Journal of Fundamental and Applied Sciences

ISSN 1112-9867

Available online at http://www.jfas.info

Research Article

Special Issue

CURRENT IMPLEMENTATION OF ADVANCE ENCRYPTION STANDARD (AES) S-BOX

M. F.Roslan[1], K. Seman[1,*], A. H. A. Halim[2] and M N. S. M. Sayuti[1]

[1]Faculty of Engineering and Built Environment, Universiti Sains Islam Malaysia, 71800 Nilai, Negeri Sembilan, Malaysia

[2]Faculty of Science and Technology, Universiti Sains Islam Malaysia, 71800 Nilai, Negeri Sembilan, Malaysia

Published online: 05 October 2017

ABSTRACT

Although the attack on cryptosystem is still not severe, the development of the scheme is still ongoing especially for the design of S-Box. Two main approach has been used, which are heuristic method and algebraic method. Algebraic method as in current AES implementation has been proven to be the most secure S-Box design to date. This review paper will concentrate on two kinds of method of constructing AES S-Box, which are algebraic approach and heuristic approach. The objective is to review a method of constructing S-Box, which are comparable or close to the original construction of AES S-Box especially for the heuristic approach. Finally, all the listed S-Boxes from these two methods will be compared in terms of their security performance which is nonlinearity and differential uniformity of the S-Box. The finding may offer the potential approach to develop a new S-Box that is better than the original one.

Keywords: block cipher; AES; S-box.

Author Correspondence, e-mail: drkzaman@usim.edu.my

doi: http://dx.doi.org/10.4314/jfas.v9i4s.30

1. INTRODUCTION

The advance encryption standard (AES) is the most widely use symmetric cipher today. The term standard is supposed to use only for United States government application. However, AES block cipher also adapted in several industries standard and in many commercial systems. Several commercial standards that include AES are the Internet security standard IPSec, TLS, the Wi-Fi encryption standard IEEE 802.11i and numerous security products around the world. To date, there are several known theoretical attacks which are still impractical due to higher complexity as mention in [1].

AES was developed through competition held by National Institutes of Standard and Technology (NIST) of the United States in 1999. The competition aims to find the new cryptosystem for government use to replace the older data encryption standard (DES) cryptosytem, which was identified no longer viable for current implementation due to its security and implementation effectiveness. The present scheme of AES was originated from Rijndael algorithm that is developed by two Belgium cryptologist, which are Joan Daemon and Vincent Rijmen. Along with Rijndael algorithm, there were another four-finalist cryptosystem in the competition which named as Mars by IBM Corporation, RC6 by RSA Laboratorie, Serpand and lastly Twofish. Rijndael algorithm provides three options of key length, which are 128, 192 and 256 bit. On the existing AES implementation, only key of 128-bit length was used.

In every block cipher like AES, the only source of nonlinearity acts as the core structure of the cipher is the implementation of S-Box. This S-Box or substitution box provides confusion to obscure the relationship between the plaintext and the ciphertext. It will ensure that the small change in the plaintext will propagate quickly across the ciphertext. In the current AES implementation, the construction of S-Box is based on algebraic structure of finite field. It has been proven that this type of S-Box is still the best construction in terms of security against differential and algebraic attack [2]. However, the ongoing research on construction of S-Box has led to discovery of the new method which has similar security with better efficiency.

The rest of the paper is organized as follows. The complete structure of AES is reviewed in section II. In section III, the current construction of the AES S-Box is shown. While in section

IV and V, algebraic and heuristic method are discussed respectively. In section VI, some cryptographic properties of S-Boxes will also be discussed. Then, all types of S-Boxes shown in section IV and V will be compared with the current implementation. Lastly, section VII will conclude the paper.

2. AES STRUCTURE

The AES receives the plaintext x of 128-bits in the form of 4×4 state matrix A, where each matrix cell represents 8-bits or 1-byte of input. This plaintext will be operated with 128-bits symmetric key k in the same form of state matrix of plaintext and the output will be the ciphertext y. Fig. 1 shows the general structure of AES.

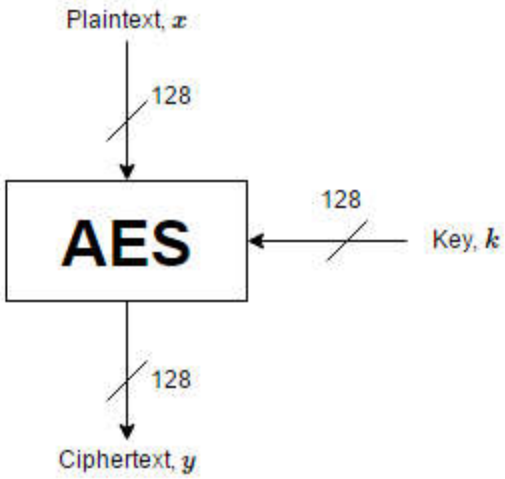

Fig.1. The general structure of AES

The state matrix will then undergo 10 rounds of encryption where each round consists of four layers data manipulation. The four layers are named as SubByte, ShiftRow, MixColumn and AddRoundKey layer. The details of each layer are as follows:

- SubByte: The element in the state matrix A is substituted with another element from the S-Box lookup table. The nonlinearity of the S-Box will introduce confusion to the plaintext and any small change in the plaintext will cause big altering to the output ciphertext. The output of this process is recognized as state matrix B. The details of this process will be explained in section III.
- ShiftRow: The state matrix B will be permuted by shifting for a certain amount by row. This layer make change to all rows except the first row.

- MixColumn: The output state matrix from ShiftRow operation will be operated with the constant MDS matrix. However, this layer is not available for the last encryption round.
- AddRoundKey: The output of MixColumn is then XORed with the sub key that was generated by key schedule. The flow details of all AES structure including the encryption and decryption process are shown in Fig. 2.

