9 & t_{13} \\ \hline t_2 & t_6 & t_{10} & t_{14} \\ \hline t_3 & t_7 & t_{11} & t_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline u_0 & u_4 & u_8 & u_{12} \\ \hline u_1 & u_5 & u_9 & u_{13} \\ \hline u_2 & u_6 & u_{10} & u_{14} \\ \hline u_3 & u_7 & u_{11} & u_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline v_0 & v_4 & v_8 & v_{12} \\ \hline v_1 & v_5 & v_9 & v_{13} \\ \hline v_2 & v_6 & v_{10} & v_{14} \\ \hline v_3 & v_7 & v_{11} & v_{15} \\ \hline \end{array}$$

Round 10
$$\xrightarrow{SB} \begin{array}{|c|c|c|c|}\hline w_0 & w_4 & w_8 & w_{12} \\ \hline w_1 & w_5 & w_9 & w_{13} \\ \hline w_2 & w_6 & w_{10} & w_{14} \\ \hline w_3 & w_7 & w_{11} & w_{15} \\ \hline \end{array} \xrightarrow{SR} \begin{array}{|c|c|c|c|}\hline x_0 & x_4 & x_8 & x_{12} \\ \hline x_1 & x_5 & x_9 & x_{13} \\ \hline x_2 & x_6 & x_{10} & x_{14} \\ \hline x_3 & x_7 & x_{11} & x_{15} \\ \hline \end{array} \xrightarrow{ARK} \begin{array}{|c|c|c|c|}\hline z_0 & z_4 & z_8 & z_{12} \\ \hline z_1 & z_5 & z_9 & z_{13} \\ \hline z_2 & z_6 & z_{10} & z_{14} \\ \hline z_3 & z_7 & z_{11} & z_{15} \\ \hline \end{array}$$

Figure 3 The intermediate values of AES-128 with extra 20 XOR operations



In Figure 1, a zero occupied byte means that there is no change in that byte, and a variable occupied byte indicates that there is a modification in that byte. In Figure 2, 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 3 enumerates all intermediate values of the AES with 20 extra XOR operations. The 20-byte modifications take place in Rounds 1–5, and after ARK transformation in each of these 5 rounds, we perform XOR operations on Bytes 0, 2, 8 and 10. We show that the 20-byte modifications do not change the input to Round 6, that is, both the AES and the AES with 20 extra XOR operations generate the same input to Round 6. In Figure 3, a variable marked by a asterisk indicates that the value at that location has been affected by the 20-byte modifications, and a plain variable shows that the value at that location is not affected by the 20-byte modifications. For example, after ARK in Round 1 in Figure 3, 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 12 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 20 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 20 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, that is, start with the initial round, and process Rounds 1–5 of the AES. Therefore, we know all intermediate values in Figure 2, from initial round to Round 5.

2 Initialise 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 20 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 Sections 3.1.1–3.1.4.

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 3).

Routine Computation One
Initial round : $\underrightarrow{\text{ARK}}$

Round 1: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$ $\underrightarrow{\text{ARK}}$ $\underrightarrow{\oplus}$

Round 2: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$ $\underrightarrow{\text{ARK}}$ $\underrightarrow{\oplus}$

Round 3: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$ $\underrightarrow{\text{ARK}}$ $\underrightarrow{\oplus}$

Round 4: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$.

All intermediate values from the computation of this time are stored in array called Buffer One (note that Routine Computation One produces 19 intermediate values). We denote the input and output of MC in Round 4 by

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

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:

$$\{M'_0, M'_2, M'_8, M'_{10}\} \leftarrow \left\{N_0^*, N_2^*, N_8^* N_{10}^*\right\} \leftarrow \left\{O_0^*, O_2^*, O_8^*, O_{10}^*\right\}$$
$$\leftarrow \left\{Q_1^*, Q_3^*, Q_9^*, Q_{11}^*\right\} \leftarrow \left\{R_1^*, R_3^*, R_9^*, R_{11}^*\right\} \leftarrow \left\{S_1^*, S_3^*, S_9^*, S_{11}^*\right\}$$
$$\leftarrow \left\{T_5^*, T_7^*, T_{13}^*, T_{15}^*\right\} \leftarrow \{U_4, U_6, U_{12}, U_{14}\}$$

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

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

Our next target is to modify the values $\{T_5^*, T_7^*, T_{13}^*, T_{15}^*\}$ of 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).

$$\begin{cases} \begin{bmatrix} 02 & 03 & 01 & 01 \end{bmatrix} \begin{bmatrix} T_4^* \\ T_5^* \\ T_6^* \\ T_7^* \end{bmatrix} = U_4 \\ \begin{bmatrix} 01 & 01 & 02 & 03 \end{bmatrix} \begin{bmatrix} T_4^* \\ T_5^* \\ T_6^* \\ T_7^* \end{bmatrix} = U_6 \end{cases} \quad (1)$$

$$\begin{cases} \begin{bmatrix} 02 & 03 & 01 & 01 \end{bmatrix} \begin{bmatrix} T_{12}^* \\ T_{13}^* \\ T_{14}^* \\ T_{15}^* \end{bmatrix} = U_{12} \\ \begin{bmatrix} 01 & 01 & 02 & 03 \end{bmatrix} \begin{bmatrix} T_{12}^* \\ T_{13}^* \\ T_{14}^* \\ T_{15}^* \end{bmatrix} = U_{14} \end{cases} \quad (2)$$

In Equation (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 SR) and SB^{-1} (inverse SB), 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}^*, 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}^* and by solving (4).

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} O_0^* \\ O_1 \\ O_2^* \\ O_3 \end{bmatrix} = Q_4^* \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} O_0 \\ O_1 \\ O_2^* \\ O_3 \end{bmatrix} = Q_3^* \end{cases} \quad (3)$$

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} O_8^* \\ O_9 \\ O_{10}^* \\ O_{11} \end{bmatrix} = Q_9^* \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} O_8^* \\ O_9 \\ O_{10}^* \\ O_{11} \end{bmatrix} = Q_{11}^* \end{cases} \quad (4)$$

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):

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

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

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 initialised to zero), $i \in \{0, 2, 8, 10\}$.

3.1.2 *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 $\{R'_0, R'_2, R'_8, R'_{10}\}$. After we have determined the values of $\{R'_0, R'_2, R'_8, R'_{10}\}$ and perform Routine Computation One second time, our target is that the intermediate value after MC in Round 4 is

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

The moves of determining the values of $\{R'_0, R'_2, R'_8, R'_{10}\}$ are shown below:

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

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 the output of MC in Round 4 are

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

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 Equation (5), and we can solve (5) to get the values of T_0^* and T_2^*. In Equation (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).

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} T_0^* \\ T_1^* \\ T_2^* \\ T_3^* \end{bmatrix} = U_1 \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} T_0^* \\ T_1^* \\ T_2^* \\ T_3^* \end{bmatrix} = U_3 \end{cases} \quad (5)$$

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} T_8^* \\ T_9^* \\ T_{10}^* \\ T_{11}^* \end{bmatrix} = U_9 \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} T_8^* \\ T_9^* \\ T_{10}^* \\ T_{11}^* \end{bmatrix} = U_{11} \end{cases} \quad (6)$$

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 $\{R'_0, R'_2, R'_8, R'_{10}\}$ are computed as follows: (note that R_0^*, R_2^*, R_8^* and R_{10}^* are obtained from Buffer Two):

$$R'_0 = R_0^* \oplus \text{SB}^{-1}\left(S_0^*\right), R'_2 = R_2^* \oplus \text{SB}^{-1}\left(S_2^*\right)$$

$$R'_8 = R_8^* \oplus \text{SB}^{-1}\left(S_8^*\right), R'_{10} = R_{10}^* \oplus \text{SB}^{-1}\left(S_{10}^*\right)$$

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, $i \in \{0, 2, 8, 10\}$.

3.1.3 *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 3). 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
Initial round : $\underrightarrow{\text{ARK}}$
Round 1: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$ $\underrightarrow{\text{ARK}}$ $\underrightarrow{\oplus}$
Round 2: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$ $\underrightarrow{\text{ARK}}$ $\underrightarrow{\oplus}$
Round 3: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$ $\underrightarrow{\text{ARK}}$ $\underrightarrow{\oplus}$
Round 4: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$

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:

$$\{V'_0, V'_2, V'_8, V'_{10}\} \leftarrow \left\{W^*_0, W^*_2, W^*_8, W^*_{10}\right\}$$
$$\leftarrow \left\{X^*_0, X^*_2, X^*_8, X^*_{10}\right\} \leftarrow \{Y_1, Y_3, Y_9, Y_{11}\}$$

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

$$\begin{bmatrix} X^*_0 & X_4 & X^*_8 & X_{12} \\ X^*_1 & X_5 & X^*_9 & X_{13} \\ X^*_2 & X_6 & X^*_{10} & X_{14} \\ X^*_3 & X_7 & X^*_{11} & X_{15} \end{bmatrix} \xrightarrow{\text{MC}} \begin{bmatrix} Y^*_0 & Y_4 & Y^*_8 & Y_{12} \\ Y_1 & Y_5 & Y_9 & Y_{13} \\ Y^*_2 & Y_6 & Y^*_{10} & Y_{14} \\ Y_3 & Y_7 & Y_{11} & Y_{15} \end{bmatrix}$$

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 Equation (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).

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} X^*_0 \\ X^*_1 \\ X^*_2 \\ X^*_3 \end{bmatrix} = Y_1 \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} X^*_0 \\ X^*_1 \\ X^*_2 \\ X^*_3 \end{bmatrix} = Y_3 \end{cases} \tag{7}$$

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} X^*_8 \\ X^*_9 \\ X^*_{10} \\ X^*_{11} \end{bmatrix} = Y_9 \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} X^*_8 \\ X^*_9 \\ X^*_{10} \\ X^*_{11} \end{bmatrix} = Y_{11} \end{cases} \tag{8}$$

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):

$$V'_0 = V^*_0 \oplus \text{SB}^{-1}\left(W^*_0\right), V'_2 = V^*_2 \oplus \text{SB}^{-1}\left(W^*_2\right),$$
$$V'_8 = V^*_8 \oplus SB^{-1}\left(W^*_8\right), V'_{10} = V^*_{10} \oplus \text{SB}^{-1}\left(W^*_{10}\right).$$

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, $i \in \{0, 2, 8, 10\}$.

3.1.4 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 2)):

$$Z'_0 = Z^*_0 \oplus Z_0, Z'_2 = Z^*_2 \oplus Z_2,$$
$$Z'_8 = Z^*_8 \oplus Z_8, Z'_{10} = Z^*_{10} \oplus Z_{10}.$$

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 5-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
Initial round : $\underrightarrow{\text{ARK}}$
Round 1: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$ $\underrightarrow{\text{ARK}}$ $\underrightarrow{\oplus}$
Round 2: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$ $\underrightarrow{\text{ARK}}$ $\underrightarrow{\oplus}$
Round 3: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$ $\underrightarrow{\text{ARK}}$ $\underrightarrow{\oplus}$
Round 4: $\underrightarrow{\text{SB}}$ $\underrightarrow{\text{SR}}$ $\underrightarrow{\text{MC}}$.

Remark 2: *The most important part of the δ algorithm is solving those eight groups of linear Equations (1)–(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*

20 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'_s, 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 3, 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 20 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 20 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 20 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}. Figure 4 outlines different locations of the 20 bytes. In Figure 3, $\{G'_i, M'_i, R'_i, V'_i, Z'_i\}$ operate in Round {1, 2, 3, 4, 5}, and they can also operate in Rounds {2, 3, 4, 5, 6}, {3, 4, 5, 6, 7}, {4, 5, 6, 7, 8} or {5, 6, 7, 8, 9}. Therefore, there are 4 different combinations for the byte locations, and there are five different combinations for the round numbers in AES-128. In total, there are 20 (= 4 × 5) variants of the δ algorithm for AES-128. The δ algorithm has 28 (= 4 × 7) variants for AES-192, and 36 (= 4 × 9) variants for AES-256.

