IET Information Security

Research Article

Advanced encryption standard based on key dependent S-Box cube

ISSN 1751-8709
Received on 25th February 2018
Revised 23rd January 2019
Accepted on 12th March 2019
E-First on 29th April 2019
doi: 10.1049/iet-ifs.2018.5043
www.ietdl.org

Athmane Seghier[1] ✉, *Jianxin Li*[1], *Da Zhi Sun*[2]

[1]*Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, People's Republic of China*
[2]*School of Computer Science and Technology, Tianjin University, Tianjin, People's Republic of China*
✉ *E-mail: seghierathmane@buaa.edu.cn*

Abstract: The advanced encryption standard (AES) is the most encryption algorithm used around the world. This fact made it coveted by several cryptanalysis attacks. Some of them succeed to reveal certain weakness, due to its static parameters known by the public mass. The substitution box (S-Box) is among the most sensitive parameters of this standard, which plays a crucial role in the protection against these attacks. In this study, a new version of AES is introduced, which uses a set of AES-like S-Boxes instead of the original one. These S-Boxes, generated from different irreducible polynomials, are dynamically chosen, depending on the encryption key. This introduction is conducted by the design and the implementation of a new block cipher version of AES, based on a key dependent S-Box cube. Where a cube of six interchangeable S-Boxes key-dependently generated is constructed. This cube determines the active S-Box for a given round, through its random rotations guided by the round key. Furthermore, the improvement of the cryptographic robustness, provided by this method is demonstrated. This is accomplished through a comparison with the original AES, in terms of statistical and cryptographic randomness properties.

1 Introduction

The protection of sensitive data transiting public networks is primordial. This end-to-end protection is provided by the use of encryption cryptographic primitives. Nowadays, the block cipher advanced encryption standard (AES) is the most used cryptographic primitive. The AES [1], proposed as Rijndael was adopted by the National Institute of Standard Technology (NIST), in October 2000 as a private key cryptography standard algorithm. Shortly thereafter, it was approved as a standard by the Federal Information Processing Standards [2].

Several cryptanalysis attacks targeting the AES appeared during the last century [3]; particularly the linear [4] and differential [5] attacks focusing on the S-Box that provides confusion in this standard. This fact led to the appearance of several solutions offering an alternative to the standard S-Box, in order to strengthen the security of this cryptosystem. Since both cryptanalysis attacks cited above must be applied on a known S-Boxes. The main approaches investigated to enhance the AES security are the methods based on the introduction of the chaotic maps concept [6] and the ones based on key-dependent S-Box approaches [7]. As a result, several dynamic key-dependent S-Box solutions saw the day. Aiming to overcome the linear and differential attacks hypotheses, based on the imperative knowledge of the S-Box. However, although it was proved in [8] that the key-dependent S-Box methods are slower than key independent ones, they are estimated to be more secure.

S-Box is considered as the key stone of every block cipher, including the AES. As long as it is the main part of the modern symmetric encryption algorithms that determines the robustness against attacks, by offering a maximum non-linearity and a minimum difference propagation probability. Furthermore, even if the S-Box used by the AES is constructed by a determined irreducible polynomial and affine constant, it is quite feasible to use different ones, making this construction dynamic.

In this study, a new key-dependent S-Box method is proposed. It is based on dynamic S-Boxes forming a cube, generated with the same concept used for the standard AES S-Box construction. However, the irreducible polynomials and affine constant values used are key-dependent. In this way, a security layer is added against linear and differential attacks, by making the information on the S-Box used in each round inaccessible. While keeping a constant protection level ensured by each AES-like S-Box used.

The remainder of the paper is organised as follows: Section 2 discusses the main related work to the key-dependent methods. Section 3 provides the basic knowledge of the AES operations, by focusing on the byte substitution step. The proposed method is introduced in Section 4. Section 5 describes the statistical and randomness cryptographic experimental comparison results. Finally, conclusion and future work are given in Section 6.

2 Related work

Several approaches have contributed to the improvement of the AES algorithm, by proposing an alternative to the original S-Box. In this section, some approaches based on dynamic S-Box, constructed depending on the cipher key, are reviewed.