However, in the original design of Rijndael algorithm, three key length option was available which were 128-bits, 192-bits and 256-bits. NIST only adopted 128-bits key length for standard use. For longer key bits, more round of encryption needed, which are 12 and 14 round for 192-bits and 256-bits respectively.

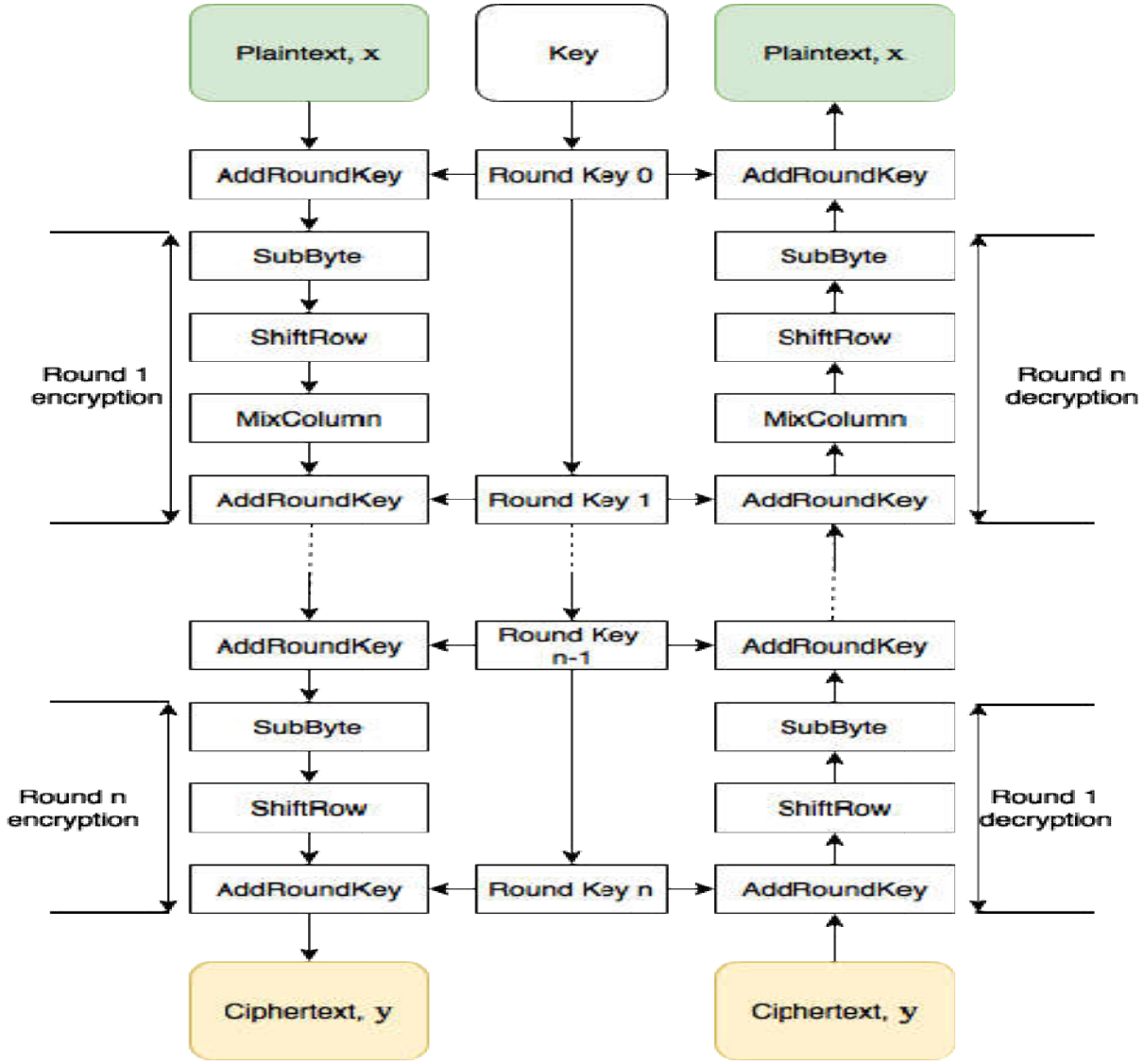

Fig.2. The flow structure of AES

3. CURRENT IMPLEMENTATION OF AES S-BOX

The core of any block cipher like AES is the creation of S-Box or substitution box. The construction of current AES S-Box involves two processes, which are finite field inversion and the affine transformation as shown in Fig. 3. To increase efficiency, all the result of these two processes on every element in $GF(2^8)$ will be stored in a lookup table called as S-Box. The existence of S-Box as the only nonlinear part of block cipher provides confusion to the input text and obscures the relationship between plaintext and the ciphertext.

Fig.3. The original construction of AES S-Box which composed of $GF(2^8)$ and affine transformation

In current implementation, this source of confusion is originated from the algebraic complexity of arithmetic operation in finite field $GF(2^8)$. This field can be considered as the extension field of $GF(2)$, which only have two elements $\{0,1\}$. In the extension field $GF(2^8)$, the operation between element cannot be done in binary arithmetic. Instead, the elements of the field need to be represented as the polynomial with maximum degree of 7. Thus, from this definition, each element can be shown as follow

$$A(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0, a_i \in GF(2) = \{0,1\} \quad (1)$$

This representation also fit nicely to all element $GF(2^8)$ as every polynomial can be stored as 8-bit tuple as follows

$$\text{A}=(\text{a}_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0) \quad (2)$$

The $GF(2^8)$ inversion involve the operation of finding inverses for all element in that field. There are 256 elements including 0. We can define this field inversion as follows

$$F(x) = \begin{cases} x^{-1} & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases} \quad (3)$$

From Equation (3), the inversion of every element are defined by using multiplicative modulo

$p(x)$ where $p(x)$ is the primitive polynomial in $GF(2^8)$. The element 0 is mapped to itself. In the existing implementation, $p(x) = x^8 + x^4 + x^3 + x + 1$ has been used. In this paper, this primitive polynomial will be notated in hexadecimal notation $\{11b\}$. This will also be applied to other type of primitive polynomial.