Figure 4 Different locations of the 20 bytes

0	0	0	0		0	G'_4	0	G'_{12}		0	0	0	0
G'_1	0	G'_9	0		0	0	0	0		0	G'_5	0	G'_{13}
0	0	0	0		0	G'_6	0	G'_{14}		0	0	0	0
G'_3	0	G'_{11}	0		0	0	0	0		0	G'_7	0	G'_{15}

0	0	0	0		0	M'_4	0	M'_{12}		0	0	0	0
M'_1	0	M'_9	0		0	0	0	0		0	M'_5	0	M'_{13}
0	0	0	0		0	M'_6	0	M'_{14}		0	0	0	0
M'_3	0	M'_{11}	0		0	0	0	0		0	M'_7	0	M'_{15}

0	0	0	0		0	R'_4	0	R'_{12}		0	0	0	0
R'_1	0	R'_9	0		0	0	0	0		0	R'_5	0	R'_{13}
0	0	0	0		0	R'_6	0	R'_{14}		0	0	0	0
R'_3	0	R'_{11}	0		0	0	0	0		0	R'_7	0	R'_{15}

0	0	0	0		0	V'_4	0	V'_{12}		0	0	0	0
V'_1	0	V'_9	0		0	0	0	0		0	V'_5	0	V'_{13}
0	0	0	0		0	V'_6	0	V'_{14}		0	0	0	0
V'_3	0	V'_{11}	0		0	0	0	0		0	V'_7	0	V'_{15}

0	0	0	0		0	Z'_4	0	Z'_{12}		0	0	0	0
Z'_1	0	Z'_9	0		0	0	0	0		0	Z'_5	0	Z'_{13}
0	0	0	0		0	Z'_6	0	Z'_{14}		0	0	0	0
Z'_3	0	Z'_{11}	0		0	0	0	0		0	Z'_7	0	Z'_{15}

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 3 in Section 3, we apply 20 extra XOR operations to the intermediate values after ARK in Rounds 1–5 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 1–5 (instead of with the intermediate values after ARK), and we still get the same result, $i \in \{0, 2, 8, 10\}$. There are 20-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. SB, SR, MC and ARK) 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 $C = \text{AES}_K(P)$.
2. By accepting *P* and *K* as two inputs, use the δ algorithm to generate 20 output bytes:

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

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, $i \in \{0, 2, 8, 10\}$. The details of computing the δAES round keys is described in Section 4.1.
5. After carrying out the transformations above, the δAES uses the same round function (i.e. SB, SR, MC and ARK) to process the plaintext *P* with modified AES round keys, and finally, the δAES also generates the same cipher-text *C*, denoted by $C = \delta\text{AES}(P)$. Appendix provides some examples of the AES and the AES with 20 extra exclusive-or operations.

4.1 AES round keys and δAES round keys

Suppose *K* is a 128-bit AES cipher key, and after key expansion, the AES round keys are denoted by

K_0^i	K_4^i	K_8^i	K_{12}^i
K_1^i	K_5^i	K_9^i	K_{13}^i
K_2^i	K_6^i	K_{10}^i	K_{14}^i
K_3^i	K_7^i	K_{11}^i	K_{15}^i

where i is the round number, $i \in \{1, 2, \dots, 10\}$. The round key used in the initial round is the secret key K itself, and the secret key is denoted without the superscript i.

The δAES round keys come from the following routine (see Figure 5):

1 In Initial Round, Rounds 6–10, use the corresponding AES round keys without any changes.

2 In Rounds 1–5, use the modified AES round keys. After applying 20 XOR operations to the AES round keys, the δAES round key i is calculated by the following formulas:

$$\begin{cases} K_y^i \oplus \beta, & y \in \{0, 2, 8, 10\} \\ K_y^i, & y \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15\} \end{cases}$$

where y is the byte index of the block, $i \in \{1, 2, 3, 4, 5\}$ and β is equal to G'_y, M'_y, R'_y, V'_y or Z'_y when i is equal to 1, 2, 3, 4 or 5, respectively.

Figure 5 AES round keys and the corresponding δAES round keys

AES Round Keys — The Corresponding δAES Round Keys

cipher Key K Initial Round
$$\begin{array}{|c|c|c|c|}\hline K_0 & K_4 & K_8 & K_{12} \\ \hline K_1 & K_5 & K_9 & K_{13} \\ \hline K_2 & K_6 & K_{10} & K_{14} \\ \hline K_3 & K_7 & K_{11} & K_{15} \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline K_0 & K_4 & K_8 & K_{12} \\ \hline K_1 & K_5 & K_9 & K_{13} \\ \hline K_2 & K_6 & K_{10} & K_{14} \\ \hline K_3 & K_7 & K_{11} & K_{15} \\ \hline \end{array}$$

Round Key 1
$$\begin{array}{|c|c|c|c|}\hline K_0^1 & K_4^1 & K_8^1 & K_{12}^1 \\ \hline K_1^1 & K_5^1 & K_9^1 & K_{13}^1 \\ \hline K_2^1 & K_6^1 & K_{10}^1 & K_{14}^1 \\ \hline K_3^1 & K_7^1 & K_{11}^1 & K_{15}^1 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|}\hline G'_0 & 0 & G'_8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline G'_2 & 0 & G'_{10} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline K_0^1 \oplus G'_0 & K_4^1 & K_8^1 \oplus G'_8 & K_{12}^1 \\ \hline K_1^1 & K_5^1 & K_9^1 & K_{13}^1 \\ \hline K_2^1 \oplus G'_2 & K_6^1 & K_{10}^1 \oplus G'_{10} & K_{14}^1 \\ \hline K_3^1 & K_7^1 & K_{11}^1 & K_{15}^1 \\ \hline \end{array}$$

Round Key 2
$$\begin{array}{|c|c|c|c|}\hline K_0^2 & K_4^2 & K_8^2 & K_{12}^2 \\ \hline K_1^2 & K_5^2 & K_9^2 & K_{13}^2 \\ \hline K_2^2 & K_6^2 & K_{10}^2 & K_{14}^2 \\ \hline K_3^2 & K_7^2 & K_{11}^2 & K_{15}^2 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|}\hline M'_0 & 0 & M'_8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline M'_2 & 0 & M'_{10} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline K_0^2 \oplus M'_0 & K_4^2 & K_8^2 \oplus M'_8 & K_{12}^2 \\ \hline K_1^2 & K_5^2 & K_9^2 & K_{13}^2 \\ \hline K_2^2 \oplus M'_2 & K_6^2 & K_{10}^2 \oplus M'_{10} & K_{14}^2 \\ \hline K_3^2 & K_7^2 & K_{11}^2 & K_{15}^2 \\ \hline \end{array}$$

Round Key 3
$$\begin{array}{|c|c|c|c|}\hline K_0^3 & K_4^3 & K_8^3 & K_{12}^3 \\ \hline K_1^3 & K_5^3 & K_9^3 & K_{13}^3 \\ \hline K_2^3 & K_6^3 & K_{10}^3 & K_{14}^3 \\ \hline K_3^3 & K_7^3 & K_{11}^3 & K_{15}^3 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|}\hline R'_0 & 0 & R'_8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline R'_2 & 0 & R'_{10} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline K_0^3 \oplus R'_0 & K_4^3 & K_8^3 \oplus R'_8 & K_{12}^3 \\ \hline K_1^3 & K_5^3 & K_9^3 & K_{13}^3 \\ \hline K_2^3 \oplus R'_2 & K_6^3 & K_{10}^3 \oplus R'_{10} & K_{14}^3 \\ \hline K_3^3 & K_7^3 & K_{11}^3 & K_{15}^3 \\ \hline \end{array}$$

Round Key 4
$$\begin{array}{|c|c|c|c|}\hline K_0^4 & K_4^4 & K_8^4 & K_{12}^4 \\ \hline K_1^4 & K_5^4 & K_9^4 & K_{13}^4 \\ \hline K_2^4 & K_6^4 & K_{10}^4 & K_{14}^4 \\ \hline K_3^4 & K_7^4 & K_{11}^4 & K_{15}^4 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|}\hline V'_0 & 0 & V'_8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline V'_2 & 0 & V'_{10} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline K_0^4 \oplus V'_0 & K_4^4 & K_8^4 \oplus V'_8 & K_{12}^4 \\ \hline K_1^4 & K_5^4 & K_9^4 & K_{13}^4 \\ \hline K_2^4 \oplus V'_2 & K_6^4 & K_{10}^4 \oplus V'_{10} & K_{14}^4 \\ \hline K_3^4 & K_7^4 & K_{11}^4 & K_{15}^4 \\ \hline \end{array}$$

Round Key 5
$$\begin{array}{|c|c|c|c|}\hline K_0^5 & K_4^5 & K_8^5 & K_{12}^5 \\ \hline K_1^5 & K_5^5 & K_9^5 & K_{13}^5 \\ \hline K_2^5 & K_6^5 & K_{10}^5 & K_{14}^5 \\ \hline K_3^5 & K_7^5 & K_{11}^5 & K_{15}^5 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|}\hline Z'_0 & 0 & Z'_8 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline Z'_2 & 0 & Z'_{10} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline K_0^5 \oplus Z'_0 & K_4^5 & K_8^5 \oplus Z'_8 & K_{12}^5 \\ \hline K_1^5 & K_5^5 & K_9^5 & K_{13}^5 \\ \hline K_2^5 \oplus Z'_2 & K_6^5 & K_{10}^5 \oplus Z'_{10} & K_{14}^5 \\ \hline K_3^5 & K_7^5 & K_{11}^5 & K_{15}^5 \\ \hline \end{array}$$

Round Key 6
$$\begin{array}{|c|c|c|c|}\hline K_0^6 & K_4^6 & K_8^6 & K_{12}^6 \\ \hline K_1^6 & K_5^6 & K_9^6 & K_{13}^6 \\ \hline K_2^6 & K_6^6 & K_{10}^6 & K_{14}^6 \\ \hline K_3^6 & K_7^6 & K_{11}^6 & K_{15}^6 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline K_0^6 & K_4^6 & K_8^6 & K_{12}^6 \\ \hline K_1^6 & K_5^6 & K_9^6 & K_{13}^6 \\ \hline K_2^6 & K_6^6 & K_{10}^6 & K_{14}^6 \\ \hline K_3^6 & K_7^6 & K_{11}^6 & K_{15}^6 \\ \hline \end{array}$$

Round Key 7
$$\begin{array}{|c|c|c|c|}\hline K_0^7 & K_4^7 & K_8^7 & K_{12}^7 \\ \hline K_1^7 & K_5^7 & K_9^7 & K_{13}^7 \\ \hline K_2^7 & K_6^7 & K_{10}^7 & K_{14}^7 \\ \hline K_3^7 & K_7^7 & K_{11}^7 & K_{15}^7 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline K_0^7 & K_4^7 & K_8^7 & K_{12}^7 \\ \hline K_1^7 & K_5^7 & K_9^7 & K_{13}^7 \\ \hline K_2^7 & K_6^7 & K_{10}^7 & K_{14}^7 \\ \hline K_3^7 & K_7^7 & K_{11}^7 & K_{15}^7 \\ \hline \end{array}$$

Round Key 8
$$\begin{array}{|c|c|c|c|}\hline K_0^8 & K_4^8 & K_8^8 & K_{12}^8 \\ \hline K_1^8 & K_5^8 & K_9^8 & K_{13}^8 \\ \hline K_2^8 & K_6^8 & K_{10}^8 & K_{14}^8 \\ \hline K_3^8 & K_7^8 & K_{11}^8 & K_{15}^8 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline K_0^8 & K_4^8 & K_8^8 & K_{12}^8 \\ \hline K_1^8 & K_5^8 & K_9^8 & K_{13}^8 \\ \hline K_2^8 & K_6^8 & K_{10}^8 & K_{14}^8 \\ \hline K_3^8 & K_7^8 & K_{11}^8 & K_{15}^8 \\ \hline \end{array}$$

Round Key 9
$$\begin{array}{|c|c|c|c|}\hline K_0^9 & K_4^9 & K_8^9 & K_{12}^9 \\ \hline K_1^9 & K_5^9 & K_9^9 & K_{13}^9 \\ \hline K_2^9 & K_6^9 & K_{10}^9 & K_{14}^9 \\ \hline K_3^9 & K_7^9 & K_{11}^9 & K_{15}^9 \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline K_0^9 & K_4^9 & K_8^9 & K_{12}^9 \\ \hline K_1^9 & K_5^9 & K_9^9 & K_{13}^9 \\ \hline K_2^9 & K_6^9 & K_{10}^9 & K_{14}^9 \\ \hline K_3^9 & K_7^9 & K_{11}^9 & K_{15}^9 \\ \hline \end{array}$$

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

Compared with the AES algorithm, the δAES needs to do some extra transformations, that is, 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 $\{G_i', M_i', R_i', V_i', Z_i'\} \in \{0, 2, 8, 10\}$.

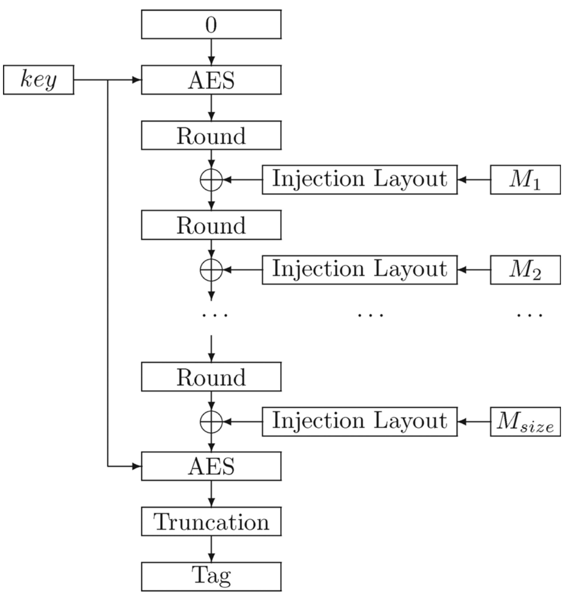
5 Description of the ALPHA-MAC

ALPHA-MAC is a MAC function which uses the building blocks of AES. Similarly to AES, the ALPHA-MAC supports keys of 128, 192 and 256 bits. The word length is 32 bits, and the injection layout places the 4 bytes of each message word [m_0, m_1, m_2, m_3] into a 4×4 array. The format of the injection layout is shown as follows:

$$\begin{bmatrix} m_0 & 0 & m_1 & 0 \\ 0 & 0 & 0 & 0 \\ m_2 & 0 & m_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Like AES, the ALPHA-MAC round function contains SB, SR, MC and ARK, and the output of each injection layout acts as the corresponding 128-bit round key. The message padding method appends a single 1 followed by the minimum number of 0 bits such that the length of the result is a multiple of 32. In the initialisation, the state is set to all zeros and AES is applied to the state. For every message word, the chaining method carries out an iteration, and each iteration maps the bits of the message word to an injection input. After that, a sequence of AES round functions are applied to the state, with the round keys replaced by the injection input. In the final transformation, AES is applied to the state. The MAC tag is the first l_m bits of the resulting final state. The length of l_m may have any value less than or equal to 128. The ALPHA-MAC function is depicted in Figure 6.