In [9], a method based on an initial S-Box constructed from a new irreducible polynomial is presented. Where the values of the S-Box are selected by adding to the original process a shifting value obtained from the cipher key. In [10], the authors advanced a method that applies for the standard S-Box a rotation for each round. The rotation value is calculated by Xoring all values of a round key so that the new position of S-Box values becomes the initial position plus the rotation value. In [11], a method that took an initial S-Box constructed in the same way with the original S-Box is proposed, and then the columns and the rows of the resulting S-Box are inverted. From this step, the cipher key is used to generate a dynamic S-Box through the shift columns, row columns, and shift account algorithms. In [12], the authors proposed a dynamic S-Box based on the variation of the affine transformation constant, using the most non-linear letter of the key. In [13], the authors proposed a method that permutes for each round the whole elements of the initial S-Box by the use of a swap function that takes values from a new S-Box generated dynamically by a random function using the round key as a seed. Kazlauskas *et al.* [14] proposed a method that calculates permutation indexes between a pair of S-Box elements, and apply them for the construction of a new S-Box called Sboxm, starting from an initial S-Box and the encryption key. In [15], the three-dimensional concept was introduced in both encryption key and S-Box. By performing a dynamic S-Box through a rotation obtained

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Fig. 1 *Affine transformation*

with a random value used in the key rotation. Afterwards, the resulting key value is used to process the S-Box rotation from the initial state.

3 Advanced encryption standard (AES)

In this section, the fundamental notions of AES are introduced, in order to facilitate the comprehension of the notions addressed in the next section.

AES is an iterative block cipher supporting three distinct numbers of rounds (10, 12, 14), according to the key size (128, 192, 256) bits. Each round consists of four (04) transformations as follows:

i. ByteSub: this step is ensured by the S-Box, which provides non-linearity and confusion.
ii. Shiftrow: the rotations applied in this step provide inter-column diffusion.
iii. MixColumn: this step consists of a linear combination, which provides inter-byte diffusion.
iv. AddRoundKey: in this step, the round key bytes are XORed with each byte resulting from the previous step, in order to provide confusion.

AES takes its operation modules on finite field mathematics, and the original S-Box used in the first transformation of the standard is generated by two different operations:

the multiplicative inverse over the Galois Field GF(2^8) of 256 bytes (0–255); using as a modulus, a particular irreducible polynomial: $m(x) = x^8 + x^4 + x^3 + x + 1$, represented by the hexadecimal notation $\{11B\}$.

The second operation is the affine transformation over GF(2^8) as defined in Fig. 1, where $[b'_0, \ldots, b'_7]$ is the result of the affine transformation for a given byte b through the vector product $[b_0, \ldots, b_7]$ of its multiplicative inverse, with the affine matrix, and the addition of vector [0, 1, 1, 0, 0, 0, 1, 1], corresponding to the additive constant having the hexadecimal value $\{63_H\}$ [16].

3.1 Substitution box (S-Box)

A good S-Box is the one that achieves a high quality for most of its cryptographic properties. However, it is difficult to find an S-Box that fulfils all the cryptographic criteria at the same time. This is due to the fact that some of these properties are contradictory to each other's. This is reflected in the weakening of some criteria in trying to improve others. Thereby, if the construction of the S-Box is conducted by the same method followed by AES, then the pair of the multiplicative inverse modulus and the affine constant are the main responsible in the determination of its cryptographic properties.

The S-Box criteria adopted in this study are those obtained by a multiplicative inverse modulus belonging to the set of irreducible polynomials. These criteria are characterised by a high non-linearity and algebraic degree, equal to, respectively, 112 and 7, and a low-differential uniformity equal to 4. This kind of S-Box is called AES-like S-Box.