The selection of function like in Equation (3) was originated from the previous SHARK and Square cipher developed by the same developer of Rijndael. These two cipher adapt the idea from [3], where highly nonlinear permutation function is introduced. However, the use of non-permutation function as in Equation (3) alone does not enough as the S-Box function can be expressed as rational number of low degree [4].

To bypass the drawback face by SHARK and Square cipher, while maintaining the use of highly nonlinear permutation function as in Equation (3), Rijmen and Daemen introduced the new composition of nonlinear permutation function with simple linear transformation (Fig. 3). This composition is motivated by the wide-trail strategy introduce in [5]. This strategy aims to hide the trail of linear transformation and achieving low difference propagation probabilities in order to secure the cipher against differential and linear cryptanalysis [6]. In this composition, each element of $GF(2^8)$ can be viewed as the element of product modulo $p(x)$ as follows

$$GF(2^8) \cong Z_2[x] / (x^8 + x^4 + x^3 + x + 1) \tag{4}$$

where all elements $A(x) \in GF(2^8)$ except zero is mapped to its inverses $A(x)^{-1} \in GF(2^8)$ and the element 0 is mapped to itself. Then, the output of the $GF(2^8)$ inverses will undergo the linear affine transformation as follows

$$\begin{aligned} B_i = {} & (x^7 + x^6 + x^5 + x^4 + 1)A(x) + \\ & (x^7 + x^6 + x^3 + x^2) \bmod (x^8 + 1) \end{aligned} \tag{5}$$

The affine transformation as in Equation (5) often represented as bitwise matrix form as follows

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b_{0'} \\ b_{1'} \\ b_{2'} \\ b_{3'} \\ b_{4'} \\ b_{5'} \\ b_{6'} \\ b_{7'} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \text{mod2} \tag{6}$$

In this bitwise representation of Equation (6), the output of $GF(2^8)$ inversion and affine transformation are notated as $B' = (b_{0'},...,b_{7'})^T$ and $B = (b_0,...,b_7)^T$ respectively. This construction produces S-Box with no fixed point and opposite fixed point, thus maintaining the nonlinearity of the permutation function as in Equation (3). This composition increase the confusion to the plaintext and as the consequences the attack complexity become higher. The significant of the affine transform is to destroy all the algebraic structure of $GF(2^8)$, which in turn prevent the cipher from the algebraic attack [7].

By all the above operation, the S-Box construction can be denoted as the composed function where the $GF(2^8)$ denoted as F and linear affine transformation denoted as L as follows

$$S = H \circ L \circ F \tag{7}$$

where H is the additive constant in the affine transformation. This additive constant $\{63\}$ remove all fixed point $S(a) = a$ and the opposite fixed point $S(a) = \overline{a}$. The setting of Equation (7) can be analyzed using simple algebraic operation to reveal its unexpected algebraic properties that were not deeply studied at the beginning of the design time. This analysis will produce one single equation called as the algebraic expression which later taken as the source of modification of S-Box. The derivation of this expression using Equation (3) and the affine matrix in Equation (6). From Equation (3), we can define the field inversion of every nonzero element in $GF(2^8)$ as $x^{-1} = x^{2^8-2} = x^{254}$. Thus, we can rewrite Equation (3) as

$$F(x) = x^{254} \tag{8}$$

As for linear affine matrix L, it is linear map thus it can be expressed as a linearized polynomial with eight terms as follows

$$L(x) = \sum_{i=0}^{7} \lambda_i x^{2^i} \tag{9}$$

From Equation (9), the linearize polynomial can be derived into

$$\begin{aligned} L(x) = {} & \{8f\}x^{128} + \{b5\}x^{64} + \{01\}x^{32} + \{f4\}x^{16} \\ & + \{25\}x^{8} + \{f9\}x^{4} + \{09\}x^{2} + \{05\}x \end{aligned} \tag{10}$$

The coefficient of the above linearize polynomial can be derived using several methods. In [8] derived the equation using the method of q-polynomial, while in [9] used partition equivalence in order to resolve the coefficient of the expression. Another paper which discussed in brief about the expression like in [10] used the approach of the trace function.

The last part of the AES S-Box lies within the affine transformation which is the additive constant $\{63\}$ that can be denoted as function H in Equation (7). Combining Equation (8) and Equation (10) with respect to AES S-Box composition in Equation (7), the algebraic expression of the S-Box can be shown as follows

$$\begin{aligned} S(x) = {} & \{63\} + \{8f\}x^{127} + \{b5\}x^{191} + \{01\}x^{223} \\ & + \{f4\}x^{239} + \{25\}x^{247} + \{f9\}x^{251} \\ & + \{09\}x^{253} + \{05\}x^{254} \end{aligned} \tag{11}$$

This final form of S-Box algebraic expression has risen up several questions among cryptologist community about its security as it has only nine terms, which might be the possible source of known cryptanalysis such as interpolation attacks.

It is of interest to review every possible way to construct and modify the current implementation of AES S-Box. The construction that is originated from finite field operation and algebraic representation will be considered as the algebraic method. This method is believed to be the most secure way to generate S-Box as it achieves the best cryptographic properties. Meanwhile, there is another way other that algebraic method which may produce equivalence strength which is heuristic method. Both methods will be discussed separately in the next two section.