Figure 6 ALPHA-MAC construction



6 Applying the property to ALPHA-MAC

We study the internal structure of the ALPHA-MAC by employing the proposed five-round algebraic property of AES, which is described in Section 3. Firstly, we present a method to find second preimages of the ALPHA-MAC by solving eight groups of linear functions, based on the assumption that an authentication key or an intermediate value of this MAC is known. Each of these eight groups of linear functions contains two equations. We divide the second-preimage search algorithm into two steps: the backwards-and-forwards (BNF) search and the backwards-and-backwards (BNB) search. The BNF search provides an idea for extending 32- to 128-bit collisions[2] by solving four groups of linear functions. Given a key (or an intermediate value) and one four-block message, the BNB search can generate another four-block message such that these two messages produce 32-bit collisions, which are a prerequisite for the BNF search. To do the BNB search, we need to solve another four groups of linear functions. By combining the BNB search with the BNF search, we can find second preimages of ALPHA-MAC. Secondly, we show that the second-preimage finding method can also be used to generate internal collisions. The proposed collision search method can find two five-block messages such that they produce 128-bit collisions under a selected key (or a selected intermediate value).

6.1 The second-preimage search algorithm

The second-preimage search algorithm aims to find a five-block second-preimage $\tilde{M}$ for a selected five-block message M, under a selected key (or a selected intermediate value). The assumption of this search is that we know two values: a selected key (or a selected intermediate value) and a selected five-block message M. The result of the search is that M and $\tilde{M}$ generate the same 128-bit value after five rounds of ALPHA-MAC iterations, under the selected key (or the selected intermediate value).

We use Figure 7 to illustrate the second-preimage search. Figure 7 depicts five consecutive rounds of the ALPHA-MAC for two different five-block messages M and $\tilde{M}$. We assume that we are able to select an intermediate value of the round functions in some round (e.g. in Round $y-3$), and select five consecutive message blocks $M(M_{y-3}, M_{y-2}, M_{y-1}, M_y, M_{y+1})$. Then we can find another five-block message $\tilde{M}(\tilde{M}_{y-3}, \tilde{M}_{y-2}, \tilde{M}_{y-1}, \tilde{M}_y, \tilde{M}_{y+1})$ such that these two five-block messages collide on 128 bits in Round $y+1$ after ARK. Note that the intermediate value is:

$$\begin{bmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{bmatrix}$$

Figure 7 The five-block collisions

Round $y-3$:

$$\begin{array}{|c|c|c|c|}\hline a_0 & a_4 & a_8 & a_{12} \\ \hline a_1 & a_5 & a_9 & a_{13} \\ \hline a_2 & a_6 & a_{10} & a_{14} \\ \hline a_3 & a_7 & a_{11} & a_{15} \\ \hline \end{array} \xrightarrow{SB\circ SR} \begin{array}{|c|c|c|c|}\hline b_0 & b_4 & b_8 & b_{12} \\ \hline b_1 & b_5 & b_9 & b_{13} \\ \hline b_2 & b_6 & b_{10} & b_{14} \\ \hline b_3 & b_7 & b_{11} & b_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline d_0 & d_4 & d_8 & d_{12} \\ \hline d_1 & d_5 & d_9 & d_{13} \\ \hline d_2 & d_6 & d_{10} & d_{14} \\ \hline d_3 & d_7 & d_{11} & d_{15} \\ \hline \end{array} \xrightarrow[ARK]{(M_{y-3})} \begin{array}{|c|c|c|c|}\hline d_0^* & d_4 & d_8^* & d_{12} \\ \hline d_1 & d_5 & d_9 & d_{13} \\ \hline d_2^* & d_6 & d_{10}^* & d_{14} \\ \hline d_3 & d_7 & d_{11} & d_{15} \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline \tilde{a}_0 & \tilde{a}_4 & \tilde{a}_8 & \tilde{a}_{12} \\ \hline \tilde{a}_1 & \tilde{a}_5 & \tilde{a}_9 & \tilde{a}_{13} \\ \hline \tilde{a}_2 & \tilde{a}_6 & \tilde{a}_{10} & \tilde{a}_{14} \\ \hline \tilde{a}_3 & \tilde{a}_7 & \tilde{a}_{11} & \tilde{a}_{15} \\ \hline \end{array} \xrightarrow{SB\circ SR} \begin{array}{|c|c|c|c|}\hline \tilde{b}_0 & \tilde{b}_4 & \tilde{b}_8 & \tilde{b}_{12} \\ \hline \tilde{b}_1 & \tilde{b}_5 & \tilde{b}_9 & \tilde{b}_{13} \\ \hline \tilde{b}_2 & \tilde{b}_6 & \tilde{b}_{10} & \tilde{b}_{14} \\ \hline \tilde{b}_3 & \tilde{b}_7 & \tilde{b}_{11} & \tilde{b}_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline \tilde{d}_0 & \tilde{d}_4 & \tilde{d}_8 & \tilde{d}_{12} \\ \hline \tilde{d}_1 & \tilde{d}_5 & \tilde{d}_9 & \tilde{d}_{13} \\ \hline \tilde{d}_2 & \tilde{d}_6 & \tilde{d}_{10} & \tilde{d}_{14} \\ \hline \tilde{d}_3 & \tilde{d}_7 & \tilde{d}_{11} & \tilde{d}_{15} \\ \hline \end{array} \xrightarrow[ARK]{(\tilde{M}_{y-3})} \begin{array}{|c|c|c|c|}\hline \tilde{d}_0^* & \tilde{d}_4 & \tilde{d}_8^* & \tilde{d}_{12} \\ \hline \tilde{d}_1 & \tilde{d}_5 & \tilde{d}_9 & \tilde{d}_{13} \\ \hline \tilde{d}_2^* & \tilde{d}_6 & \tilde{d}_{10}^* & \tilde{d}_{14} \\ \hline \tilde{d}_3 & \tilde{d}_7 & \tilde{d}_{11} & \tilde{d}_{15} \\ \hline \end{array}$$

Round $y-2$:

$$\begin{array}{|c|c|c|c|}\hline d_0^* & d_4 & d_8^* & d_{12} \\ \hline d_1 & d_5 & d_9 & d_{13} \\ \hline d_2^* & d_6 & d_{10}^* & d_{14} \\ \hline d_3 & d_7 & d_{11} & d_{15} \\ \hline \end{array} \xrightarrow{SB\circ SR} \begin{array}{|c|c|c|c|}\hline f_0 & f_4 & f_8 & f_{12} \\ \hline f_1 & f_5 & f_9 & f_{13} \\ \hline f_2 & f_6 & f_{10} & f_{14} \\ \hline f_3 & f_7 & f_{11} & f_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline g_0 & g_4 & g_8 & g_{12} \\ \hline g_1 & g_5 & g_9 & g_{13} \\ \hline g_2 & g_6 & g_{10} & g_{14} \\ \hline g_3 & g_7 & g_{11} & g_{15} \\ \hline \end{array} \xrightarrow[ARK]{(M_{y-2})} \begin{array}{|c|c|c|c|}\hline g_0^* & g_4 & g_8^* & g_{12} \\ \hline g_1 & g_5 & g_9 & g_{13} \\ \hline g_2^* & g_6 & g_{10}^* & g_{14} \\ \hline g_3 & g_7 & g_{11} & g_{15} \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline \tilde{d}_0^* & \tilde{d}_4 & \tilde{d}_8^* & \tilde{d}_{12} \\ \hline \tilde{d}_1 & \tilde{d}_5 & \tilde{d}_9 & \tilde{d}_{13} \\ \hline \tilde{d}_2^* & \tilde{d}_6 & \tilde{d}_{10}^* & \tilde{d}_{14} \\ \hline \tilde{d}_3 & \tilde{d}_7 & \tilde{d}_{11} & \tilde{d}_{15} \\ \hline \end{array} \xrightarrow{SB\circ SR} \begin{array}{|c|c|c|c|}\hline \tilde{f}_0 & \tilde{f}_4 & \tilde{f}_8 & \tilde{f}_{12} \\ \hline \tilde{f}_1 & \tilde{f}_5 & \tilde{f}_9 & \tilde{f}_{13} \\ \hline \tilde{f}_2 & \tilde{f}_6 & \tilde{f}_{10} & \tilde{f}_{14} \\ \hline \tilde{f}_3 & \tilde{f}_7 & \tilde{f}_{11} & \tilde{f}_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline \tilde{g}_0 & \tilde{g}_4 & \tilde{g}_8 & \tilde{g}_{12} \\ \hline \tilde{g}_1 & \tilde{g}_5 & \tilde{g}_9 & \tilde{g}_{13} \\ \hline \tilde{g}_2 & \tilde{g}_6 & \tilde{g}_{10} & \tilde{g}_{14} \\ \hline \tilde{g}_3 & \tilde{g}_7 & \tilde{g}_{11} & \tilde{g}_{15} \\ \hline \end{array} \xrightarrow[ARK]{(\tilde{M}_{y-2})} \begin{array}{|c|c|c|c|}\hline \tilde{g}_0^* & \tilde{g}_4 & \tilde{g}_8^* & \tilde{g}_{12} \\ \hline \tilde{g}_1 & \tilde{g}_5 & \tilde{g}_9 & \tilde{g}_{13} \\ \hline \tilde{g}_2^* & \tilde{g}_6 & \tilde{g}_{10}^* & \tilde{g}_{14} \\ \hline \tilde{g}_3 & \tilde{g}_7 & \tilde{g}_{11} & \tilde{g}_{15} \\ \hline \end{array}$$

Round $y-1$:

$$\begin{array}{|c|c|c|c|}\hline g_0^* & g_4 & g_8^* & g_{12} \\ \hline g_1 & g_5 & g_9 & g_{13} \\ \hline g_2^* & g_6 & g_{10}^* & g_{14} \\ \hline g_3 & g_7 & g_{11} & g_{15} \\ \hline \end{array} \xrightarrow{SB\circ SR} \begin{array}{|c|c|c|c|}\hline h_0 & h_4 & h_8 & h_{12} \\ \hline h_1 & h_5 & h_9 & h_{13} \\ \hline h_2 & h_6 & h_{10} & h_{14} \\ \hline h_3 & h_7 & h_{11} & h_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline i_0 & i_4 & i_8 & i_{12} \\ \hline i_1 & i_5 & i_9 & i_{13} \\ \hline i_2 & i_6 & i_{10} & i_{14} \\ \hline i_3 & i_7 & i_{11} & i_{15} \\ \hline \end{array} \xrightarrow[ARK]{(M_{y-1})} \begin{array}{|c|c|c|c|}\hline i_0^* & i_4 & i_8^* & i_{12} \\ \hline i_1 & i_5 & i_9 & i_{13} \\ \hline i_2^* & i_6 & i_{10}^* & i_{14} \\ \hline i_3 & i_7 & i_{11} & i_{15} \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline \tilde{g}_0^* & \tilde{g}_4 & \tilde{g}_8^* & \tilde{g}_{12} \\ \hline \tilde{g}_1 & \tilde{g}_5 & \tilde{g}_9 & \tilde{g}_{13} \\ \hline \tilde{g}_2^* & \tilde{g}_6 & \tilde{g}_{10}^* & \tilde{g}_{14} \\ \hline \tilde{g}_3 & \tilde{g}_7 & \tilde{g}_{11} & \tilde{g}_{15} \\ \hline \end{array} \xrightarrow{SB\circ SR} \begin{array}{|c|c|c|c|}\hline \tilde{h}_0 & \tilde{h}_4 & \tilde{h}_8 & \tilde{h}_{12} \\ \hline \tilde{h}_1 & \tilde{h}_5 & \tilde{h}_9 & \tilde{h}_{13} \\ \hline \tilde{h}_2 & \tilde{h}_6 & \tilde{h}_{10} & \tilde{h}_{14} \\ \hline \tilde{h}_3 & \tilde{h}_7 & \tilde{h}_{11} & \tilde{h}_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline \tilde{i}_0 & \tilde{i}_4 & \tilde{i}_8 & \tilde{i}_{12} \\ \hline \tilde{i}_1 & \tilde{i}_5 & \tilde{i}_9 & \tilde{i}_{13} \\ \hline \tilde{i}_2 & \tilde{i}_6 & \tilde{i}_{10} & \tilde{i}_{14} \\ \hline \tilde{i}_3 & \tilde{i}_7 & \tilde{i}_{11} & \tilde{i}_{15} \\ \hline \end{array} \xrightarrow[ARK]{(\tilde{M}_{y-1})} \begin{array}{|c|c|c|c|}\hline \tilde{i}_0^* & \tilde{i}_4 & \tilde{i}_8^* & \tilde{i}_{12} \\ \hline \tilde{i}_1 & \tilde{i}_5 & \tilde{i}_9 & \tilde{i}_{13} \\ \hline \tilde{i}_2^* & \tilde{i}_6 & \tilde{i}_{10}^* & \tilde{i}_{14} \\ \hline \tilde{i}_3 & \tilde{i}_7 & \tilde{i}_{11} & \tilde{i}_{15} \\ \hline \end{array}$$

