# A New Approach for Designing Key-Dependent S-Box Defined over GF ($2^4$) in AES

Hanem M. El-Sheikh, Omayma A. El-Mohsen, *Senior Member, IACSIT,* Talaat Elgarf, and Abdelhalim Zekry, *Senior Member, IACSIT*

***Abstract*—In this paper a new approach for designing S-box in Advanced Encryption Standard (AES) is proposed. The proposed S-box is constructed from small S-boxes defined over GF ($2^4$) instead of GF ($2^8$) as in traditional AES. Rijndael Algorithm (RA), as one of AES standards, is modified by applying the new approach. The Modified Rijndael Algorithm (MRA) is constructed by replacing the S-box of RA by small S-boxes, and the key expansion procedure of RA is modified consequently. Each one of the small S-boxes has different equation and each equation is extracted using one of the three irreducible polynomials existing in GF ($2^4$). So, detecting different equations by cryptanalysts is very difficult compared to the S-box of RA which uses one equation and one irreducible polynomial. The substitution from small S-boxes is done based on the round key, so this achieves diffusion, confusion and therefore security for MRA. The MRA is tested using avalanche effect and strict avalanche criterion (SAC) to evaluate security. The performance evaluation is calculated and proved that MRA is more suitable for the applications that require security and QoS such as voice over IP (VoIP).**

***Index Terms*—AES, key dependent S-box, finite field, cryptographic algorithms, strict avalanche criterion.**

## I. Introduction

The Advanced Encryption System (AES) was launched as a symmetrical cryptography standard algorithm by the National Institute of Standard and Technology (NIST) in October 2000, after a four year effort to replace the aging DES, NIST announced the selection of Rijndael, as in [1], [2] as the proposed AES (NIST 2004). Draft of the Federal Information Processing Standard (FIPS), as in [3] for the AES was published in February 2001; Standardization of AES was approved after public review and comments, and published a final standard FIPS PUB-197, as in [3] in December 2001. Standardization was effective in May 2002 (NIST 2004). Rijndael submitted by Joan Daemen and Vincent Rijmen (Daemen 1998), is a symmetric key, iterated block cipher based on the arithmetic in the Galois Field of $2^8$ elements – GF ($2^8$). The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192, or 256 bits. The AES specification uses the same three key size alternatives but limits the block length to 128 bits, as in [4]. The input to the encryption and decryption algorithms is a single 128-bit block and this block is depicted as a square matrix of bytes. The key that is provided as input is expanded into an array of key schedule words, each word is four bytes and the total key schedule is 44 words for the 128-bit key. Four different stages are used:

1) SubByte transformation: Is a non linear byte Substitution, using a substitution table (S-box), which is constructed by multiplicative inverse and affine transformation. It provides nonlinearity and confusion.

2) ShiftRows transformation: Is a simple byte transposition, the bytes in the last three rows of the state are cyclically shifted; the offset of the left shift varies from diffusion one to three bytes. It provides inter-column.

3) MixColumns transformation: Is equivalent to a matrix multiplication of columns of the states. Each column vector is multiplied by a fixed matrix. It should be noted that the bytes are treated as polynomials rather than numbers. It provides inter-byte diffusion.

4) AddRoundKey transformation: Is a simple XOR between the working state and the roundkey. This transformation is its own inverse. It adds confusion.

For both encryption and decryption, the cipher begins with an AddRoundKey stage, followed by nine rounds that each includes all four stages, followed by a tenth round of three stages.

This paper is organized as follows: In Section II, a proposed methodology for designing small S-boxes is illustrated. In Section III, the modification of the cipher key schedule is described. Algorithm for applying the proposed methodology for designing small S-boxes is performed in Section IV. The evaluation criteria of MRA are examined in Section V. The performance evaluation of MRA is explained in Section VI.

## II. A Proposed Methodology for Designing Small S-Boxes

S-box is the important part of RA because it gives nonlinearity to cryptosystems, but it causes the most of delay of the encryption algorithm. Many efforts were emulated to redesign, reconstruct or renew the design and implementation of the S-box, as in [5], [6]. In the RA, two modifications are done for obtaining the MRA, the first modification is performed in the S-box methodology and the second modification is performed in the cipher key expansion process. In this section, the first modification is illustrated and the second modification is illustrated in section III. In the proposed methodology for designing small S-boxes, two

Manuscript received February 17, 2012; revised March 31, 2012.

Hanem M. El-Sheikh was with the Faculty of Engineering, Alexandria University, Alexandria, Egypt (e-mail: honymora@yahoo.com).

Omayma A. Mohsen is with the Switching Department, and the scientific committee at the National Telecommunication Institute.

Abdelhalim Zekry is with the electronics at faculty of Engineering, Ain Shams University, Cairo, Egypt.

processes are used for substituting input byte by another byte in SubByte transformation. The first process is the construction of S-box Table and Inv S-box Table from small S-boxes and the second process is the substitution method from S-box Table and Inv S-box Table which is done depending on the modified cipher key.

### A. Construction of S-Box Table and Inv S-Box Table

In RA, S-box is based on an operation of inversion in Galois Field GF ($2^8$) using one irreducible polynomial and based on byte substitution, as in [7]. In MRA, small S-boxes are constructed based on an operation of inversion in GF ($2^4$) and based on nibble substitution. The field GF ($2^4$) has 16 binary polynomials of degree at most 3. Also, there are only three irreducible polynomials of degree 4 namely $P_1(x) = x^4 + x + 1$, $P_2(x) = x^4 + x^3 + 1$, and $P_3(x) = x^4 + x^3 + x^2 + x + 1$. For each irreducible polynomial, the multiplicative inverse of the 16 elements written in hexadecimal form is listed in Table I.