4. ALGEBRAIC APPROACH

4.1. Replacement of Primitive Polynomial and Affine Transformation

The simplest way to construct the new S-Box is by replacing the current primitive polynomial

$p(x)$ with another primitive polynomial in the field $GF(2^8)$ In [11], other primitive polynomials in the same field and additive constant has been shown able to generate unknown S-Box while maintaining the original structure of AES, thus preventing the known cryptanalysis. In that paper, the author listed all the 30 irreducible polynomials in $GF(2^8)$ that is possible for the construction of S-Box. The list of the primitive polynomial in this field is shown in Table 1. Besides the primitive polynomials, the author also presented a list of additive constants in affine transform which can be substituted in the original AES S-Box. This list consists of 36 different elements in $GF(2^8)$, which will produce the S-Box with no fix point and the opposite fix point. The list of the additive constant is shown in Table 2. Using the list in Table 1 and Table 2, several combinations can be made which will produce different S-Box that have the same or even better security performance than the existing one. This combination also can possibly be part of the key that will strengthen the cryptosystem. The author has tried several combinations by using the NIST statistical test suite and found several combinations that is even better than the current implementation.

Table 1. List of irreducible polynomials for AES S-Box

List of Irreducible Polynomials in $GF(2^8)$					
11b	13f	169	18b	1b1	1dd
11d	14d	171	18d	1bd	1e7
12b	15f	177	19f	1c3	1f3
12d	163	17b	1a3	1cf	1f5
139	165	187	1a9	1d7	1f9

Table 2. List of possible additive constant

List of Possible Additive Constant in $GF(2^8)$					
0a	35	62	ea	c7	9d
0f	38	6e	d5	bf	91
15	40	74	d4	b5	2b
2a	4a	7e	ce	b1	81
2b	4e	f5	cd	ab	31
31	54	f0	ca	a1	5e

Another uncomplicated way to reconstruct new S-Box, while maintaining the algebraic structure is by replacing the linear affine matrix as shown in Equation (6). This matrix in polynomial form as in Equation (5) which is $x^7 + x^6 + x^5 + x^4 + 1$. It is possible this polynomial is replaced by another polynomial as long as the matrix produced is non-singular. There are about 2^{63} possible polynomials for each primitive polynomial listed in Table 1 that can be used to generate affine matrix for the construction of S-Box. This possible polynomial is bounded from $[01;02;04;08;10;20;40;80]^T$ for lower end to $[fe;7f;bf;df;ef;f7;fb;fd]^T$ for higher end. Using the new combination of affine matrix with various primitive polynomial may improve the avalanche effect of the ciphertext. This has been proven in the same paper where four primitive polynomials which are $14d, 165, 17b$ and $1bd$ with various combination of affine matrix will produce the S-Box with full avalanche effect. Another paper which discuss the same method of modification is in [12], where different affine matrix was used. The end result shows that the new modified S-Box pass the randomness test.

4.2. Algebraic Expression Modification

The simple modification above seems convincing where they might have even better security than the existing one. Nevertheless, this type of modification does not improve much as the algebraic representation is still in the same condition where only nine terms are involved and thus still exposing the cipher to several known attacks such as interpolation attack. We rephrase the algebraic expression of AES S-Box as in Equation (11) from the previous section can be simplified as the summation based on [13-14] as follows

$$S(x) = w_8 + \sum_{d=0}^{7} w_d x^{255-2^d} \tag{12}$$

for certain constant $w_0,...,w_8$. Several researchers found that this is the only disadvantage of AES, which might be the source of attacks. Although in [14] it has been shown that the whole AES has very complex algebraic structure, it is of interest for cryptologist community to improve the AES S-Box in order it can resist the known attacks for a longer time.

In previous section, we have highlighted the issue of algebraic expression which only consist of nine terms. This setting will remain even though the primitive polynomial or the affine transformation is replaced. However, the replacement of the affine matrix may change the coefficient of the terms in that expression. This representation is said to be extremely sparse

multivariate quadratic equations as it has the highest degree of 254 but involve only nine terms [15]. To improve this setting, in [16] introduce new way to construct the S-Box. The new way is by switching the order of operation between affine transformation and $GF(2^8)$ inversion as in Fig. 4.

Fig.4. New S-Box structure

This simple setting yield an algebraic expression with the highest degree of 254 and 255 terms involved. This significant improvement is due to the simple modification of the composed function as in Equation (7). This new composition can be shown as follows

$$S = H \circ F \circ L \tag{13}$$

Then, using this Equation (13), the new algebraic expression of the AES S-Box is

$$\begin{aligned} S(x) &= (\{8f\}x^{128} + \{b5\}x^{64} + \{01\}x^{32} + \{f4\}x^{16} \\ &\quad + \{25\}x^8 + \{f9\}x^4 + \{09\}x^2 + \{05\}x)^{254} + \{63\} \\ &= \{05\}x^{254} + \{93\}x^{253} + \{fd\}x^{252} \\ &\quad + \cdots + \{94\}x + \{63\} \end{aligned} \tag{14}$$

However, they are several varieties using this type of modification especially the position of additive constant. This varieties will not change the setting of algebraic expression with full 255 terms. For example, the same author in [17] remodified the composition as in Equation (13) to

$$S = F \circ L \circ H \tag{15}$$

where the first operation that acts on the input plaintext is the additive constant followed by affine matrix and field inversion. This simple modification has the same security aspect as the previous modification.

In [18], the author has come out the comparison of three cases of S-Box composition function. The first case is the existing implementation of S-Box which is consider too sparse. The second case introduce some modification same as previous mentioned paper but with slightly different. In this case, the input plaintext will undergo linear affine transformation first

followed by additive constant and ended by field inversion. This composition can be represented as

$$S = F \circ H \circ L \tag{16}$$

The composition in Equation (16) maintained the setting with 255 terms and the algebraic degree of 254. Although the algebraic expression improves a lot, there is still one drawback due to this simple modification. All of S-Box function compositions mentioned in Equation (13), Equation (15) and Equation (16) improve the encryption side of expression while downgrading the decryption expression from 255 terms to only nine terms. The third cases fixed this problem by applying two affine transformation which is before and after field inverses operation. This configuration is shown in Fig. 5.