Round y:

$$\begin{array}{|c|c|c|c|}\hline i_0^* & i_4 & i_8^* & i_{12} \\ \hline i_1 & i_5 & i_9 & i_{13} \\ \hline i_2^* & i_6 & i_{10}^* & i_{14} \\ \hline i_3 & i_7 & i_{11} & i_{15} \\ \hline \end{array} \xrightarrow{SB\circ SR} \begin{array}{|c|c|c|c|}\hline j_0 & j_4 & j_8 & j_{12} \\ \hline j_1 & j_5 & j_9 & j_{13} \\ \hline j_2 & j_6 & j_{10} & j_{14} \\ \hline j_3 & j_7 & j_{11} & j_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline s_0 & s_4 & s_8 & s_{12} \\ \hline s_1 & s_5 & s_9 & s_{13} \\ \hline s_2 & s_6 & s_{10} & s_{14} \\ \hline s_3 & s_7 & s_{11} & s_{15} \\ \hline \end{array} \xrightarrow[ARK]{(M_{y})} \begin{array}{|c|c|c|c|}\hline s_0^* & s_4 & s_8^* & s_{12} \\ \hline s_1 & s_5 & s_9 & s_{13} \\ \hline s_2^* & s_6 & s_{10}^* & s_{14} \\ \hline s_3 & s_7 & s_{11} & s_{15} \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline \tilde{i}_0^* & \tilde{i}_4 & \tilde{i}_8^* & \tilde{i}_{12} \\ \hline \tilde{i}_1 & \tilde{i}_5 & \tilde{i}_9 & \tilde{i}_{13} \\ \hline \tilde{i}_2^* & \tilde{i}_6 & \tilde{i}_{10}^* & \tilde{i}_{14} \\ \hline \tilde{i}_3 & \tilde{i}_7 & \tilde{i}_{11} & \tilde{i}_{15} \\ \hline \end{array} \xrightarrow{SB\circ SR} \begin{array}{|c|c|c|c|}\hline \tilde{j}_0 & \tilde{j}_4 & \tilde{j}_8 & \tilde{j}_{12} \\ \hline \tilde{j}_1 & \tilde{j}_5 & \tilde{j}_9 & \tilde{j}_{13} \\ \hline \tilde{j}_2 & \tilde{j}_6 & \tilde{j}_{10} & \tilde{j}_{14} \\ \hline \tilde{j}_3 & \tilde{j}_7 & \tilde{j}_{11} & \tilde{j}_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline \tilde{s}_0 & \tilde{s}_4 & \tilde{s}_8 & \tilde{s}_{12} \\ \hline \tilde{s}_1 & \tilde{s}_5 & \tilde{s}_9 & \tilde{s}_{13} \\ \hline \tilde{s}_2 & \tilde{s}_6 & \tilde{s}_{10} & \tilde{s}_{14} \\ \hline \tilde{s}_3 & \tilde{s}_7 & \tilde{s}_{11} & \tilde{s}_{15} \\ \hline \end{array} \xrightarrow[ARK]{(\tilde{M}_{y})} \begin{array}{|c|c|c|c|}\hline \tilde{s}_0^* & \tilde{s}_4 & \tilde{s}_8^* & \tilde{s}_{12} \\ \hline \tilde{s}_1 & \tilde{s}_5 & \tilde{s}_9 & \tilde{s}_{13} \\ \hline \tilde{s}_2^* & \tilde{s}_6 & \tilde{s}_{10}^* & \tilde{s}_{14} \\ \hline \tilde{s}_3 & \tilde{s}_7 & \tilde{s}_{11} & \tilde{s}_{15} \\ \hline \end{array}$$

Round $y+1$:

$$\begin{array}{|c|c|c|c|}\hline s_0^* & s_4 & s_8^* & s_{12} \\ \hline s_1 & s_5 & s_9 & s_{13} \\ \hline s_2^* & s_6 & s_{10}^* & s_{14} \\ \hline s_3 & s_7 & s_{11} & s_{15} \\ \hline \end{array} \xrightarrow{SB\circ SR} \begin{array}{|c|c|c|c|}\hline n_0 & n_4 & n_8 & n_{12} \\ \hline n_1 & n_5 & n_9 & n_{13} \\ \hline n_2 & n_6 & n_{10} & n_{14} \\ \hline n_3 & n_7 & n_{11} & n_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline w_0 & w_4 & w_8 & w_{12} \\ \hline w_1 & w_5 & w_9 & w_{13} \\ \hline w_2 & w_6 & w_{10} & w_{14} \\ \hline w_3 & w_7 & w_{11} & w_{15} \\ \hline \end{array} \xrightarrow[ARK]{(M_{y+1})} \begin{array}{|c|c|c|c|}\hline w_0^* & w_4 & w_8^* & w_{12} \\ \hline w_1 & w_5 & w_9 & w_{13} \\ \hline w_2^* & w_6 & w_{10}^* & w_{14} \\ \hline w_3 & w_7 & w_{11} & w_{15} \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline \tilde{s}_0^* & \tilde{s}_4 & \tilde{s}_8^* & \tilde{s}_{12} \\ \hline \tilde{s}_1 & \tilde{s}_5 & \tilde{s}_9 & \tilde{s}_{13} \\ \hline \tilde{s}_2^* & \tilde{s}_6 & \tilde{s}_{10}^* & \tilde{s}_{14} \\ \hline \tilde{s}_3 & \tilde{s}_7 & \tilde{s}_{11} & \tilde{s}_{15} \\ \hline \end{array} \xrightarrow{SB\circ SR} \begin{array}{|c|c|c|c|}\hline \tilde{n}_0 & \tilde{n}_4 & \tilde{n}_8 & \tilde{n}_{12} \\ \hline \tilde{n}_1 & \tilde{n}_5 & \tilde{n}_9 & \tilde{n}_{13} \\ \hline \tilde{n}_2 & \tilde{n}_6 & \tilde{n}_{10} & \tilde{n}_{14} \\ \hline \tilde{n}_3 & \tilde{n}_7 & \tilde{n}_{11} & \tilde{n}_{15} \\ \hline \end{array} \xrightarrow{MC} \begin{array}{|c|c|c|c|}\hline \tilde{w}_0 & \tilde{w}_4 & \tilde{w}_8 & \tilde{w}_{12} \\ \hline \tilde{w}_1 & \tilde{w}_5 & \tilde{w}_9 & \tilde{w}_{13} \\ \hline \tilde{w}_2 & \tilde{w}_6 & \tilde{w}_{10} & \tilde{w}_{14} \\ \hline \tilde{w}_3 & \tilde{w}_7 & \tilde{w}_{11} & \tilde{w}_{15} \\ \hline \end{array} \xrightarrow[ARK]{(\tilde{M}_{y+1})} \begin{array}{|c|c|c|c|}\hline \tilde{w}_0^* & \tilde{w}_4 & \tilde{w}_8^* & \tilde{w}_{12} \\ \hline \tilde{w}_1 & \tilde{w}_5 & \tilde{w}_9 & \tilde{w}_{13} \\ \hline \tilde{w}_2^* & \tilde{w}_6 & \tilde{w}_{10}^* & \tilde{w}_{14} \\ \hline \tilde{w}_3 & \tilde{w}_7 & \tilde{w}_{11} & \tilde{w}_{15} \\ \hline \end{array}$$

In the case of a selected key, for the sake of simplicity, we assume that $(M_{y-3}, M_{y-2}, M_{y-1}, M_y, M_{y+1})$ are the first five blocks of the selected message. Our search algorithm works without assuming that $(M_{y-3}, M_{y-2}, M_{y-1}, M_y, M_{y+1})$ are the first five blocks of the selected message.

The second-preimage search algorithm has the following form:

Known: 1 A selected key or a selected intermediate value.

2 A selected five-block message $M(M_{y-3}, M_{y-2}, M_{y-1}, M_y, M_{y+1})$

Find: Another five-block message $\tilde{M}(\tilde{M}_{y-3}, \tilde{M}_{y-2}, \tilde{M}_{y-1}, \tilde{M}_y, \tilde{M}_{y+1})$ such that M and $\tilde{M}$ collide on 128 bits after ARK in Round $y + 1$.

Method: Solve eight groups of linear functions. These eight groups of functions are named as (9)–(16) in this section.

The second-preimage search algorithm consists of two steps: the BNF search and the BNB search. The BNF search can extend 32- to 128-bit collisions, given two messages M and $\tilde{M}$ which collide on 32 bits, namely Bytes s_4, s_{12}, s_6 and s_{14}, after MC in Round y (see Figure 7). Given a key (or an intermediate value) and one four-block message, the BNB search is able to find another four-block message such that these two messages collide on Bytes s_4, s_{12}, s_6 and s_{14} after MC in Round y. The BNB search generates those 32-bit collisions which are required for the BNF search. By merging the BNB search with the BNF search, we can find second preimages of the ALPHA-MAC.

6.1.1 The BNF search

The BNF search has the following form:

Known: 1 A selected key or a selected intermediate value.

2 Two four-block messages $M(M_{y-3}, M_{y-2}, M_{y-1}, M_y, M_{y+1})$ and $\tilde{M}(\tilde{M}_{y-3}, \tilde{M}_{y-2}, \tilde{M}_{y-1}, \tilde{M}_y, \tilde{M}_{y+1})$ colliding on 32 bits (Bytes s_4, s_{12}, s_6 and s_{14}) after MC in Round y.

Extend: 32-bit collisions to 128-bit collisions in Round $y + 1$.

Method: Solve four groups of linear functions. These four groups of functions are numbered as (9)–(12) in this section.

The BNF search assumes that we are able to find two messages M and $\tilde{M}$, which collide on Bytes s_4, s_{12}, s_6 and s_{14} after MC in Round y. Based on the algebraic property of the MC transformation and the structure of ALPHA-MAC, we can extend these 32- to 128-bit collisions within three rounds by solving four groups of linear equations.

6.1.2 Extending 32- to 64-bit collisions

We use the differential XOR property before and after the MC transformation. In Round y before MC, by XORing those two intermediate values, we get the following result:

$$\begin{bmatrix} \tilde{j}_0 \oplus j_0 & \tilde{j}_4 \oplus j_4 & \tilde{j}_8 \oplus j_8 & \tilde{j}_{12} \oplus j_{12} \\ \tilde{j}_1 \oplus j_1 & \tilde{j}_5 \oplus j_5 & \tilde{j}_9 \oplus j_9 & \tilde{j}_{13} \oplus j_{13} \\ \tilde{j}_2 \oplus j_2 & \tilde{j}_6 \oplus j_6 & \tilde{j}_{10} \oplus j_{10} & \tilde{j}_{14} \oplus j_{14} \\ \tilde{j}_3 \oplus j_3 & \tilde{j}_7 \oplus j_7 & \tilde{j}_{11} \oplus j_{11} & \tilde{j}_{15} \oplus j_{15} \end{bmatrix} \xrightarrow{\text{MC}}$$

$$\begin{bmatrix} ? & 0 & ? & 0 \\ 0 & \tilde{s}_5 \oplus s_5 & 0 & \tilde{s}_{13} \oplus s_{13} \\ ? & 0 & ? & 0 \\ 0 & \tilde{s}_7 \oplus s_7 & 0 & \tilde{s}_{15} \oplus s_{15} \end{bmatrix}$$

Here, we use R (to replace $\tilde{j}_0 \oplus j_0$), S (to replace $\tilde{j}_8 \oplus j_8$), T (to replace $\tilde{j}_2 \oplus j_2$) and U (to replace $\tilde{j}_{10} \oplus j_{10}$) so that after the MC transformation in Round y, Bytes $\tilde{s}_1 \oplus s_1, \tilde{s}_3 \oplus \tilde{s}_3, \tilde{s}_9 \oplus s_9$ and $\tilde{s}_{11} \oplus s_{11}$ become zero. Now the question is 'how to decide R, S, T and U'. The answer is:

- there exists one and only one pair of (R, T) such that after MC, Bytes $\tilde{s}_1 \oplus s_1$ and $\tilde{s}_3 \oplus \tilde{s}_3$ are both zero
- there exists one and only one pair of (S, U) such that after MC, $\tilde{s}_9 \oplus s_9$ and $\tilde{s}_{11} \oplus s_{11}$ are both zero.