TABLE I: MULTIPLICATIVE INVERSION TABLE OF $P_1$, $P_2$, $P_3$ IN GF ($2^4$)

| $x$ | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **a** | **b** | **c** | **d** | **e** | **f** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x^{-1}$ **($P_1$)** | 0 | 1 | 9 | e | d | b | 7 | 6 | f | 2 | c | 5 | a | 4 | 3 | 8 |
| $x^{-1}$**($P_2$)** | 0 | 1 | c | 8 | 6 | f | 4 | e | 3 | d | b | a | 2 | 9 | 7 | 5 |
| $x^{-1}$**($P_3$)** | 0 | 1 | f | a | 8 | 6 | 5 | 9 | 4 | 7 | 3 | e | d | c | b | 2 |

In MRA, a proposed S-box table is constructed from small S-boxes using GF ($2^4$) and the number of elements in each small S-box is 16 elements. In RA, SubByte transformation is a non-linear byte substitution that acts on every byte of the state in isolation to produce a new byte value. The SubByte transformation consists of the multiplicative inverse of each byte in GF ($2^8$) and affine transformation using ($8 \times 8$) transformation matrix and (8×1) constant vector. In MRA, SubByte transformation is done by dividing every byte into two nibbles and each nibble is replaced by another nibble from the proposed S-box table. The SubNibble transformation in MRA is constructed by multiplicative inverse and affine transformation as the SubByte transformation in RA, but the multiplicative inverse in SubNibble transformation is performed for each irreducible polynomial in the finite field GF ($2^4$) described earlier in Table I, with the {00} element mapped to itself. Also, the affine transformation is done using (4×4) transformation matrices and (4×1) constant vectors. Choices among the variations (4×4) transformation matrices and (4×1) constant vectors are done and the best will be used with one of the irreducible polynomials for constructing the robust S-boxes as explained below. The robust S-boxes are chosen depends on ensuring that each S-box has no fixed points [S-box ($a$) = $a$] and the S-box is not self inverse [S-box ($a$) = Inv S-box ($a$)]. Thus, each one of the S-boxes has different equation. The SubNibble equation of these S-boxes is constructed as the SubByte equation of RA. For some nibble, the multiplicative inverse B is obtained from Table I, and the SubNibble B' can be calculated from (1).

$$B' = XB \oplus C \tag{1}$$

where X is (4×4) matrix for SubNibble and C is (4×1) vector. For $P_1(x) = x^4 + x + 1$, the best option (4×4) matrix found is matrix $a_1$ and its inverse matrix $a_1^{-1}$ and the best option (4×1) constant vectors are (0xa and 0xf). Equation (2) is the transformation equation for each bit in the nibble using $P_1(x)$ for example.

$$b'_i = b_{(i+2) \bmod 4} \oplus c_i \tag{2}$$

Using matrix $a_1$ and vector 0xa gives small S-box $S_1$ ($P_1$). Also, using matrix $a_1$ and vector 0xf gives small S-box $S_2$ ($P_1$).

$$(a_1) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \qquad (a_1)^{-1} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

For $P_2(x) = x^4 + x^3 + 1$, the best option (4×4) matrix found is matrix $a_2$ and its inverse matrix $a_2^{-1}$. Also, the best option (4 ×1) constant vectors are (0×3, 0×9, 0×*c*, and 0×*d*). Using matrix $a_2$ and vector 0×3 gives small S-box $S_1$ ($P_2$), using matrix $a_2$ and vector 0×9 gives small S-box $S_2$ ($P_2$), using matrix $a_2$ and vector 0×c gives small S-box $S_3$ ($P_2$), and using matrix $a_2$ and vector 0×d gives small S-box $S_4$ ($P_2$).

$$(a_2) = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \qquad (a_2)^{-1} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

For $P_3(x) = x^4 + x^3 + x^2 + x + 1$, the best option (4×4) matrix found is matrix $a_3$ and its inverse matrix $a_3^{-1}$. Also, the best option (4×1) constant vectors are (0×4, 0×5,0×*d*, and 0× *f*).Using matrix $a_3$ and vector 0×4 gives small S-box $S_1(P_3)$, using matrix $a_3$ and vector 0×5 gives small S-box $S_2(P_3)$, using matrix $a_3$ and vector 0×*d* gives small S-box $S_3(P_3)$, and using matrix $a_3$ and vector 0×*f* gives small S-box $S_4(P_3)$.

$$(a_3) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (a_3)^{-1} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

These ten robust small S-boxes can be arranged in more than one different order for constructing more than one S-box table. One of these tables is chosen in this proposed approach and the order of small S-boxes are: $S_1(P_2)$, $S_1(P_3)$, $S_2(P_3)$, $S_1(P_1)$, $S_3(P_3)$, $S_2(P_2)$, $S_3(P_2)$, $S_2(P_1)$, $S_4(P_2)$, and $S_4(P_3)$ which are arranged from $S_1$ to $S_{10}$ respectively for constructing the proposed S-box table as shown in Table II. Also, the proposed inverse S-box table is constructed as shown in Table III. The proposed S-box table(16×10) and Inv S-box table(16×10) is implemented as a look-up table which is smaller in size than S-box table (16×16) of RA. Also, an operation of inversion of 4-bit can be done easily in hardware implementation with a minimal amount of circuitry. Also, hardware implementation of this approach can reserve the efforts done for implementing S-box of RA using composite

field that transform GF ($2^8$) to GF (($(2^4)^2$) or transform GF($2^8$) to GF((($(2^2)^2)^2$), as in [8]-[10].