Fig.5. S-Box structure with two affine transformation

The use of two affine transform for AES was first introduced in [10] and motivated by the Camelia cipher introduced in year 2000 [19]. By applying two affine transformation, the encryption and decryption algorithm will go through the same step of processes. This configuration will yield an algebraic expression with 255 terms and highest degree of 254 for both encryption and decryption algorithm. In composition form, two affine process as in Fig. 5 can be expressed as follows

$$S = L \circ H \circ F \circ H \circ L \tag{17}$$

However, two affine transformation decrease the construction efficiency of the S-Box as more process needed. As for the security, this implementation is better than the other two cases.

Another two variants of the composition Equation (17) were stated in [20-21]. In [20], emphasize the drawback of improvement made by [22] which reduce the algebraic expression terms from 255 to 9 for decryption algorithm. Using the same approach made by [10], the author add two affine transform before and after the field inversion. The constructed S-Box was then tested using six cryptographic characteristics which are bijection, strict avalanche, algebraic degree, nonlinearity, differential uniformity and algebraic complexity. All aspects

can be said comparable to existing AES except the improvement in algebraic complexity for the new implementation.

In [21], the first affine transformation was adapted from binary Gray code transformation. This code is originally designed to spurious output from electromechanical switch. It aims is to make two successive values differ in one digits only. In bitwise form, it can be shown as follows

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} b_{0'} \\ b_{1'} \\ b_{2'} \\ b_{3'} \\ b_{4'} \\ b_{5'} \\ b_{6'} \\ b_{7'} \end{pmatrix} \tag{18}$$

while for the second affine transform after the field inversion is the same as in the existing implementation. This improvement make the algebraic expression to have 255 terms with highest degree of 254. The advantage of this implementation not only in term of algebraic expression but it is reusable from the original implementation. Using Gray code, the output S-Box can be said to has comparable security compared to existing one.

Similar approach as Gray code has been summarize in [23] by the same author. A new way has been proposed to find the affine matrix using the graph isomorphism. Using this method, there are about 5.34×10^{18} possible S-Box that can be generated. Five sample of S-Box has been shown in the paper and two of them have fulfilled the desired cryptographic characteristics.

5. HEURISTIC APPROACH

The modification of AES S-Box not only limited to manipulating primitive polynomial and field inversion as mentioned in previous section. The ongoing research has shown that the S-Boxes also can be modified using several other methods, which can be classified as heuristic approach. There are several of method of obtaining S-Boxes under heuristic approach. This paper will classify them into two part, which are chaotic maps approach and non-chaotic map approach.

5.1. Chaotic maps S-Box

Chaotic maps are the maps that exhibits some chaotic behaviour which seems to be useful for generating good S-Boxes. This chaotic behaviour is highly unpredicted which have random look deterministic nature and usually use in the field of nonlinear dynamical system. Using chaotic maps, there are four step methods to construct cryptographically strong S-Boxes which are choosing chaotic maps, discretising the maps, key scheduling and cryptanalysis [24]. One example of chaotic maps that is also the most common used is the logistic map, which can be expressed as follows

$$\tau(x) = \mu x(1-x) \tag{19}$$

The parameter μ range is $[0,4]$. However, the chaotic behavior only exists when $\mu \geq 3.56995$ and the graph will diverge when this parameter exceeds 4. The problem exists when this type of map or any other chaotic maps only deals with real number while in cryptographic, especially for S-Box deals with integer. Some example of mapping from real number to integer can be found in [25]. This set of converted integers usually used as the initial table range from 0 to 255. This initial table is then permuted several times more using second chaotic maps or any other suitable method to achieve good cryptographic characteristic S-Boxes. The method of acquiring the initial binary integer is by denoting the floating-point number x as

$$x = 0.b_1 b_2, ..., b_i, .., \; x \in I = [0,1] \tag{20}$$

where the Ith bit $b_i(x)$ can be expressed as

$$b_i(x) = \sum_{r=1}^{2^i-1} (-1)^{r-1} \theta_{(r/2^i)}(x) \tag{21}$$

and $\theta(x)$ is a threshold function define as

$$\theta(x) = \begin{cases} 0, & x < t \\ 1, & x \geq t \end{cases} \tag{22}$$

Finally, the original real number generated using chaotic map as in Equation (19) can be expressed as integer binary sequence $B_i^n = f(x) = \{b_i(\tau^n(x)\}_{n=0}^{\infty}$. In the same paper, the author

used another chaotic map which is Bake Map to reshuffle the initial permutation table several times.

Another example like in [26] use the combination of cubic map, discrete chaos maps system (DCMS) and the affine transformation. The implementation is similar with [25] where the initial set of permutation is generated using cubic map. The only difference is the implementation of affine transformation like in the original AES S-Box. In [27], the construction of good S-Box was done using genetics algorithm which consist of selection, crossover adjustment and mutation operators. Like in previous example, the initial population (set of S-Box) which consist of individual S-Box was generated first via logistic and Bake map. Then, using the genetic algorithm, a stronger S-Box was selected and the strength is even better compared to previous implementation in [25-26].

In more recent paper like [28], a more complicated map was used. This paper proposes the use of discrete chaotic map based on the composition and permutation. The permutation operation was motivated by [29] code. Using that code, the author developed a new one dimensional chaotic map in [30]. Using this map, a new S-Box was developed and has better cryptographic properties compared to S-Box in [25-27]. However, the same author also proposed another S-Box that based on the composition of classical chaotic map such as logistic, Chebyshev and tent map in [31]. The resulted S-Box is much better in term of nonlinearity compared to [28]. While in [32], the new S-Box was generated using another complex chaotic map. The generated S-Box was based on tangent delay for elliptic cavity chaotic sequence. All real number generated by that map was mapped to 16×16 S-Box matrix. Based on the security analysis, the generated S-Box has low nonlinearity compared to another mentioned chaotic map S-Box before.