According to the MC transformation, we have the following formula:

$$\begin{bmatrix} ? & 0 & ? & 0 \\ 0 & \tilde{s}_5 \oplus s_5 & 0 & \tilde{s}_{13} \oplus s_{13} \\ ? & 0 & ? & 0 \\ 0 & \tilde{s}_7 \oplus s_7 & 0 & \tilde{s}_{15} \oplus s_{15} \end{bmatrix} \xrightarrow{\text{MC}}$$

$$\begin{bmatrix} R & \tilde{j}_4 \oplus j_4 & S & \tilde{j}_{12} \oplus j_{12} \\ \tilde{j}_1 \oplus j_1 & \tilde{j}_5 \oplus j_5 & \tilde{j}_9 \oplus j_9 & \tilde{j}_{13} \oplus j_{13} \\ T & \tilde{j}_6 \oplus j_6 & U & \tilde{j}_{14} \oplus j_{13} \\ \tilde{j}_3 \oplus j_3 & \tilde{j}_7 \oplus j_7 & \tilde{j}_{11} \oplus j_{11} & \tilde{j}_{15} \oplus j_{15} \end{bmatrix}$$

To find out the values of (R, T) and (S, U), we need to solve the following two groups of equations.

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} R \\ \tilde{j}_1 \oplus j_1 \\ T \\ \tilde{j}_3 \oplus j_3 \end{bmatrix} = 0 \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} R \\ \tilde{j}_1 \oplus j_1 \\ T \\ \tilde{j}_3 \oplus j_3 \end{bmatrix} = 0 \end{cases} \tag{9}$$

$$\begin{cases} [01\ \ 02\ \ 03\ \ 01]\begin{bmatrix} S \\ \tilde{j}_9 \oplus j_9 \\ U \\ \tilde{j}_{11} \oplus j_{11} \end{bmatrix} = 0 \\ [03\ \ 01\ \ 01\ \ 02]\begin{bmatrix} S \\ \tilde{j}_9 \oplus j_9 \\ U \\ \tilde{j}_{11} \oplus j_{11} \end{bmatrix} = 0 \end{cases} \tag{10}$$

In the two equations in (9), there are two variables R and T, and therefore there exists one and only one pair of (R, T) to make these two equations hold simultaneously. Similarly, we can decide the values of S and U by solving the two equations in (10).

Once we get the values of R, S, T and U, message block $\tilde{M}_{y-1}$ can be constructed as follows:

1 Set the values of $\tilde{j}_0^{\text{new}}, \tilde{j}_8^{\text{new}}, \tilde{j}_2^{\text{new}}, \tilde{j}_{10}^{\text{new}}$, as follows: $\tilde{j}_{10}^{\text{new}} = j_0 \oplus R, \tilde{j}_8^{\text{new}} = j_8 \oplus S, \tilde{j}_2^{\text{new}} = \tilde{j}_2 \oplus T$ and $\tilde{j}_{10}^{\text{new}} = j_{10} \oplus U$. Use $\tilde{j}_0^{\text{new}}$ to replace $\tilde{j}_0$, $\tilde{j}_8^{\text{new}}$ to replace $\tilde{j}_8$, $\tilde{j}_2^{\text{new}}$ to replace $\tilde{j}_2$ and $\tilde{j}_{10}^{\text{new}}$ to replace $\tilde{j}_{10}$.

2 Perform SR^{-1} (inverse SR) and SB^{-1} (inverse SB). As SR^{-1} and SB^{-1} are permutation and substitution, they do not change the properties we have found. Now we have the outputs of ARK in Round $y-1$.

3 Compute the value of $\tilde{M}_{y-1}^{\text{new}}$ as follows:

$$\tilde{M}_{y-1}^{\text{new}} = \left(\tilde{j}_0^{\text{new}} \oplus \tilde{i}_0\right) \left\| \left(\tilde{j}_8^{\text{new}} \oplus \tilde{i}_8\right) \right. \\ \left\| \left(\tilde{j}_{10}^{\text{new}} \oplus \tilde{i}_2\right) \right\| \left(\tilde{j}_2^{\text{new}} \oplus \tilde{i}_{10}\right)$$

Use $\tilde{M}_{y-1}^{\text{new}}$ to replace $\tilde{M}_{y-1}$.

At this stage, two messages $(M_{y-3}, M_{y-2}, M_{y-1})$ and $(\tilde{M}_{y-3}, \tilde{M}_{y-2}, \tilde{M}_{y-1}^{\text{new}})$ collide on 64 bits (Bytes s_4, s_{12}, s_6, s_{14}, s_1, s_9, s_3 and s_{11}) in Round y after MC.

6.1.3 Extending 64- to 96-bit collisions

We only need to focus on Rounds y and $y+1$ to extend 64- to 96-bit collisions. The idea is to choose message block $\tilde{M}_y$ to cancel out the differences between Bytes (s_5, s_{13}, s_7, s_{15}) and Bytes $(\tilde{s}_5, \tilde{s}_{13}, \tilde{s}_7, \tilde{s}_{15})$ in Round y. The method of choosing $\tilde{M}_y$ is exactly same as the method for constructing $\tilde{M}_{y-1}$ in Section 6.1.2.

By taking the outputs of ARK in Round y, we perform the SB and SR operations, and then XOR the results after SB and SR:

$$\begin{bmatrix} n_0 & n_4 & n_8 & n_{12} \\ n_1 & n_5 & n_9 & n_{13} \\ n_2 & n_6 & n_{10} & n_{14} \\ n_3 & n_7 & n_{11} & n_{15} \end{bmatrix} \oplus \begin{bmatrix} \tilde{n}_0 & n_4 & \tilde{n}_8 & n_{12} \\ \tilde{n}_1 & n_5 & \tilde{n}_9 & n_{13} \\ \tilde{n}_2 & n_6 & \tilde{n}_{10} & n_{14} \\ \tilde{n}_3 & n_7 & \tilde{n}_{11} & n_{15} \end{bmatrix}$$

$$= \begin{bmatrix} n_0 \oplus \tilde{n}_0 & 0 & n_8 \oplus \tilde{n}_8 & 0 \\ n_1 \oplus \tilde{n}_1 & 0 & n_9 \oplus \tilde{n}_9 & 0 \\ n_2 \oplus \tilde{n}_2 & 0 & n_{10} \oplus \tilde{n}_{10} & 0 \\ n_3 \oplus \tilde{n}_3 & 0 & n_{11} \oplus \tilde{n}_{11} & 0 \end{bmatrix} \xrightarrow{\text{MC}} \begin{bmatrix} ? & 0 & ? & 0 \\ 0 & 0 & 0 & 0 \\ ? & 0 & ? & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Here, we use π to replace $n_o \oplus \tilde{n}_0$, ρ to replace $n_8 \oplus \tilde{n}_8$, ϕ to replace $n_2 \oplus \tilde{n}_2$ and ω to replace $n_{10} \oplus \tilde{n}_{10}$ so that after MC in Round $y+1$, Bytes $w_1 \oplus \tilde{w}_1$, $w_9 \oplus \tilde{w}_9, w_3 \oplus \tilde{w}_3$ and $w_{11} \oplus \tilde{w}_{11}$. are zero:

$$\begin{bmatrix} \pi & 0 & \rho & 0 \\ n_1 \oplus \tilde{n}_1 & 0 & n_9 \oplus \tilde{n}_9 & 0 \\ \phi & 0 & \omega & 0 \\ n_3 \oplus \tilde{n}_3 & 0 & n_{11} \oplus \tilde{n}_{11} & 0 \end{bmatrix} \xrightarrow{\text{MC}} \begin{bmatrix} ? & 0 & ? & 0 \\ 0 & 0 & 0 & 0 \\ ? & 0 & ? & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Now the question is 'how to decide π, ρ, ϕ and ω'. The answer is:

- There exists one and only one pair of (π, ϕ) such that after MC, Bytes $w_1 \oplus \tilde{w}_1$ and $w_3 \oplus \tilde{w}_3$ are both zero. The values of (π, ϕ) can be decided by solving (11).
- There exists one and only one pair of (ρ, ω) such that after MC, Bytes $w_9 \oplus \tilde{w}_9$ and $w_{11} \oplus \tilde{w}_{11}$ are both zero. By solving (12), we get the values of (ρ, ω).

$$\begin{cases} [01\ \ 02\ \ 03\ \ 01]\begin{bmatrix} \pi \\ n_1 \oplus \tilde{n}_1 \\ \phi \\ n_3 \oplus \tilde{n}_3 \end{bmatrix} = 0 \\ [03\ \ 01\ \ 01\ \ 02]\begin{bmatrix} \pi \\ n_1 \oplus \tilde{n}_1 \\ \phi \\ n_3 \oplus \tilde{n}_3 \end{bmatrix} = 0 \end{cases} \tag{11}$$

$$\begin{cases} [01\ \ 02\ \ 03\ \ 01]\begin{bmatrix} \rho \\ n_9 \oplus \tilde{n}_9 \\ \omega \\ n_{11} \oplus \tilde{n}_{11} \end{bmatrix} = 0 \\ [03\ \ 01\ \ 01\ \ 02]\begin{bmatrix} \rho \\ n_9 \oplus \tilde{n}_9 \\ \omega \\ n_{11} \oplus \tilde{n}_{11} \end{bmatrix} = 0 \end{cases} \tag{12}$$

Once we know the values of π, ϕ, ρ and ω, message block $\tilde{M}_y$ can be chosen as follows:

1 Set the values of $\tilde{n}_0^{\text{new}}, \tilde{n}_8^{\text{new}}, \tilde{n}_2^{\text{new}}$ and $\tilde{n}_{10}^{\text{new}}$ as follows: $\tilde{n}_0^{\text{new}} = n_0 \oplus \pi, \tilde{n}_8^{\text{new}} = n_8 \oplus \rho, \tilde{n}_2^{\text{new}} = n_2 \oplus \phi$ and $\tilde{n}_{10}^{\text{new}} = n_{10} \oplus \omega$. Use $\tilde{n}_0^{\text{new}}$ to replace $\tilde{n}_0, \tilde{n}_0^{\text{new}}$ to replace $\tilde{n}_8, \tilde{n}_2^{\text{new}}$ to replace $\tilde{n}_2$ and $\tilde{n}_{10}^{\text{new}}$ to replace $\tilde{n}_{10}$.

2 Perform SR^{-1} and SB^{-1}. Since SR^{-1} and SB^{-1} are permutation and substitution, they do not affect the properties we have found. Now we have the outputs of ARK in Round *y*.

3 Compute the value of $\tilde{M}_y$ as follows:

$$\tilde{M}_y = \left(\tilde{n}_0^{\text{new}} \oplus \tilde{s}_0\right) \middle\| \left(\tilde{n}_8^{\text{new}} \oplus \tilde{s}_8\right) \\ \left\| \left(\tilde{n}_{10}^{\text{new}} \oplus \tilde{s}_2\right) \right\| \left(\tilde{n}_2^{\text{new}} \oplus \tilde{s}_{10}\right)$$

So far, two messages $(M_{y-3}, M_{y-2}, M_{y-1}, M_y)$ and $(\tilde{M}_{y-3}, \tilde{M}_{y-2}, \tilde{M}_{y-1}^{\text{new}}, \tilde{M}_y)$ collide on 96 bits (i.e. Bytes w_1, w_3, w_4, w_5, w_6, w_7, w_9, w_{11}, w_{12}, w_{13}, w_{14} and w_{15}) in Round $y + 1$ after MC transformation.

6.1.4 Extending 96- to 128-bit collisions

This step is straightforward as we can select message M_{y+1} arbitrarily, and construct message $\tilde{M}_{y+1}$ to cancel the differences between Bytes w_0, w_8, w_2 and w_{10}. The construction is provided as follows:

$$\tilde{M}_{y+1} = \left(\left(w_0 \oplus \tilde{w}_0\right) \middle\| \left(w_8 \oplus \tilde{w}_8\right) \\ \left\| \left(w_2 \oplus \tilde{w}_2\right) \right\| \left(w_{10} \oplus \tilde{w}_{10}\right)\right) \oplus M_{y+1}$$

6.1.5 The BNB search

The BNB search has the following form:

Known:	1 A selected key or a selected intermediate value. 2 One selected four-block message $M(M_{y-3}, M_{y-2}, M_{y-1}, M_y)$
Find:	Another four-block message $\tilde{M}(\tilde{M}_{y-3}, \tilde{M}_{y-2}, \tilde{M}_{y-1}, \tilde{M}_y)$ such that these two messages collide on 32 bits (Bytes s_4, s_{12}, s_6 and s_{14}) after MC in Round *y*
Method:	Solve four groups of linear functions. These four groups of functions are named as (13)–(16).

We propose a method to find 32-bit collisions on Bytes s_4, s_{12}, s_6 and s_{14} (see Figure 7) by solving four groups of linear functions. This search assumes that for a selected key (or a selected intermediate value) and a selected four-block message $(M_{y-3}, M_{y-2}, M_{y-1}, M_y)$, we can generate another four-block message $(\tilde{M}_{y-3}, \tilde{M}_{y-2}, \tilde{M}_{y-1}, \tilde{M}_y)$ such that these two messages collide on Bytes s_4, s_{12}, s_6 and s_{14} after MC in Round *y*. The method used by the BNB search is similar to the idea employed by the BNF search, but works in only one direction (i.e. only backwards).