TABLE II: THE PROPOSED S-BOX TABLE

| | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ | $S_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 3 | 4 | 5 | a | d | 9 | c | f | d | f |
| **1** | 4 | 5 | 4 | e | c | e | b | b | a | e |
| **2** | 5 | c | d | c | 5 | f | a | 9 | b | 7 |
| **3** | 8 | 9 | 8 | 1 | 0 | 2 | 7 | 4 | 6 | 2 |
| **4** | 0 | e | f | d | 7 | a | f | 8 | e | 5 |
| **5** | c | 7 | 6 | 4 | e | 6 | 3 | 1 | 2 | c |
| **6** | e | 1 | 0 | 7 | 8 | 4 | 1 | 2 | 0 | a |
| **7** | b | f | e | 3 | 6 | 1 | 4 | 6 | 5 | 4 |
| **8** | a | 0 | 1 | 5 | 9 | 0 | 5 | 0 | 4 | b |
| **9** | 2 | 6 | 7 | 2 | f | 8 | d | 7 | c | d |
| **a** | 1 | 2 | 3 | 9 | b | b | e | c | f | 9 |
| **b** | 6 | d | c | f | 4 | c | 9 | a | 8 | 6 |
| **c** | d | b | a | 0 | 2 | 7 | 2 | 5 | 3 | 0 |
| **d** | f | a | b | b | 3 | 5 | 0 | e | 1 | 1 |
| **e** | 7 | 8 | 9 | 6 | 1 | d | 8 | 3 | 9 | 3 |
| **f** | 9 | 3 | 2 | 8 | a | 3 | 6 | d | 7 | 8 |

TABLE III: INV OF PROPOSED S-BOX TABLE

| | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ | $S_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 4 | 8 | 6 | c | 3 | 8 | d | 8 | 6 | c |
| **1** | a | 6 | 8 | 3 | e | 7 | 6 | 5 | d | d |
| **2** | 9 | a | f | 9 | c | 3 | c | 6 | 5 | 3 |
| **3** | 0 | f | a | 7 | d | f | 5 | e | c | e |
| **4** | 1 | 0 | 1 | 5 | b | 6 | 7 | 3 | 8 | 7 |
| **5** | 2 | 1 | 0 | 8 | 2 | d | 8 | c | 7 | 4 |
| **6** | b | 9 | 5 | e | 7 | 5 | f | 7 | 3 | b |
| **7** | e | 5 | 9 | 6 | 4 | c | 3 | 9 | f | 2 |
| **8** | 3 | e | 3 | f | 6 | 9 | e | 4 | b | f |
| **9** | f | 3 | e | a | 8 | 0 | b | 2 | e | a |
| **a** | 8 | d | c | 0 | f | 4 | 2 | b | 1 | 6 |
| **b** | 7 | c | d | d | a | a | 1 | 1 | 2 | 8 |
| **c** | 5 | 2 | b | 2 | 1 | b | 0 | a | 9 | 5 |
| **d** | c | b | 2 | 4 | 0 | e | 9 | f | 0 | 9 |
| **e** | 6 | 4 | 7 | 1 | 5 | 1 | a | d | 4 | 1 |
| **f** | d | 7 | 4 | b | 9 | 2 | 4 | 0 | a | 0 |

### B. Substitution Method from S-Box Table

After constructing the proposed S-box table, the method of substitution from it is performed in two steps and depending on the round key. The first step is the construction of the arrays of substitution S-boxes and Inv S-boxes using the following procedure.

*1) Arrays of substitution S-boxes and Inv S-boxes procedure*

At each round, two variables ($x$ and $y$) with $2^2$ binary combinations are used for determining the small S-box number. The following procedure is used for substitution from S-box table:

*a)* For each byte value of the key, $k_i$ (for $0 \le i \le$ key length), for example, if the key length is 16 Byte, the first byte $k_1$, then $k_2$ and so on. Examine the value of $k_i$, if ($k_i$ mod 2) equals zero, thus $x$= 0, otherwise $x$=1.

*b)* For each byte value of the key ($k_i$), $y$ = the value of the most significant bit of the $k_i$, which equals zero or one.

So, at each round, the combination between $x$ and $y$ values results four states. Thus, four small S-boxes are used and the first small S-box number is taken equals to the round number as shown in Fig. 1.