The permutation of S-Box element using chaotic maps can be change by controlling the parameter of the map. For example, in logistic map Equation (19), the control parameter is variable μ and initial value x_0. Due to this condition, more than thousand S-Boxes can be generated by using chaotic map. In [33], the author introduced a method of teaching-learning based optimization (TLBO) to choose a proper value for chaotic map parameters (μ, x_0). The optimization algorithm relies on the population of solution, (choice of parameters) to search

for global solution. The global solution is believed to generate the best output permutation of S-Box. Using this optimization method, together with Henon and logistic map, the generated S-Box can be said better than the random chosen parameter implemented in [25].

Another interesting usage of chaotic map in generating S-Box can be found in [34] with the implementation of 3-dimensional chaotic system. In this paper, the author used the 3-D four wing autonomous chaotic system to produce several number of S-Box. Due to freedom to change the initial point of each variable in the system, the author has developed about 500 different S-Box and run the performance analysis on each of the S-Box. The result shows that the generated S-Boxes has good cryptographic properties but not as good as the implementation in [31]. One other example that implemented the same approach can be found in [35] used the Zhangtong chaotic system. To generate S-Box, the author used the RNG from the chaotic system and follow some predefine algorithm to produce all 256 elements of S-Box. The generated S-Box is as good as [34].

5.2. Non-Chaotic Maps S-Box

Aside from chaotic maps approach, there are various other ways to construct AES S-Box. Usually, it involves more than one combination of method to acquire all permutation of 256 elements of S-Box. The development of that method is done by stage. In 2013, in [2] proposed the S-Box constructed from non-permutation power function, which is totally opposite to the original AES S-Box. Using this kind of function will generate several redundant elements in the S-Box. The author took the composition of two power function to form a binomial power function from different cyclotomic classes motivated from [36] in the form of

$$\alpha F_1 + \beta F_2 \tag{23}$$

where $\alpha, \beta \in GF(2^8)$. The set of non-permutation function F_1 and F_2 are chosen so that it has higher nonlinearity. To deals with redundant elements, the author introduced a new algorithm named as the redundancy removal algorithm (RRA). The RRA will classify the element into two types which are the set of redundant elements and the set of non-existence element. The redundant element is then replaced with the non-existence element beginning with the smallest entry of redundant elements. The output of RRA will be one desired bijective S-Box

which is comparable to most chaotic map generated S-Box in terms of security.

The same author made another improvement to Equation (23) in [37] by filtering the combination of power function where only power functions with higher nonlinearity and fewer redundant elements were selected. Besides that, two operation has been added which are addition with another power function and multiplication with coefficient. Based on this filtering method and additional operation, only three S-Boxes produced corresponded to two trinomial power function and one binomial power function. The S-Boxes from trinomial power function perform better compared to the other S-Box from binomial power function.

Another varieties of constructing S-Box can be seen in [38] where the constructed S-Box were generated from the bee-waggle dance pattern. Bee waggle dance are the pattern of movement of bee to communicate or send information about food location, water source and new-site location to its colonies. This pattern give information about the current position of sun, directions and speed of wind. Combining this two information, the colonies can identify the desired location together with the force or distance required to get there. In order to construct good S-Box, the author initiated the permutation using the trinomial function composed of three power function. Motivated by pattern of dance with eight different direction and 16 different distance, about 108 S-Boxes has been produced. From that number, only 24 S-Boxes can be considered as strong due to higher nonlinearity.

The best S-Box is even better compared to [2]. While, in [39] also by the same author, the construction of S-Box was done by composition of method RRA and bee-waggled dance introduced in [2-38] respectively. The resulted S-Box has lower security compared to both RRA and bee-waggle dance pattern.

Some construction of S-Box is motivated from an algorithm that based on the biological system. One example can be found in [40]. In this paper, the algorithm was inspired by artificial immune algorithm family, which also referred as clonal selection algorithm. This algorithm mimicking the behavior and capabilities of antibodies in the acquired immune system. Using this algorithm, the initial generated S-Box will undergo somatic mutation function. The iteration will run several times and maximum time taken will be 10 days. While in [41], the same author introduced another biological approach method to construct S-Box.

The method called as reversed genetic algorithm. The reversed term referred to a process of selecting good S-Box from a pool of S-Box generated by the finite field inversion process as implemented in original AES S-Box. This pool of S-Box will then go through a genetic algorithm which consists of breeding function, mutation function and lastly the fitness function. The breeding function will receive a pair of S-Boxes act as the parent and undergo the crossover process to produce two child S-Boxes. These child S-Boxes will be tested for their bijectivity and nonlinearity in the next two stated function. The output S-Box can be said comparable to chaotic map and waggle bee dance S-Box as stated before.

5.3. Key-dependent S-Box

Some paper discussed the construction of dynamic S-Box that has dependency to the secret key. In [42], the author proposed two method of implementation. The first one, or less secure one, all the element in the S-Box will be XORed with the first byte of key. The second method is more secure as the 16-byte key will be XORed first to become 1-byte key. This 1-byte key is then XORed with all the element in the S-Box. The initial S-Box that is used by the author is in the existing AES. By using this way, the S-Box itself will become unknown to the adversary. However, this implementation seems unsecure especially for the first method due to the risk of unintentionally exposing part of the key to the attacker. Different approach of key dependent S-box has been implemented in more recent paper [43]. In this paper, the author proposed that each 16 bytes of key will be XORed with each 16 bytes of S-Box row. The risk of exposing the key was diminished by applying again the affine transformation to the resulted S-Box.

6. S-BOX SECURITY COMPARISON

Previous two section has listed out several examples of S-Boxes construction over finite field $GF(2^8)$. The security of these S-Boxes can be verified at least using two cryptographic properties, which are high nonlinearity (NL) and low differential uniformity (DU).

6.1. Nonlinearity (NL)

The nonlinearity of an S-Box is the main properties of the construction as it provides confusion to the input plaintext. This property measure the distance between the set of all

affine function L_n over $GF(2^8)$ and Boolean function $f: GF(2^n) \to GF(2)$ and can be expressed as

$$N_f = \min_{l \in L_n} d_h(f, l) \tag{24}$$

where d_h is the hamming distance. The higher the value of N_f, the better the S-Box nonlinearity. S-Box that is generated through finite field inversion are believed to have the best value of nonlinearity, which is 112.