6.1.6 Deciding four values ($\tilde{j}_5, \tilde{j}_7, \tilde{j}_{13}$ and $\tilde{j}_{15}$)

In the beginning, we choose $(\tilde{M}_{y-3}, \tilde{M}_{y-2}, \tilde{M}_{y-1}, \tilde{M}_y)$ randomly. Assume that the input and the output of MC in Round *y* are listed as follows:

$$\begin{bmatrix} \tilde{j}_0 & \tilde{j}_4 & \tilde{j}_8 & \tilde{j}_{12} \\ \tilde{j}_1 & \tilde{j}_5^{\text{old}} & \tilde{j}_9 & \tilde{j}_{13}^{\text{old}} \\ \tilde{j}_2 & \tilde{j}_6 & \tilde{j}_{10} & \tilde{j}_{14} \\ \tilde{j}_3 & \tilde{j}_7^{\text{old}} & \tilde{j}_{11} & \tilde{j}_{15}^{\text{old}} \end{bmatrix} \xrightarrow{\text{MC}} \begin{bmatrix} \tilde{s}_0 & \tilde{s}_4 & \tilde{s}_8 & \tilde{s}_{12} \\ \tilde{s}_1 & \tilde{s}_5 & \tilde{s}_9 & \tilde{s}_{13} \\ \tilde{s}_2 & \tilde{s}_6 & \tilde{s}_{10} & \tilde{s}_{14} \\ \tilde{s}_3 & \tilde{s}_7 & \tilde{s}_{11} & \tilde{s}_{15} \end{bmatrix}$$

Now we do not use the values of $\tilde{j}_5^{\text{old}}, \tilde{j}_7^{\text{old}}, \tilde{j}_{13}^{\text{old}}$ or $\tilde{j}_{15}^{\text{old}}$. Instead, we use $\tilde{j}_5$ (to replace $\tilde{j}_5^{\text{old}}$), $\tilde{j}_7$ (to replace $\tilde{j}_7^{\text{old}}$), $\tilde{j}_{13}$ (to replace $\tilde{j}_{13}^{\text{old}}$), and $\tilde{j}_{15}$ (to replace $\tilde{j}_{15}^{\text{old}}$) such that we get values $\tilde{s}_4, \tilde{s}_{12}, \tilde{s}_6$ and $\tilde{s}_{14}$, respectively (illustrated as follows):

$$\begin{bmatrix} \tilde{j}_0 & \tilde{j}_4 & \tilde{j}_8 & \tilde{j}_{12} \\ \tilde{j}_1 & \tilde{j}_5 & \tilde{j}_9 & \tilde{j}_{13} \\ \tilde{j}_2 & \tilde{j}_6 & \tilde{j}_{10} & \tilde{j}_{14} \\ \tilde{j}_3 & \tilde{j}_7 & \tilde{j}_{11} & \tilde{j}_{15} \end{bmatrix} \xrightarrow{\text{MC}} \begin{bmatrix} \tilde{s}_0 & \tilde{s}_4 & \tilde{s}_8 & s_{12} \\ \tilde{s}_1 & \tilde{s}_5 & \tilde{s}_9 & \tilde{s}_{13} \\ \tilde{s}_2 & s_6 & \tilde{s}_{10} & s_{14} \\ \tilde{s}_3 & \tilde{s}_7 & \tilde{s}_{11} & \tilde{s}_{15} \end{bmatrix}$$

Now the question is 'how can we make this happen'. Our answer is to solve two groups of linear functions. For the values of s_4 and s_6, we have two linear equations in (13) with only two unknown variables ($\tilde{j}_5$ and $\tilde{j}_7$). Therefore, we can solve (13) to obtain the values of $\tilde{j}_5$ and $\tilde{j}_7$

$$\begin{cases} \begin{bmatrix} 02 & 03 & 01 & 01 \end{bmatrix} \begin{bmatrix} \tilde{j}_4 \\ \tilde{j}_5 \\ \tilde{j}_6 \\ \tilde{j}_7 \end{bmatrix} = s_4 \\ \begin{bmatrix} 01 & 01 & 02 & 03 \end{bmatrix} \begin{bmatrix} \tilde{j}_4 \\ \tilde{j}_5 \\ \tilde{j}_6 \\ \tilde{j}_7 \end{bmatrix} = s_6 \end{cases} \tag{13}$$

$$\begin{cases} \begin{bmatrix} 02 & 03 & 01 & 01 \end{bmatrix} \begin{bmatrix} \tilde{j}_{12} \\ \tilde{j}_{13} \\ \tilde{j}_{14} \\ \tilde{j}_{15} \end{bmatrix} = s_{12} \\ \begin{bmatrix} 01 & 01 & 02 & 03 \end{bmatrix} \begin{bmatrix} \tilde{j}_{12} \\ \tilde{j}_{13} \\ \tilde{j}_{14} \\ \tilde{j}_{15} \end{bmatrix} = s_{14} \end{cases} \tag{14}$$

Similarly, for the values of s_{12} and s_{14}, we have two linear functions in (14) with two unknown variable ($\tilde{j}_{13}$ and $\tilde{j}_{15}$). We can solve (14) to decide the values of $\tilde{j}_{13}$ and $\tilde{j}_{15}$. After getting four vehicles ($\tilde{j}_5, \tilde{j}_7, \tilde{j}_{13}$ and $\tilde{j}_{15}$) decided, we perform the SR^{-1} and SB^{-1} transformations. As SR^{-1} transformation. As SR^{-1} is permutation and SB^{-1} is substitution $\tilde{j}_5, \tilde{j}_7, \tilde{j}_{13}$ and $\tilde{j}_{15}$ are first relocated then substituted by another four values $\tilde{i}_9, \tilde{i}_3, \tilde{i}_1$ and $\tilde{i}_{11}$, respectively. As the message injection layout does not change the values of $\tilde{i}_9, \tilde{i}_3, \tilde{i}_1$ and $\tilde{i}_{11}$ these four values are not changed after we do ARK. So, we get four known values ($\tilde{i}_9, \tilde{i}_3, \tilde{i}_1$ and i_{11}) after MC in Round y–1. Our next target is to modify message block $\tilde{M}_{y-2}$ so that we get those four values $\tilde{i}_9, \tilde{i}_3, \tilde{i}_1$ and i_{11} after MC in Round $y - 1$.

6.1.7 Modifying message block $\tilde{M}_{y-2}$

Suppose by using the original message block $\tilde{M}_{y-2}$, we have the following states in Round $y - 1$.

$$\begin{bmatrix} \tilde{g}_0^{*old} & \tilde{g}_4 & \tilde{g}_8^{*old} & \tilde{g}_{12} \\ \tilde{g}_1 & \tilde{g}_5 & \tilde{g}_9 & \tilde{g}_{13} \\ \tilde{g}_2^{*old} & \tilde{g}_6 & \tilde{g}_{10}^{*old} & \tilde{g}_{14} \\ \tilde{g}_3 & \tilde{g}_7 & \tilde{g}_{11} & \tilde{g}_{15} \end{bmatrix} \xrightarrow{SB \circ SR} \begin{bmatrix} \tilde{h}_0^{old} & \tilde{h}_4 & \tilde{h}_8^{old} & \tilde{h}_{12} \\ \tilde{h}_1 & \tilde{h}_5 & \tilde{h}_9 & \tilde{h}_{13} \\ \tilde{h}_2^{old} & \tilde{h}_6 & \tilde{h}_{10}^{old} & \tilde{h}_{14} \\ \tilde{h}_3 & \tilde{h}_7 & \tilde{h}_{11} & \tilde{h}_{15} \end{bmatrix} \xrightarrow{MC} \begin{bmatrix} ? & \tilde{i}_4 & ? & \tilde{i}_{12} \\ ? & \tilde{i}_5 & ? & \tilde{i}_{13} \\ ? & \tilde{i}_6 & ? & \tilde{i}_{14} \\ ? & \tilde{i}_7 & ? & \tilde{i}_{15} \end{bmatrix}$$

Now we replace values $(\tilde{h}_0^{old}, \tilde{h}_2^{old}, \tilde{h}_8^{old}, \tilde{h}_{10}^{old})$ with $(\tilde{h}_0, \tilde{h}_2, \tilde{h}_8, \tilde{h}_{10})$ and then we get those four values ($\tilde{i}_9, \tilde{i}_3, \tilde{i}_1$ and i_{11}) located as follows:

$$\begin{bmatrix} \tilde{g}_0^{*} & \tilde{g}_4 & \tilde{g}_8^{*} & \tilde{g}_{12} \\ \tilde{g}_1 & \tilde{g}_5 & \tilde{g}_9 & \tilde{g}_{13} \\ \tilde{g}_2^{*} & \tilde{g}_6 & \tilde{g}_{10}^{*} & \tilde{g}_{14} \\ \tilde{g}_3 & \tilde{g}_7 & \tilde{g}_{11} & \tilde{g}_{15} \end{bmatrix} \xrightarrow{SB \circ SR} \begin{bmatrix} \tilde{h}_0 & \tilde{h}_4 & \tilde{h}_8 & \tilde{h}_{12} \\ \tilde{h}_1 & \tilde{h}_5 & \tilde{h}_9 & \tilde{h}_{13} \\ \tilde{h}_2 & \tilde{h}_6 & \tilde{h}_{10} & \tilde{h}_{14} \\ \tilde{h}_3 & \tilde{h}_7 & \tilde{h}_{11} & \tilde{h}_{15} \end{bmatrix} \xrightarrow{MC} \begin{bmatrix} ? & \tilde{i}_4 & ? & \tilde{i}_{12} \\ \tilde{i}_1 & \tilde{i}_5 & \tilde{i}_9 & \tilde{i}_{13} \\ ? & \tilde{i}_6 & ? & \tilde{i}_{14} \\ \tilde{i}_3 & \tilde{i}_7 & \tilde{i}_{11} & \tilde{i}_{15} \end{bmatrix}$$

Based on the property of MC transformation, we can form the following two groups of linear functions:

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} \tilde{h}_0 \\ \tilde{h}_1 \\ \tilde{h}_2 \\ \tilde{h}_3 \end{bmatrix} = \tilde{i}_1 \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} \tilde{h}_0 \\ \tilde{h}_1 \\ \tilde{h}_2 \\ \tilde{h}_3 \end{bmatrix} = \tilde{i}_3 \end{cases} \tag{15}$$

$$\begin{cases} \begin{bmatrix} 01 & 02 & 03 & 01 \end{bmatrix} \begin{bmatrix} \tilde{h}_8 \\ \tilde{h}_9 \\ \tilde{h}_{10} \\ \tilde{h}_{11} \end{bmatrix} = \tilde{i}_9 \\ \begin{bmatrix} 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} \tilde{h}_8 \\ \tilde{h}_9 \\ \tilde{h}_{10} \\ \tilde{h}_{11} \end{bmatrix} = \tilde{i}_{11} \end{cases} \tag{16}$$

We know the values of $\tilde{h}_1, \tilde{h}_3, \tilde{h}_9$ and $\tilde{h}_{11}$ from the original message block $\tilde{M}_{y-2}$. We can get the values of $(\tilde{h}_0, \tilde{h}_2)$ by solving (15), and get the values of $(\tilde{h}_8, \tilde{h}_{10})$ by solving (16). After finding the values of $(\tilde{h}_0, \tilde{h}_2, \tilde{h}_8, \tilde{h}_{10})$, we perform SR^{-1} and SB^{-1}, and obtain the corresponding four values $(\tilde{g}_0^*, \tilde{g}_2^*, \tilde{g}_8^*, \tilde{g}_{10}^*)$. Once we know the values of $(\tilde{g}_0^*, \tilde{g}_2^*, \tilde{g}_8^*, \tilde{g}_{10}^*)$, we replace $\tilde{M}_{y-2}$ with $\tilde{M}_{y-2}^{new}$. $\tilde{M}_{y-2}^{new}$ is constructed as follows (note that $\tilde{g}_0, \tilde{g}_8, \tilde{g}_2$ and $\tilde{g}_{10}$ are known from the message block $\tilde{M}_{y-3}$ in Round $y - 3$):

$$\tilde{M}_{y-3}^{new} = \left(\tilde{g}_0^* \oplus \tilde{g}_0\right) \Big\| \left(\tilde{g}_8^* \oplus \tilde{g}_8\right) \Big\| \left(\tilde{g}_2^* \oplus \tilde{g}_2\right) \Big\| \left(\tilde{g}_{10}^* \oplus \tilde{g}_{10}\right)$$

6.1.8 Combining the BNB search with the BNF search

The second-preimage search algorithm combines the BNB search with the BNF search. To search for a second preimage of the ALPHA-MAC, we perform the following steps:

1. Select a key or an intermediate value.
2. Select a five-block message $M(M_{y-3}, M_{y-2}, M_{y-1}, M_y, M_{y+1})$.
3. Generate the second preimage $\tilde{M}(\tilde{M}_{y-3}, \tilde{M}_{y-2}, \tilde{M}_{y-1}, \tilde{M}_y, \tilde{M}_{y+1})$ randomly. We need to guarantee that $\tilde{M}_{y-3}$ is not equal to M_{y-3}.
4. Perform the BNB search to generate 32-bit collisions. The BNB search is done by modifying message block $\tilde{M}_{y-2}$.
5. Use the BNF search to extend those 32- to 128-bit collisions. The BNF search is carried out by modifying the values of $\tilde{M}_{y-1}, \tilde{M}_y$ and $\tilde{M}_{y+1}$. Message $\tilde{M}(\tilde{M}_{y-3}, \tilde{M}_{y-2}, \tilde{M}_{y-1}, \tilde{M}_y, \tilde{M}_{y+1})$ is a second preimage of message $M(M_{y-3}, M_{y-2}, M_{y-1}, M_y, M_{y+1})$ under the selected key (or the selected intermediate value).