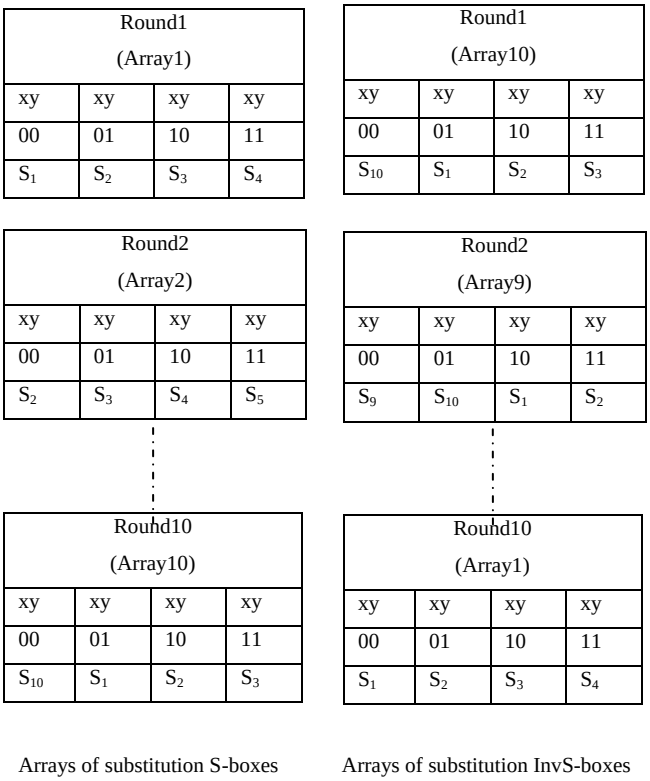



Fig. 1. Arrays of substitution small S-boxes and Inv of small S-boxes.

The second step is the construction of the round key sub-matrix as follows.

*2) Round Key Sub-matrix*

Using the arrays of substitution S-boxes and Inv S-boxes procedure, the round key sub-matrix (4×4) is extracted at each round and contains the number of small S-boxes which will be used in substitution. These Sub-matrices are stored also as look-up table before starting the encryption algorithm. For example, suppose the following plaintext and cipher key are used in this propose.

Plaintext_hex = 328831*e*0435*a*3137*f*6309807*a*88*da*234

Cipher Key_he = 2*b*28*ab*097*eae*f7*cf*15*d*2154*f*16*a*6883*c*

Using MRA, and key schedule modification procedure in Section III, the first extracted key is $k_1$.

$k_1$_hex = 0b2423423*c*37*e*3*a*463*abe*6288*eb*18b91

Round key sub- matrix related to $k_1$ is extracted and will be used in substitution from S-box table (16×10). For example, in round1 ($S_1$, $S_2$, $S_3$, and $S_4$) are used as in Fig. 1, and each byte of $k_1$_hex is used for calculating $x$ and $y$ variables as follows:

Hex2*dec* (0*b*) =11, 11mod2=1, thus $x$=1

Hex2bin (0*b*) =00001011, thus $y$=0, thus $x$ $y$=10

So, the small S-box ($S_3$) is used in substitution.

Applying the procedure on all bytes of $k_1$_hex, the following round key sub-matrix is resulted.

| | | | |
|---|---|---|---|
| 0b | 3c | 63 | 8e |
| 24 | 37 | ab | b1 |
| 23 | e3 | e6 | 8b |
| 42 | a4 | 28 | 91 |

$k_1$-matrix

| | | | |
|---|---|---|---|
| $S_3$ | $S_1$ | $S_3$ | $S_2$ |
| $S_1$ | $S_3$ | $S_4$ | $S_4$ |
| $S_3$ | $S_4$ | $S_2$ | $S_4$ |
| $S_1$ | $S_2$ | $S_1$ | $S_4$ |

$k_1$ sub-matrix

## III. Key Schedule Modification

The 128 bits of the original cipher key is arranged as a (4 × 4) matrix of bytes. Let $w[0]$, $w[1]$, $w[2]$, and $w[3]$ be the four columns of the original key. The four columns can be recursively expanded to obtain 40 more columns, as in (3). Let the columns up to $w[i-1]$ have been defined then,

$$w[i] = \begin{cases} w[i-4] \oplus w[i-1] & \text{if } i \bmod 4 \neq 0 \\ SubByte\ [w[i-4] \oplus R(w[i-1])] & \text{otherwise} \end{cases} \quad (3)$$

where, $R(w[i-1])$ is a cyclic rotation of the bytes within the column and the addition of a round constant (rcon), $w[i-4]$ XOR with the $R(w[i-1])$ results four columns. After four columns are obtained, the new approach of substitution algorithm above is applied on each byte and each byte is passed into two processes. Process 1 uses arrays of substitution from S-boxes procedure above for determining $x$ and $y$ values which are used to determine the small S-box number depending on the round number. Process2 divides each byte into two nibbles (4-bit); each nibble is replaced by another nibble from substitution S-box table using small S-box number which is determined from process1. These two processes are applied on all 16 bytes of four columns for obtaining the first round key ($k_1$) and the algorithm is repeated for ten rounds.

## IV. Algorithm for Applying The Proposed Small S-Boxes Methodology

Using the previous plaintext and cipher key, the plaintext XOR with cipher key, the temporary state ($S_1$) is obtained which has 16 bytes and is arranged in (4× 4) matrix. Each byte is divided into two nibbles (left and right). The substitution algorithm in Fig. 2 is applied on each nibble (4-bit) individually. The substitution algorithm invokes the round key sub-matrix at each round for determining which small S-box is used in substitution each nibble. Each nibble is replaced by another nibble from the proposed S-box table using the above $k_1$ sub-matrix. So, for example the byte 19, the first byte of state $S_1$, is replaced by 47 because nibble 1 and 9 is replaced by nibble 4 and 7 respectively from $S_3$. This method is repeated for the 16 bytes, so the state $S_{1'}$ is resulted by substitution algorithm and it is the SubByte transformation of state $S_1$ using round key sub-matrix and the proposed S-box table.