6.2. Differential Uniformity

The differential uniformity of bijective $n \times n$ S-Box is defined as

$$DU(F) = \max_{a,b \in \mathbb{F}_2^n} |\{x \in \mathbb{F}_2^n : F(x+a) + F(x) = b\}| \tag{25}$$

This property take the largest value of difference distribution table. The ideal value should be between 2 and 4. For AES S-Box, the DU is 4.

The S-Box that is generated using finite field inversion is expected to have the same security as the original one. However, for S-Box from heuristic method, the average value for both properties are less than the original AES S-Box. All type of S-Boxes with its NL and DU value is listed in Table 3 for comparison. Some paper listed more than one S-Boxes. The best one will be listed in Table 3.

Table 3. S-Box comparison

Proposed S-Box	NL	DU	Technique
AES S-Box	112	4	Finite field inversion
[16]	112	4	Modified finite field inversion
[18, 20-21, 23]	112	4	Two affine transform with finite field inversion
[25]	104	10	Logistic map and Bake map
[26]	112	12	DCMS and affine transformation
[27]	108	10	Chaotic map and genetic algorithm
[28]	106	10	Chaotic map and composition method
[31]	108	8	Chaotic map and composition method
[32]	108	12	Elliptic curve chaotic sequence
[33]	110	10	Chaotic map with TLBO

[34]	108	10	3-D four wing autonomous chaotic system
[35]	110	10	Chaotic scaled Zhongtang System
[37] S-Box 2	108	6	Trinomial power function
[37] S-Box 3	106	6	Binomial Power function
[38]	108	6	Bee Waggle Dance
[39] Opt 1	102	6	Hybrid Heuristic
[39] Opt 2	104	6	Hybrid Heuristic
[40]	104	6	Modified Immune algorithm
[41] S-Box 4	110	2	Reversed genetic algorithm
[41] S-Box 5	112	6	Reversed genetic algorithm

6.3. Remarks

A summary of security performance comparison is given in Table 3. Based on this comparison, we can conclude that the overall performance of S-Box from heuristic approach is still under par compared to original AES S-Box. However, there are some construction that has comparable NL and better DU compared to original AES S-Box. S-Box that is generated through chaotic map alone may not produce better S-Box due to the random sequence that is originated from random chosen parameters. However, when there is combination of chaotic map and other structured method, we can see that the resulted S-Box has the security performance that is approaching to original AES S-Box. Using some hybrid of two method may not produce the best S-Box compared to single method. This has been shown in [39] where two method from [37-38] combined to produce one hybrid method. Meanwhile S-Box that is generated using finite field inversion still maintained the security performance. However, there is some drawback related to implementation efficiency when there are two affine transform involves. Lastly, for S-Box that has dependency to secret key, the security performance may vary as it based on user input secret key. Besides, some implementation may have a risk of exposing the part of the key to adversary.

7. CONCLUSION

In this paper, we have listed out several implementations of S-Box. This implementation can

be classified into two approach which are algebraic approach and heuristic approach. From the comparison of S-Box in Table 3, the S-Box generated using finite field inversion have the best security compared to heuristic method. However, the search for S-Box using heuristic method that has the same security as finite field inversion method is still an open problem.

8. ACKNOWLEDGEMENTS

The authors would like to express special thanks to Ministry of Higher Education (MOHE), Malaysia for supporting and financing this research project under the Transdisciplinary Research Grant Scheme (TRGS/FKAB/50216/59). The author would also thanks to anonymous reviewers for their helpful comment and suggestion.

9. REFERENCES

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

[2] Isa H, Jamil N, Z'aba M R. S-box construction from non-permutation power functions. In ACM 6th International Conference on Security of Information and Networks, 2013, pp. 46-53

[3] Nyberg K. On the construction of highly nonlinear permutations. In Workshop on the Theory and Application of Cryptographic Techniques, 1992, pp. 92-98

[4] Landau S. Polynomials in the nation's service: Using algebra to design the advanced encryption standard. American Mathematical Monthly, 2004, 1:89-117

[5] Daemen J, Rijmen V. The wide trail design strategy. In IMA International Conference on Cryptography and Coding, 2001, pp. 222-238

[6] Nyberg K. Perfect nonlinear S-boxes. In Advances in Cryptology-Workshop on the Theory and Application of Cryptographic Techniques, 1991, pp. 378-386

[7] Paar I C, Pelzl I J. The advanced encryption standard (AES). In Understanding cryptography. Berlin: Springer, 2010, pp. 87-118

[8] Jingmei L, Baodian W, Xinmei W. New method to determine algebraic expression of Rijndael S-box. In 3rd ACM international conference on Information security, 2004, pp.

181-185

[9] Baodian W, Dongsu L, Wenping M, Xinmei W. Property of finite fields and its cryptography application. Electronics Letters, 2003, 39(8):655-656

[10] Sakallı M T, Aslan B, Buluş E, Mesut A Ş, Büyüksaraçoğlu F, Karaahmetoğlu O. On the algebraic expression of the AES S-Box like S-Boxes. In F. Zavoral, J. Yaghob, P. Pichappan, & E. El-Qawasmeh (Eds.), Networked digital technologies. Springer: Berlin, 2010, pp. 213-227

[11] Das S, Zaman J U, Ghosh R. Generation of AES S-Boxes with various modulus and additive constant polynomials and testing their randomization. Procedia Technology, 2013, 10:957-962

[12] Zakaria N H, Mahmod R, Udzir N I, Zukarnain Z A. Enhancing advanced encryption standard (AES) S-box generation using affine transformation. Journal of Theoretical and Applied Information Technology, 2015, 72(1):18-22

[13] Daemen J, Rijmen V. AES proposal: Rijndael. 1999, http://www.cs.miami.edu/home/burt/learning/Csc688.012/rijndael/rijndael_doc_V2.pdf