The routine of finding second preimages is shown in Table 1, and Figure 8 depicts this finding. The name of the BNB search comes from the fact that searching for $\tilde{M}_{y-2}$ is carried out by moving backwards and then backwards, and the name of the BNF search comes from the fact that searching for $\tilde{M}_{y-1}, \tilde{M}_y$ and $\tilde{M}_{y+1}$ is performed by moving backwards and then forwards (see Table 1). A personal computer takes about 1 sec to find a second preimage of the ALPHA-MAC. A found second preimage of a selected key K (see Table 2) and a selected five-block message M (see Table 3) is M (shown in Table 3). The 128-bit colliding value is listed in Table 4 (note that these two messages are listed after injection layout).

Figure 8 The second-preimage search

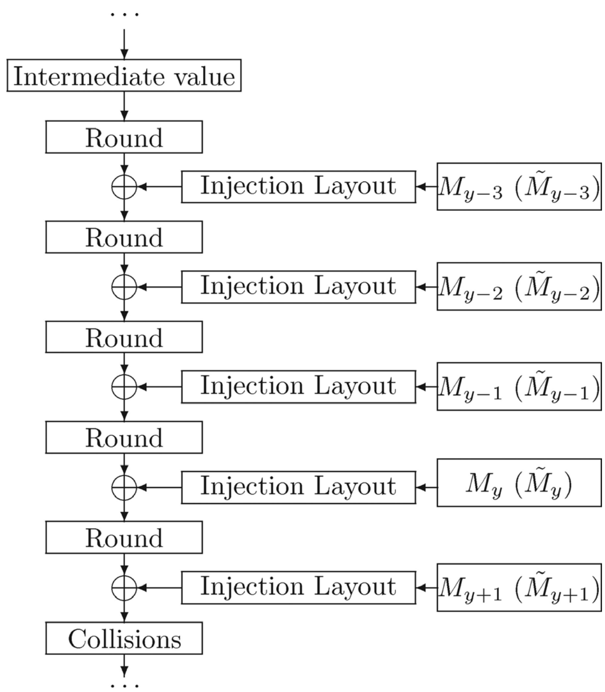


Table 1 Second-preimage search = BNB search + BNF search

Search	*R*	*Round y − 2*	*Di*	*Round y − 1*	*Di*	*Round y*
BNB	1				⇐	$\tilde{s}_4 \rightharpoonup s_4, \tilde{s}_{12} \rightharpoonup s_{12}$, $\tilde{s}_6 \rightharpoonup s_6, \tilde{s}_{14} \rightharpoonup s_{14}$
	2		⇐	$h_0^{\text{old}} \rightharpoonup h_0, h_2^{\text{old}} \rightharpoonup h_2$, $\tilde{h}_8^{\text{old}} \rightharpoonup \tilde{h}_8, \tilde{h}_{10}^{\text{old}} \rightharpoonup \tilde{h}_{10}$		
	3	$M_{y-2} \rightharpoonup M_{y-2}^{\text{new}}$				
		Round y − 1	*Di*	*Round y*	*Di*	*Round y + 1*
BNF	4	Modify M_{y-1}	⇐	collisions on s_4, s_{12}, s_6 and s_{14}		
	5		⇒	collisions on s_4, s_{12}, s_6, s_{14}, s_1, s_9, s_3 and s_{11}		
	6			modify $\tilde{M}_y$	⇒	96-bit collisions
	7					modify M_{y+1} → 128-bit collisions

Note: *Di* – Direction; *R* – Routine.

Table 2 The selected key K

83	55	2d	81
88	2c	05	67
c1	63	be	c2
2a	a2	52	a4

Table 3 Two five-block messages

M (the selected message)																			
M_{y-3}				M_{y-2}				M_{y-1}				M_y				M_{y+1}			
c4	0	8c	0	e6	0	2a	0	77	0	fd	0	ef	0	a1	0	81	0	9f	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
94	0	f3	0	95	0	04	0	4c	0	37	0	68	0	09	0	25	0	2c	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\tilde{M}$ *(the found second preimage)*																			
$\tilde{M}_{y-3}$				$\tilde{M}_{y-2}$				$\tilde{M}_{y-1}$				$\tilde{M}_y$				$\tilde{M}_{y+1}$			
1d	0	43	0	22	0	04	0	e4	0	83	0	2f	0	e5	0	69	0	06	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1c	0	0d	0	2f	0	30	0	2f	0	9b	0	d4	0	30	0	f4	0	3a	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4 The 128-bit collisions

7d	69	88	d7
02	cb	1f	af
b9	d8	7b	5e
0e	10	79	21

6.2 *The collision search algorithm*

Known: A selected key or a selected intermediate value.

Find: Two five-block messages M and $\tilde{M}$ such that they collide under the selected key or the intermediate value.

Method: Employ the second-preimage search.

In the second-preimage search, we choose the first five-block message arbitrarily, and once it is decided, we do not modify it. All we need to do is modify the second five-block message so that 128-bit collisions happen. Therefore, the second-preimage search can also be used to find two colliding five-block messages under a selected key (or a selected intermediate value).

7 Conclusion

We described a five-round algebraic property of the AES algorithm. In the presented property, we change 20 bytes from 5 intermediate values at some fixed locations in 5 consecutive rounds by carrying out 20 extra XOR operations, and we show that after 5 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 by employing the δ algorithm, we constructed a modified version of the AES, the δAES. For a plaintext and a key, the AES and the δAES produce the same ciphertext.

We then showed that the five-round algebraic property of the AES can be used to analyse the internal structure of the ALPHA-MAC, a MAC function whose underlying block cipher is AES. We provided a second-preimage search algorithm and a collision search algorithm.

References

Akkar, M.L. and Giraud, C. (2001) 'An implementation of DES and AES, secure against some attacks', *Cryptographic Hardware and Embedded Systems – CHES 2001, Lecture Notes in Computer Science*, Vol. 2162, Springer-Verlag, pp.309–318.

Barkan, E. and Biham, E. (2002) 'How many ways can you write Rijndael?', *Advances in Cryptology – ASIACRYPT 2002, Lecture Notes in Computer Science*, Vol. 2501, Springer-Verlag, pp.160–175.

Biham, E. and Shamir, A. (1993) *Differential Cryptanalysis of the Data Encryption Standard*, New York, NY: Springer-Verlag.

Biryukov, A. (2007) 'The design of a stream cipher LEX', *Selected Areas in Cryptography – SAC 2006, Lecture Notes in Computer Science*, Vol. 4356, Springer-Verlag, pp.67–75.

Courtois, N., Klimov, A., Patarin, J. and Shamir, A. (2000) 'Efficient algorithms for solving overdefined systems of multivariate polynomial equations', *Advances in Cryptology – EUROCRYPT 2000, Lecture Notes in Computer Science*, Vol. 1807, Springer-Verlag, pp.392–407.

Courtois, N. and Pieprzyk, J. (2002) 'Cryptanalysis of block ciphers with overdefined systems of equations', *Advances in Cryptology – ASIACRYPT 2002, Lecture Notes in Computer Science*, Vol. 2501, Springer-Verlag, pp.267–287.

Daemen, J., Knudsen, L. and Rijmen, V. (1997) 'The block cipher square', *Fast Software Encryption – FSE 1997, Lecture Notes in Computer Science*, Vol. 1267, Springer-Verlag, pp.149–165.

Daemen, J. and Rijmen, V. (2001) 'AES proposal: Rijndael', *AES Round 1 Technical Evaluation CD-1: Documentation*, National Institute of Standards and Technology.

Daemen, J. and Rijmen, V. (2005) 'A new MAC construction ALRED and a specific instance ALPHA-MAC', *Fast Software Encryption – FSE 2005, Lecture Notes in Computer Science*, Vol. 3557, Springer-Verlag, pp.1–17.

Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D. and Whiting, D. (2001) 'Improved cryptanalysis of Rijndael', *Fast Software Encryption – FSE 2000, Lecture Notes in Computer Science*, Vol. 1978, Springer-Verlag, pp.213–230.

Gilbert, H. and Minier, M. (2000) 'A collision attack on 7 rounds of Rijndael', *The Third Advanced Encryption Standard Candidate Conference*, pp.230–241.

Golic, J. and Tymen, C. (2002) 'Multiplicative masking and power analysis of AES', *Cryptographic Hardware and Embedded Systems – CHES 2002, Lecture Notes in Computer Science*, Vol. 2523, Springer-Verlag, pp.198–212.

Lucks, S. (2000) 'Attacking seven rounds of Rijndael under 192 bit and 256-bit keys', *The Third Advanced Encryption Standard Candidate Conference*, pp.215–229.

Matsui, M. (1994) 'Linear cryptoanalysis method for DES cipher', Advances in Cryptology – EUROCRYPT '93, Workshop on the Theory and Application of Cryptographic Techniques, Lecture Notes in Computer Science, Vol. 765, Springer-Verlag, pp.386–397.

Murphy, S. and Robshaw, M. (2002) 'Essential algebraic structure within the AES', *Advances* in *Cryptology – CRYPTO 2002, Lecture Notes in Computer Science*, Vol. 2442, Springer-Verlag, pp.1–16.

National Institute of Standards and Technology (2001) *Federal Information Processing Standards (FIPS) 197: Advanced Encryption Standard (AES)*.

Notes

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

[2]Here and in the rest of this section 'collisions' stands for 'internal collisions'.

Appendix

Examples of AES with 20 XOR operations

We provide seven examples of the outputs of the five algorithm and their corresponding plaintexts, secret keys and ciphertexts in Figure A1(a)–(g).

Figure A1 The values of *P*, *K*, AES round keys and the 20 bytes

Plaintext P

3b	d1	bd	7c
87	ad	29	25
4d	1d	cf	41
d2	d7	07	ae

cipher Key K

03	43	9c	c9
a6	f4	57	9f
26	c5	90	29
11	30	39	e2

Key Expansion $\longrightarrow$ AES Round Keys

Initial Round Key

03	43	9c	c9
a6	f4	57	9f
26	c5	90	29
11	30	39	e2

Round Key 1

d9	9a	06	cf
03	f7	a0	3f
be	7b	eb	c2
cc	fc	c5	27

Round Key 2

ae	34	32	fd
26	d1	71	4e
72	09	e2	20
46	ba	7f	58

Round Key 3

85	b1	83	7e
91	40	31	7f
18	11	f3	d3
12	a8	d7	8f

Round Key 4

5f	ee	6d	13
f7	b7	86	f9
6b	7a	89	5a
e1	49	9e	11

Round Key 5

d6	38	55	46
49	fe	78	81
e9	93	1a	40
9c	d5	4b	5a

Round Key 6

fa	c2	97	d1
40	be	c6	47
57	c4	de	9e
c6	13	58	02

Round Key 7

1a	d8	4f	9e
4b	f5	33	74
20	e4	3a	a4
f8	eb	b3	b1

Round Key 8

08	d0	9f	01
02	f7	c4	b0
e8	0c	36	92
f3	18	ab	1a

Round Key 9

f4	24	bb	ba
4d	ba	7e	ce
4a	46	70	e2
8f	97	3c	26

Round Key 10

49	6d	d6	6c
d5	6f	11	df
bd	fb	8b	69
7b	ec	d0	f6

The twenty bytes

ed	0	dc	0
0	0	0	0
9b	0	04	0
0	0	0	0

58	0	81	0
0	0	0	0
fd	0	81	0
0	0	0	0

72	0	ed	0
0	0	0	0
5d	0	26	0
0	0	0	0

cd	0	3e	0
0	0	0	0
c6	0	9f	0
0	0	0	0

69	0	19	0
0	0	0	0
4f	0	04	0
0	0	0	0

Ciphertext

da	68	03	a0
c9	7e	cc	09
4d	b6	93	8a
38	ea	62	48

(a)

Figure A1 The values of P, K, AES round keys and the 20 bytes (continued)