The state $S_1$=

| | | | |
|---|---|---|---|
| 19 | 3*d* | *e*3 | *be* |
| *a*0 | f4 | *e*2 | 2*b* |
| 9*a* | *c*6 | 8*d* | 2*a* |
| *e*9 | f8 | 48 | 08 |

The state $S_1$'=

| | | | |
|---|---|---|---|
| 47 | 8*f* | 98 | *d*8 |
| 13 | 2*f* | 6*c* | *cf* |
| 73 | 07 | 0*a* | *c*9 |
| 72 | 30 | 0*a* | *a*5 |

By using Inv-Substitution algorithm, the state $S_1$ is recovered from the state $S_{1'}$ as shown in Fig. 3. By examining each nibble of state $S_{1'}$, it is noted that there are some similar nibbles such as nibble 7, which comes from different small S-boxes ($S_3$ , $S_1$, and $S_4$) as shown in $k_1$ sub- matrix. It is difficult for the analyst to try to obtain the Inv SubNibble value for nibble 7 because it is difficult to know which equation is used for SubNibble transformation. So, the proposed S-box table can be implemented as a secure look-up table. Changing the cipher key periodically, the round key sub-matrices are changed respectively and can be extracted and stored safely before the starting of encryption algorithm (MRA). Thus, the MRA can achieve more security than RA. Also, the possibility of creating new sub-matrices when the cipher key is changed can thwart the attempts of the cryptanalysts for deducing the cipher key.

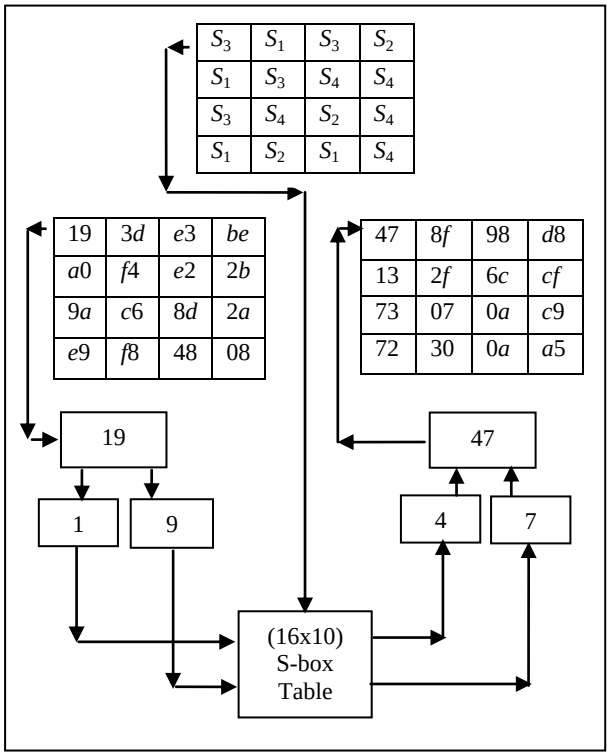



Fig. 2. Substitution algorithm.

## V. Evaluation Criteria of MRA

A good S-box, that satisfies a lot of criteria for its nonlinear properties, determines the performance of the whole block cipher. In addition, to a large degree, it also determines the intensity of the block cipher. To evaluate the proposed small S-boxes, each one must achieve the nonlinearity property which is required for a good S-box. The nonlinearity is easily detected from the Boolean function of each small S-box. Also, some of verification methods must be applied for evaluation the performance of MRA such as: avalanche effect measurements and strict avalanche criterion test. These tests are implemented using MATLAB (R2006b) and the previous plaintext and cipher key in hexadecimal format are used.

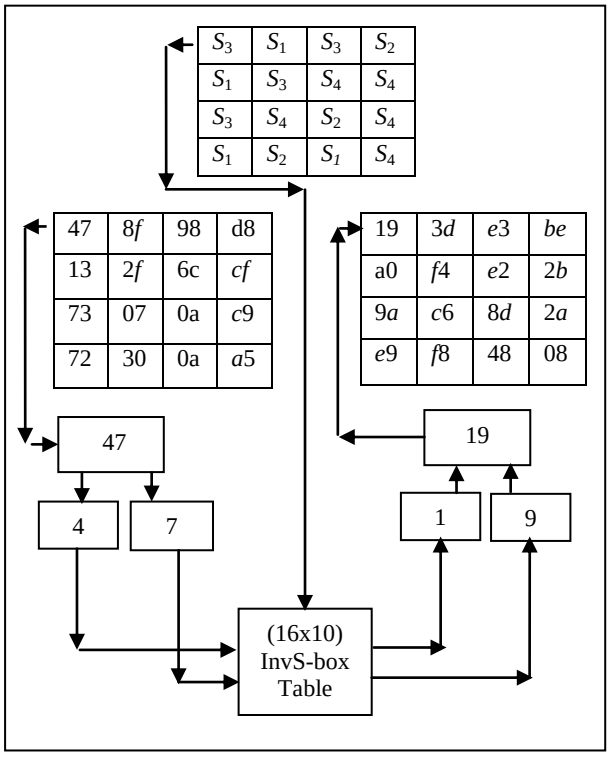



Fig. 3. Inv-Substitution algorithm.