[14] Ferguson N, Schroeppel R, Whiting D. A simple algebraic representation of Rijndael. In S. Vaudenay, & A. M. Youssef (Eds.), Selected areas in cryptography. Springer: Berlin, 2001, pp. 103-111

[15] Murphy S, Robshaw M. Essential algebraic structure within the AES. In Advances in Cryptology-Workshop on the Theory and Application of Cryptographic Techniques, 2002, pp. 1-6

[16] Liu J, Wei B, Cheng X, Wang X. An AES S-box to increase complexity and cryptographic analysis. In 19th IEEE International Conference on Advanced Information Networking and Applications, 2005, pp. 724-728

[17] Jingmei L, Baodian W, Xinmei W. One AES S-box to increase complexity and its cryptanalysis. Journal of Systems Engineering and Electronics, 2007, 18(2):427-433

[18] Karaahmetoğlu O, Sakallı M T, Buluş E, Tutănescu I. A new method to determine algebraic expression of power mapping based S-boxes. Information Processing Letters, 2013, 113(7):229-235

[19] Aoki K, Ichikawa T, Kanda M, Matsui M, Moriai S, Nakajima J, Tokita T. Camellia: A 128-bit block cipher suitable for multiple platforms-design and analysis. In 7th Annual International Workshop on Selected Areas in Cryptography, 2000, pp. 39-56

[20] Cui L, Cao Y. A new S-box structure named Affine-Power-Affine. International Journal of Innovative Computing, Information and Control, 2007, 3(3):751-759

[21] Tran M T, Bui D K, Duong A D. Gray S-box for advanced encryption standard. In IEEE International Conference on Computational Intelligence and Security, 2008, pp. 253-258

[22] Jing-Mei L, Bao-Dian W, Xiang-Guo C, Xin-Mei W. Cryptanalysis of Rijndael S-box and improvement. Applied Mathematics and Computation, 2005, 170(2):958-975

[23] Tran B N, Nguyen T D, Tran T D. A new S-box structure based on graph isomorphism. In IEEE International Conference on Computational Intelligence and Security, 2009, pp. 463-467

[24] Jakimoski G, Kocarev L. Chaos and cryptography: Block encryption ciphers based on chaotic maps. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 2001, 48(2):163-169

[25] Tang G, Liao X, Chen Y. A novel method for designing S-boxes based on chaotic maps. Chaos, Solitons and Fractals, 2005, 23(2):413-419

[26] Xu G, Zhao G, Min L. The Design of dynamical S-boxes based on discrete chaos map system. In IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009, pp. 473-478

[27] Wang Y, Wong K W, Li C, Li Y. A novel method to design S-box based on chaotic map and genetic algorithm. Physics Letters A, 2012, 376(6):827-833

[28] Lambić D. A novel method of S-box design based on chaotic map and composition method. Chaos, Solitons and Fractals. 2014, 58:16-21

[29] Lehmer D H. Teaching combinatorial tricks to a computer. In Symposium in Applied Mathematics Combinatorial Analysis, 1960, pp. 179-193

[30] Lambić D. A new discrete chaotic map based on the composition of permutations. Chaos, Solitons and Fractals, 2015, 78:245-248

[31] Lambić D. A novel method of S-box design based on chaotic map and composition

method. Chaos, Solitons and Fractals, 2014, 58:16-21

[32] Alkhaldi A H, Hussain I, Gondal M A. A novel design for the construction of safe S-boxes based on TDERC sequence. Alexandria Engineering Journal, 2015, 54(1):65-69

[33] Farah T, Rhouma R, Belghith S. A novel method for designing S-box based on chaotic map and teaching-learning-based optimization. Nonlinear Dynamics, 2017, 88(2):1059-1074

[34] Liu G, Yang W, Liu W, Dai Y. Designing S-boxes based on 3-D four-wing autonomous chaotic system. Nonlinear Dynamics, 2015, 82(4):1867-1877

[35] Çavuşoğlu Ü, Zengin A, Pehlivan I, Kaçar S. A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system. Nonlinear Dynamics, 2017, 87(2):1081-1094

[36] Mamadolimov A, Isa H, Mohamad M S. Practical bijective S-box design. In 5th Asian Mathematical Conference, 2009, pp. 584-588

[37] Isa H, Jamil N, Z'aba M R. Improved S-Box construction from binomial power functions. Malaysian Journal of Mathematical Sciences, 2015, 9(S):21-35

[38] Isa H, Jamil N, Z'aba M R. Construction of cryptographically strong S-Boxes inspired by bee waggle dance. New Generation Computing, 2016, 34(3):221-238

[39] Isa H, Jamil N, Z'aba M R. Hybrid heuristic methods in constructing cryptograpgically strong S-Boxes. International Journal of Cryptology Research, 2016, 6(1):1-15

[40] Ivanov G, Nikolov N, Nikova S. Cryptographically strong S-boxes generated by modified immune algorithm. In International Conference on Cryptography and Information Security in the Balkans, 2015, pp. 31-42

[41] Ivanov G, Nikolov N, Nikova S. Reversed genetic algorithms for generation of bijective S-boxes with good cryptographic properties. Cryptography and Communications, 2016, 8(2):247-276

[42] Arrag S, Hamdoun A, Tragha A, Khamlich S E. Implementation of stronger AES by using dynamic S-box dependent of master key. Journal of Theoretical and Applied Information Technology, 2013, 53(2):196-204

[43] Lakshmi R, Mohan H S. Implementation and performance analysis of modified AES algorithm with key-dependent dynamic S-Box and key multiplication. Int. J. Math. Comput.

Appl. Res., 2015, 5(3):1-10

How to cite this article:
Roslan M F, Seman K, Halim AHA, Sayuti MNSM. Current implementation of advance encryption standard (AES) s-box. J. Fundam. Appl. Sci., 2017, 9*(4S), 518-542.*