Plaintext P

b7	4d	6b	86
9b	8a	36	53
4e	f8	ca	1c
5d	63	08	2b

cipher Key K

6d	4e	e7	cf
db	c1	c7	0c
27	86	33	23
b4	70	4a	5c

Key Expansion $\longrightarrow$

AES Round Keys

Initial Round Key

6b	4e	e7	cf
db	c1	c7	0c
27	86	33	23
b4	70	4a	5c

Round Key 1

92	dc	3b	f4
fd	3c	fb	f7
6d	eb	d8	fb
3e	4e	04	58

Round Key 2

f8	24	1f	eb
f2	ce	35	c2
07	ec	34	cf
81	cf	cb	93

Round Key 3

d9	fd	e2	ff
78	b6	83	cc
db	37	03	41
68	a7	6c	09

Round Key 4

52	af	4d	44
33	85	06	47
cd	fa	f9	35
69	ce	a2	5d

Round Key 5

e2	4d	00	44
a5	20	26	61
81	7b	82	b7
72	bc	1e	43

Round Key 6

2d	60	60	24
0c	2c	0a	6b
9b	e0	62	d5
69	d5	cb	88

Round Key 7

12	72	12	36
0f	23	29	42
5f	bf	dd	08
5f	8a	41	c9

Round Key 8

be	cc	de	e8
3f	1c	35	77
82	3d	e0	e8
5a	d0	91	58

Round Key 9

50	9c	42	aa
a4	b8	8b	fa
e8	d5	35	dd
c1	11	80	d8

Round Key 10

4b	d7	95	3f
65	dd	50	aa
89	5c	69	b4
6d	7c	fc	24

The twenty bytes

fb	0	bb	0
0	0	0	0
07	0	11	0
0	0	0	0

74	0	64	0
0	0	0	0
67	0	af	0
0	0	0	0

85	0	e3	0
0	0	0	0
39	0	1b	0
0	0	0	0

9b	0	ec	0
0	0	0	0
10	0	bb	0
0	0	0	0

ca	0	05	0
0	0	0	0
90	0	0d	0
0	0	0	0

Ciphertext

f8	05	b3	09
e6	f8	42	45
36	b8	4d	e3
bb	44	a0	21

(b)

Figure A1 The values of *P*, *K*, AES round keys and the 20 bytes (continued)

Plaintext P

16	2f	90	60
1c	c5	84	f8
a8	c6	c1	a4
0f	00	c2	ae

cipher Key K

48	f1	43	a0
c5	59	7b	bd
68	d5	21	32
88	ca	43	fb

Key Expansion $\longrightarrow$ AES Round Keys

Initial Round Key

48	f1	43	a0
c5	59	7b	bd
68	d5	21	32
88	ca	43	fb

Round Key 1

33	c2	81	21
e6	bf	c4	79
67	b2	93	a1
68	a2	e1	1a

Round Key 2

87	45	c4	e5
d4	6b	af	d6
c5	77	e4	45
95	37	d6	cc

Round Key 3

75	30	f4	11
ba	d1	7e	a8
8e	f9	1d	58
4c	7b	ad	61

Round Key 4

bf	8f	7b	6a
d0	01	7f	d7
61	98	85	dd
ce	b5	18	79

Round Key 5

a1	2e	55	3f
11	10	6f	b8
d7	4f	ca	17
cc	79	61	18

Round Key 6

ed	c3	96	a9
e1	f1	9e	26
7a	35	ff	e8
b9	c0	a1	b9

Round Key 7

5a	99	0f	a6
7a	8b	15	33
2c	19	e6	0e
6a	aa	0b	b2

Round Key 8

4e	80	8f	29
1b	5a	4f	7c
d1	02	e4	ea
19	e4	ef	5d

Round Key 9

12	92	1d	34
56	0c	43	3f
57	55	b1	5b
eb	0f	e0	bd

Round Key 10

51	c3	de	ea
6f	63	20	1f
2d	78	c9	92
f3	fc	1c	a1

The twenty bytes

bd	0	26	0
0	0	0	0
3e	0	ec	0
0	0	0	0

68	0	86	0
0	0	0	0
c6	0	46	0
0	0	0	0

f0	0	4e	0
0	0	0	0
0d	0	ee	0
0	0	0	0

22	0	79	0
0	0	0	0
a9	0	bc	0
0	0	0	0

1e	0	7c	0
0	0	0	0
12	0	82	0
0	0	0	0

Ciphertext

69	2d	52	ad
27	7b	50	8d
69	2c	d3	8c
42	72	a6	ac

(c)

Figure A1 The values of *P*, *K*, AES round keys and the 20 bytes (continued)

Plaintext P

87	d8	b0	e1
04	bd	51	ef
a6	8d	77	b0
67	3f	a2	8b

cipher Key K

2c	03	f5	58
17	35	f9	e3
2a	a7	82	5d
46	64	91	c7

Key Expansion $\longrightarrow$ AES Round Keys

Initial Round Key

2c	03	f5	c7
17	35	f9	5d
2a	a7	82	c3
46	64	91	58

Round Key 1

3c	3f	ca	92
5b	6e	97	74
ec	4b	c9	94
2c	48	d9	1e

Round Key 2

ac	93	59	ec
79	17	80	88
9e	d5	1c	f4
63	2b	f2	cb

Round Key 3

17	84	dd	16
bd	aa	2a	de
50	85	99	11
7c	57	a5	49

Round Key 4

02	86	5b	4d
3f	95	bf	61
6b	ee	77	66
3b	6c	c9	80

Round Key 5

fd	7b	20	6d
0c	99	26	47
a6	48	3f	59
d8	b4	7d	fd

Round Key 6

7d	06	26	4b
c7	5e	78	3f
f2	ba	85	dc
e4	50	2d	d0

Round Key 7

48	4e	68	23
41	1f	67	58
82	38	bd	61
57	07	2a	fa

Round Key 8

a2	ec	84	a7
ae	b1	d6	8e
af	97	2a	4b
71	76	5c	a6

Round Key 9

a0	4c	c8	6f
1d	ac	7a	f4
8b	1c	36	7d
2d	5b	07	a1

Round Key 10

29	65	ad	c2
e2	4e	34	c0
b9	a5	93	ee
85	de	d9	78

The twenty bytes

26	0	8e	0
0	0	0	0
b5	0	bc	0
0	0	0	0

db	0	de	0
0	0	0	0
5a	0	82	0
0	0	0	0

3d	0	44	0
0	0	0	0
49	0	ad	0
0	0	0	0

4c	0	79	0
0	0	0	0
00	0	de	0
0	0	0	0

57	0	e9	0
0	0	0	0
3f	0	1b	0
0	0	0	0

Ciphertext

79	9c	7e	58
bd	b7	2b	11
5b	eb	b0	48
56	b0	37	b7

(d)

Figure A1 The values of *P*, *K*, AES round keys and the 20 bytes (continued)

Plaintext P

b9	22	bd	ea
ca	e9	4b	ac
66	b4	67	96
de	3b	64	fc

cipher Key K

10	db	32	b9
db	d6	e0	43
4c	c3	8c	10
68	f8	63	00

Key Expansion $\longrightarrow$ AES Round Keys

Initial Round Key

10	db	32	b9
db	d6	e0	43
4c	c3	8c	10
68	f8	63	00

Round Key 1

0b	d0	e2	5b
11	c7	27	64
2d	ce	42	52
3e	c6	a5	a5

Round Key 2

4a	9a	78	23
11	d6	f1	95
2b	e5	a7	f5
07	c1	64	c1

Round Key 3

64	fe	86	a5
f7	21	d0	45
53	b6	11	e4
21	e0	84	45

Round Key 4

02	fc	7a	df
9e	bf	6f	2a
3d	8b	9a	7e
27	c7	43	06

Round Key 5

f7	0b	71	ae
6d	d2	bd	97
52	d9	43	3d
b9	7e	3d	3b

Round Key 6

5f	54	25	8b
4a	98	25	b2
b0	69	2a	17
5d	23	1e	25

Round Key 7

28	7c	59	d2
ba	22	07	b5
8f	e6	cc	db
60	43	5d	78

Round Key 8

7d	01	58	8a
03	21	26	93
33	d5	19	c2
d5	96	cb	b3

Round Key 9

ba	bb	e3	69
26	07	21	b2
5e	8b	92	50
ab	3d	f6	45

Round Key 10

bb	00	e3	8a
75	72	53	e1
30	bb	29	79
52	6f	99	dc

The twenty bytes

b4	0	bb	0
0	0	0	0
c9	0	17	0
0	0	0	0

81	0	2a	0
0	0	0	0
e4	0	9c	0
0	0	0	0

36	0	e4	0
0	0	0	0
bc	0	3d	0
0	0	0	0

9b	0	1f	0
0	0	0	0
ae	0	f8	0
0	0	0	0

0	0	73	0
0	0	0	0
45	0	75	0
0	0	0	0

Ciphertext

ad	75	0e	98
d1	b6	06	f7
9f	92	00	d4
a2	cb	6f	e2

(e)

Figure A1 The values of P, K, AES round keys and the 20 bytes (continued)

Plaintext P

eb	6b	cb	f3
90	15	45	68
25	db	57	7c
55	37	26	6c

cipher Key K

f4	b2	70	19
9e	76	c7	a4
73	1f	96	c2
8b	8c	36	39

Key Expansion $\longrightarrow$ AES Round Keys

Initial Round Key

f4	b2	70	19
9e	76	c7	a4
73	1f	96	c2
8b	8c	36	39

Round Key 1

bc	0e	7e	67
bb	cd	0a	ae
61	7e	e8	2a
5f	d3	e5	dc

Round Key 2

5a	54	2a	4d
5e	93	99	37
e7	99	71	5b
da	09	ec	30

Round Key 3

c4	90	ba	f7
67	f4	6d	5a
e3	7a	0b	50
39	30	dc	ec

Round Key 4

72	e2	58	af
34	c0	ad	f7
2d	57	5c	0c
51	61	bd	51

Round Key 5

0a	e8	b0	1f
ca	0a	a7	50
fc	ab	f7	fb
28	49	f4	a5

Round Key 6

79	91	21	3e
c5	cf	68	38
fa	51	a6	5d
e8	a1	55	f0

Round Key 7

3e	af	8e	b0
89	46	2e	16
76	27	81	dc
5a	fb	ae	5e

Round Key 8

f9	56	d8	68
0f	49	67	71
2e	09	88	54
bd	46	e8	b6

Round Key 9

41	17	cf	a7
2f	66	01	70
60	69	e1	b5
f8	be	56	e0

Round Key 10

26	31	fe	59
fa	9c	9d	ed
81	e8	09	bc
a4	1a	4c	ac

The twenty bytes

42	0	e9	0
0	0	0	0
de	0	72	0
0	0	0	0

6c	0	40	0
0	0	0	0
f5	0	f5	0
0	0	0	0

94	0	25	0
0	0	0	0
1b	0	95	0
0	0	0	0

35	0	4a	0
0	0	0	0
67	0	28	0
0	0	0	0

f3	0	21	0
0	0	0	0
b1	0	f4	0
0	0	0	0

Ciphertext

79	45	9a	b1
69	34	69	9d
04	ae	3b	5d
78	2a	47	01

(f)

Figure A1 The values of *P*, *K*, AES round keys and the 20 bytes (continued)

Plaintext P

99	30	d8	1f
7e	99	08	84
c4	24	91	d1
b0	21	92	1e

cipher Key K

11	b2	60	3c
2f	30	c2	6b
3a	c5	8c	c4
d0	82	1e	5b

Key Expansion $\longrightarrow$ AES Round Keys

Initial Round Key

11	b2	60	3c
2f	30	c2	6b
3a	c5	8c	c4
d0	82	1e	5b

Round Key 1

6f	dd	bd	81
33	03	c1	aa
03	c6	48	8c
3b	b9	a7	fc

Round Key 2

c1	1c	a1	20
57	54	95	3f
b3	75	3d	b1
37	8e	29	d5

Round Key 3

b0	ac	0d	2d
9f	cb	5e	61
b0	c5	f8	49
80	0e	27	f2

Round Key 4

57	fb	f6	db
a4	6f	31	50
39	fc	04	4d
58	56	71	83

Round Key 5

14	ef	19	c2
47	28	19	49
d5	29	2d	60
e1	b7	c6	45

Round Key 6

0f	e0	f9	3b
97	bf	a6	ef
bb	92	bf	df
c4	73	b5	f0

Round Key 7

90	70	89	b2
09	b6	10	ff
37	a5	1a	c5
26	55	e0	10

Round Key 8

06	76	ff	4d
af	19	09	f6
fd	58	42	87
11	44	a4	b4

Round Key 9

5f	29	d6	9b
b8	a1	a8	5e
70	28	6a	ed
f2	b6	12	a6

Round Key 10

31	18	ce	55
ed	4c	e4	ba
54	7c	16	fb
e6	50	42	e4

The twenty bytes

fd	0	93	0
0	0	0	0
de	0	c7	0
0	0	0	0

7c	0	fa	0
0	0	0	0
e9	0	48	0
0	0	0	0

d4	0	3e	0
0	0	0	0
2d	0	60	0
0	0	0	0

e4	0	fb	0
0	0	0	0
d6	0	02	0
0	0	0	0

8f	0	aa	0
0	0	0	0
33	0	4f	0
0	0	0	0

Ciphertext

f2	c4	0a	90
5b	8f	0a	41
05	ae	83	56
26	df	06	0d

(g)