### A. Avalanche Effect Measurements

Avalanche effect is a characteristic of an encryption algorithm in which a small change in the plaintext or key (for example, flipping a single bit) give rise to large change in the cipher text (more than half the bits flip). If a block cipher does not exhibit the avalanche effect to a significant degree, thus it has poor randomization, and a cryptanalyst can make predictions about the input, being given only the output. This may be sufficient to partially or completely break the algorithm. The avalanche effect was calculated for ten rounds using the previous plaintext and cipher key and it is found that the number of vectors that have more than 64 bit changed is 64 vectors of MRA that can be counted from Table IV and 71 vectors of RA that can be counted from Table V. Thus, the avalanche effect is achieved for MRA. The interesting note is that when the number of rounds of MRA is decreased to five rounds the number of avalanche vectors which have more than 64 bit changed increase to 79 vectors. So, the number of rounds which achieve high randomization is five rounds in MRA. So, it is very important issue that the optimum no of rounds that give high randomization must be detected, this issue is out of work of this paper

### B. Strict Avalanche Criterion Test

It is a property of Boolean functions of relevance in cryptography. A function is said to satisfy the strict avalanche criterion if, whenever a single input bit is complemented, each of the output bits should change with a probability of one half, as in [11]. In MRA, each bit of 128-bit of plaintext is changed in sequence with preserving the difference of the two input sequences is 1-bit. It is noted that, a 1-bit change of input sequence causes close to 0.5 probability of each bit change per output sequence. Table IV indicates that each element in the strict avalanche matrix has a value close to one- half. So, the new proposal satisfies the strict avalanche criterion and by comparing it with RA as shown in Table V, there is no a critical difference between them. Also, reducing a number of rounds to five rounds gives good results rather than ten rounds.

TABLE IV: STRICT AVALANCHE MATRIX FOR MRA

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.5078 | 0.4531 | 0.4844 | 0.5078 | 0.4141 | 0.5156 | 0.5156 | 0.4609 |
| 0.5156 | 0.4766 | 0.5156 | 0.4766 | 0.5703 | 0.5234 | 0.4766 | 0.5234 |
| 0.4844 | 0.4609 | 0.5078 | 0.4453 | 0.4844 | 0.5469 | 0.5156 | 0.4922 |
| 0.4375 | 0.4063 | 0.4531 | 0.5547 | 0.4922 | 0.4453 | 0.4922 | 0.4688 |
| 0.4297 | 0.4531 | 0.4688 | 0.4922 | 0.5391 | 0.5547 | 0.4766 | 0.4922 |
| 0.4844 | 0.3984 | 0.5078 | 0.5078 | 0.5078 | 0.5781 | 0.4688 | 0.5234 |
| 0.5859 | 0.4688 | 0.4688 | 0.4531 | 0.5234 | 0.4844 | 0.4922 | 0.4531 |
| 0.5156 | 0.5391 | 0.4609 | 0.5313 | 0.4141 | 0.4063 | 0.5078 | 0.3906 |
| 0.4141 | 0.5781 | 0.4609 | 0.5156 | 0.5391 | 0.4141 | 0.5313 | 0.4609 |
| 0.4766 | 0.4688 | 0.3828 | 0.4063 | 0.4766 | 0.5000 | 0.4766 | 0.5547 |
| 0.4688 | 0.5313 | 0.5469 | 0.4922 | 0.5000 | 0.4922 | 0.5313 | 0.5547 |
| 0.5000 | 0.5313 | 0.5391 | 0.5469 | 0.5547 | 0.5078 | 0.5156 | 0.5000 |
| 0.5391 | 0.5000 | 0.6016 | 0.4609 | 0.5313 | 0.4297 | 0.5625 | 0.5781 |
| 0.5781 | 0.5156 | 0.4844 | 0.4766 | 0.5625 | 0.4844 | 0.5313 | 0.4453 |
| 0.4609 | 0.4609 | 0.5078 | 0.4844 | 0.6094 | 0.4766 | 0.4609 | 0.4844 |
| 0.5703 | 0.5469 | 0.4922 | 0.5000 | 0.5391 | 0.5313 | 0.5000 | 0.5703 |

TABLEV: STRICT AVALANCH MATRIX FOR RA

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.4844 | 0.5156 | 0.5703 | 0.5000 | 0.5156 | 0.5391 | 0.4375 | 0.5938 |
| 0.5313 | 0.5156 | 0.4453 | 0.5234 | 0.5859 | 0.4922 | 0.5234 | 0.4844 |
| 0.4531 | 0.4766 | 0.4766 | 0.5078 | 0.5547 | 0.4453 | 0.4766 | 0.4766 |
| 0.5625 | 0.4453 | 0.5625 | 0.4375 | 0.4766 | 0.4844 | 0.5313 | 0.5000 |
| 0.5078 | 0.5078 | 0.4766 | 0.5078 | 0.5000 | 0.5156 | 0.3984 | 0.4297 |
| 0.6094 | 0.5391 | 0.4531 | 0.5938 | 0.4609 | 0.5078 | 0.5156 | 0.5313 |
| 0.4844 | 0.4609 | 0.4844 | 0.4922 | 0.5391 | 0.4766 | 0.4766 | 0.5234 |
| 0.4922 | 0.4219 | 0.5000 | 0.4375 | 0.5781 | 0.5234 | 0.5859 | 0.5000 |
| 0.5469 | 0.5000 | 0.4922 | 0.5000 | 0.5625 | 0.5703 | 0.4453 | 0.6094 |
| 0.4141 | 0.5469 | 0.4609 | 0.5156 | 0.5313 | 0.4297 | 0.4922 | 0.4141 |
| 0.5078 | 0.5469 | 0.5234 | 0.4766 | 0.4844 | 0.5547 | 0.5547 | 0.5859 |
| 0.4766 | 0.4844 | 0.4766 | 0.5078 | 0.5391 | 0.5000 | 0.4922 | 0.4688 |
| 0.5391 | 0.4922 | 0.5078 | 0.4922 | 0.5234 | 0.5156 | 0.4531 | 0.4141 |
| 0.4219 | 0.5234 | 0.4844 | 0.4375 | 0.4844 | 0.4688 | 0.5156 | 0.5547 |
| 0.5156 | 0.4844 | 0.4922 | 0.5078 | 0.6172 | 0.5156 | 0.5547 | 0.5391 |
| 0.5547 | 0.5391 | 0.5313 | 0.4453 | 0.5313 | 0.5000 | 0.4453 | 0.5234 |

## VI. PERFORMANCE EVALUATION OF MRA

Evaluating the MRA is done compared to the RA through diffusion and confusion, throughput and memory size of the

proposed S-box.

### A. Diffusion and Confusion

The terms Diffusion and Confusion were introduced by Claude Shannon to thwart cryptanalysis based on statistical analysis. In the proposed S-box table, it is noted that each nibble in GF ($2^4$) has the same SubNibble in more than one of small S-boxes ($S_1$ to $S_{10}$), for example, the SubNibble for nibble 0x0 from $S_5$, and $S_9$ are the same which is 0xd. Also, the SubNibble for nibble 0x0 from $S_8$, and $S_{10}$ are the same which is 0xf and this fact is noted for all nibbles in S-box table and Inv S-box table. The analyst cannot detect the relation between the SubNibble and the nibble of the proposed S-box table; because each small S-box is constructed from equation differs from the others. So, detecting all equations is very difficult for the proposed S-box table while S-box table of RA has one equation.

So, the advantage of using small S-boxes for constructing the proposed S-box table is that the proposed S-box table has multiple different equations which cannot be detected easily when the relation between output and input is known. The key expansion procedure of RA is modified by applying the new approach which uses the proposed S-box table at the SubByte transformation step for extracting the round keys as discussed in section III; this makes the attempt to deduce the key is very difficult. Also, the substitution from the proposed S-box table depends on the round key makes the statistical relationship between the plaintext and the ciphertext as complex as possible. Thus, diffusion and confusion are achieved for MRA.

### B. Throughput

Any encryption algorithm causes decreasing in a system throughput due to the time consumed for the encryption and decryption process. The SubByte transformation using S-box causes the most of delay of the encryption algorithm. For increasing the throughput of an encryption algorithm, the delay caused by the S-box must be decreased. When the delay of the encryption algorithm is decreased, the encryption algorithm can be used in real time applications such as voice over Internet Protocol (VoIP), as in [12]. The MRA achieves lower delay than RA as proved below.

#### 1) For the MRA

The substitution is done nibble by nibble. For substituting each nibble from small S-box (16x1), assume the maximum time it takes =16 time unit, where 16 is the number of elements in each small S-box, assuming (16x4) memory size is used for each S-box. Thus the time for substituting one byte = 2*16 time unit = 32 time unit.

The time for 16 bytes =32 time unit* 16=512 time unit

The time for 10 rounds= 5120 time unit

#### 2) For RA

The substitution is done byte by byte. For substituting each byte from S-box table (16x16), assume the maximum time it takes =256 time unit, where 256 is the number of elements in S-box table.

Thus the time for substituting one byte = 256 time unit

The time for 16 bytes =256 time unit* 16=4096 time unit

The time for 10 rounds= 40960 time unit

From the above calculations, it is noted that the delay of SubByte transformation in MRA is equal 1:8 of the delay of SubByte transformation of RA. Thus, the throughput of MRA is very high than RA throughput. With reducing the number of rounds of MRA to half as mentioned above, the throughput can be duplicated.

### C. The memory Size

In MRA, the proposed S-box table is implemented as a look-up table in memory. The size of memory occupied for MRA is very small comparing with RA as calculated in the following.

#### 1) For MRA

The proposed S-box table consists of 10 small S-boxes. Each small S-box occupies a size of memory calculated as follows.

The size of memory for each small S-box = 16 *4 = 64 bits

The size of memory for 10 S-boxes=64*10=640 bits

#### 2) For RA

The S-box of RA occupies a size of memory calculated follows.

The size of memory for S-box = 256*8= 2048 bits.

From the above calculations, it is noted that the proposed S-box of MRA occupies less memory size than S-box of RA.

The above calculations are calculated by hand and the hardware implementation of MRA is out the scope of this paper.

## VII. Conclusion

In this paper, the new algorithm (MRA) with a proposed S-box constructed from small S-boxes is introduced. Using small S-boxes defined over GF ($2^4$) with different equations and different irreducible polynomials achieve diffusion, confusion, and security of MRA. In addition, Substitution from the proposed S-box table based on round key results more confusion and security for MRA. As shown from the results, the encryption time of MRA is lower than the encryption time of RA, this means that MRA achieves higher throughput than RA. Therefore, MRA can be used for real time applications such as VoIP. Also, by decreasing the number of rounds to half doesn't affect the security of MRA, but makes the MRA faster. As well as, the proposed S-box occupies smaller memory size than the S-box of RA; this makes it suitable for small size applications.

## References


[1] J. Daemen, V. Rijmen, "The block cipher Rijindael", in Proc. *Third International Conference on smart card Research and Applications, CARDIS'98, Lecture Notes in computer Science*, Berlin, vol.1820, Springer, 2000, pp. 277_284.

[2] J. Daemen and V. Rijmen, "The Design of Rijndael: AES – The Advanced Encryption Standard." Springer-Verlag, 2002.

[3] Federal Information Processing Standards Publications (FIPS 197), "Advanced Encryption Standard (AES)", vol. 26 Nov. 2001.

[4] William Stallings, *Cryptography and Network Security*, Prentice Hall, 2008.

[5] A. Fahmy, M. Shaarawy, K. El-Hadad, G. Salama, and K. Hassanain, "A proposal for A key-dependent AES", in *Proc. 3rd International Conference: Sciences of Electronic, SETIT 2005*, pp. 27-31, March, 2005 – TUNISIA.

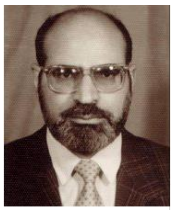
[6] Rohiem, Elagooz and H. Dahshan, "Anovel approach for designing the S-box of advance ecryption standard using chaotic map", *Radio science conference*, NRSC May 2005.

[7] Rudolf Lidl and Harald Niederreiter, *Introduction to finite fields and their applications*, Cambridge, New York, 2002.
[8] C. Paar, "Fast Arithmetic Architecture for Public-Key Algorithms over Galois Fields GF ((2n) m)", in *Proc. EUROCRYPT'97, LNCS* vol. 1233, pp. 363-378, Springer-verlag, 1997.
[9] A. Rudra *et al.*, "Efficient Implementation of Rijndael Encryption with Composite Field Arithmetic," in *Proc.CHES 2001, LNCS* vol. 2162, pp. 175-188, 2001.
[10] S. Morioka and A. Satoh, "An Optimized S-box Circuit Architecture for Low Power AES Design," in *Proc. CHES 2002, LNCS* vol. 2523, pp. 172-186, 2003.
[11] R. Forre. The strict avalanche criterion: spectral properties of booleans functions and an extended definition. Advances in cryptology, in: S. Goldwasser (Ed.), Crypto'88, Lecture Notes in Computer Science, vol. 403, *Springer-Verlag*, 1990, pp. 450–468.
[12] Faiz Yousif, Alaa Eldin Rohiem, and A. Elbayoumy, "Security evaluation of VoIP cryptographic algorithms", in *Proc. 6th ICEENG Conference on Electrical Engineering*, pp. 27-29 May, 2008.

**Hanem M. El-Sheikh** received B.Sc. degree in Electrical Engineering (Communication and Electrophysical Engineering Department) from Faculty of Engineering, Alexandria University, Alexandria, Egypt in 1985. She received the diploma degree in Electrical Engineering (Computer and Systems Engineering Department), Faculty of Engineering, Ain Shams University, Cairo, Egypt in 1997. She received M. Sc. degree in Electrical Engineering (Computer and Systems Engineering Department), Faculty of Engineering, Ain Shams University, Cairo, Egypt in 2005. She completed the Ph.D degree in Electrical Engineering (Electronics and Communication Engineering Department), Faculty of Engineering, Ain Shams University, Cairo, Egypt in 2012. She has worked in Switching Department at National Telecommunication Institute, Cairo, Egypt since 1997. She published papers in international conferences and received the best paper award in the area of VoIP technology and Cryptography in 2011. Her research interests are in network security, finite fields, cryptography, VoIP technology andsecurity, and digital hardware design. Dr. Hanem has CCNA Certificate and has been Cisco instructor (CCIE) since 2002.

**Omayma A. Mohsen** received B.Sc., M.Sc. and Ph. D degrees in Electronics and Communication Engineering in 1982, 1989 and 1997 respectively from Faculty of Engineering, Cairo University, Cairo, Egypt. Major Field of study was communication engineering, traffic in telecommunication networks and testing of hybrid telecommunication circuits. She joined National Telecommunication Institute since 1984. In 1989 she was promoted as teacher assistant. In 1997 she was Promoted as Associate Professor. Previous research study included Reliability in MPLS network, Multicast in Metro Ethernet network and VoIP Security. Current research study includes Mobility in MPLS Network, NGN QoS, VANET and Security in NGN. Dr. Omayma A. Mohsen is a member of the Switching Department, and member of the scientific committee at the National Telecommunication Institute. Dr. Omayma is an ITU local Expert since July 2010, and senior member of the IACSIT.

**Talaat Elgarf** received B.Sc. and diploma degrees in Electrical Engineering (Communication Engineering Department) from Military Technical College, Cairo, Egypt in 1976 and 1990 respectively. He received M.Sc. and Ph. D degrees in Electrical Engineering (Electronics and Communication Engineering Department), Faculty of Engineering, Ain Shams University, Cairo, Egypt in 1990 and 1993 respectively. Currently, he is professor of communications in Electrical Engineering department (Higher Technological Institute) at 10$^{th}$ of Ramadan City. He published papers in international conferences in the areas of Communications and Communication Security. He also supervised many Master theses and Doctorate in the area of Cryptographic Systems and VoIP Security. His research interests are in Digital Communication Security Systems, Cipher systems and VoIP security. Prof. Talaat is a member of ICT (Information and Communications Technology) Research Council of ASRT (Academy of Scientific Research andTechnology) since 2005.



**Abdelhalim Zekry** is a professor of electronics at faculty of Engineering, Ain Shams University, Cairo, Egypt. He worked as a staff member on several universities. He published more than 110 conference and periodical papers. He also supervised more than 60 Master theses and 17 Doctorate theses in the area of electronics and electronics for communications. He focuses his research programs on the field of microelectronics and electronic applications for communications. Prof. Zekry is a member of the IEEE and senior member of the IACSIT.