Note that the multiplicative inverse used in AES is performed by one of the 30 existing irreducible polynomials. However, although the AES-like S-Boxes share the same non-linearity, algebraic degree, and differential uniformity criteria, it was proved that some irreducible polynomials provide better structure than the conventional one, in terms of Boolean expression (BE) distribution and avalanche effect (AE) [17, 18]. On the other hand, the affine transformation can also play an important role in the BE and the iteration cycle increasing [19], in addition to the iterative period (IP) enhancement and the strict avalanche criterion (SAC) distance reduction [20]. Thus, the original S-Box can be replaced by any S-Box constructed using one of the 30 irreducible polynomials presented in Table 1 [21]. Moreover, this construction can be conducted by replacing the original affine constant value 63_H with any value comprised between 01_H and FF$_H$.

Table 1 Irreducible polynomials list

N	Polynomial representation	Hexadecimal representation
1	$X^8 + X^4 + X^3 + X + 1$	11B$_H$
2	$X^8 + X^4 + X^3 + X^2 + 1$	11D$_H$
3	$X^8 + X^5 + X^3 + X + 1$	12B$_H$
4	$X^8 + X^5 + X^3 + X^2 + 1$	12D$_H$
5	$X^8 + X^5 + X^4 + X^3 + 1$	139$_H$
6	$X^8 + X^5 + X^4 + X^3 + X^2 + X + 1$	13F$_H$
7	$X^8 + X^6 + X^3 + X^2 + 1$	14D$_H$
8	$X^8 + X^6 + X^4 + X^3 + X^2 + X + 1$	15F$_H$
9	$X^8 + X^6 + X^5 + X + 1$	163$_H$
10	$X^8 + X^6 + X^5 + X^2 + 1$	165$_H$
11	$X^8 + X^6 + X^5 + X^3 + 1$	169$_H$
12	$X^8 + X^6 + X^5 + X^4 + 1$	171$_H$
13	$X^8 + X^6 + X^5 + X^4 + X^2 + X + 1$	177$_H$
14	$X^8 + X^6 + X^5 + X^4 + X^3 + X + 1$	17B$_H$
15	$X^8 + X^7 + X^2 + X + 1$	187$_H$
16	$X^8 + X^7 + X^3 + X + 1$	18B$_H$
17	$X^8 + X^7 + X^3 + X^2 + 1$	18D$_H$
18	$X^8 + X^7 + X^4 + X^3 + X^2 + X + 1$	19F$_H$
19	$X^8 + X^7 + X^5 + X + 1$	1A3$_H$
20	$X^8 + X^7 + X^5 + X^3 + 1$	1A9$_H$
21	$X^8 + X^7 + X^5 + X^4 + 1$	1B1$_H$
22	$X^8 + X^7 + X^5 + X^4 + X^3 + X^2 + 1$	1BD$_H$
23	$X^8 + X^7 + X^6 + X + 1$	1C3$_H$
24	$X^8 + X^7 + X^6 + X^3 + X^2 + X + 1$	1CF$_H$
25	$X^8 + X^7 + X^6 + X^4 + X^2 + X + 1$	1D7$_H$
26	$X^8 + X^7 + X^6 + X^4 + X^3 + X^2 + 1$	1DD$_H$
27	$X^8 + X^7 + X^6 + X^5 + X^2 + X + 1$	1E7$_H$
28	$X^8 + X^7 + X^6 + X^5 + X^4 + X + 1$	1F3$_H$
29	$X^8 + X^7 + X^6 + X^5 + X^4 + X^2 + 1$	1F5$_H$
30	$X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + 1$	1F9$_H$

4 Proposed method

AES based on key-dependent S-Boxes cube is a new version that satisfies the original AES specifications concerning the block and the key size, as well as the four rounds operations. The main difference resides on the number of available S-Boxes during the sub-bytes step of the algorithm. The S-Boxes in question are chosen to construct a cube, by placing each S-Box in one of the six cube faces, and they are selected through random rotations depending on the values of the encryption key and its expansion.

The advantage of this method is that the displacements through the cube allowing the choice of the active S-Box, for each round of the encryption case, are easily obtained during the decryption process. Since in both cases (encryption/decryption), the selecting path is obtained directly from the key. The S-Boxes composing the substitution cube are constructed and selected as shown in Fig. 2.

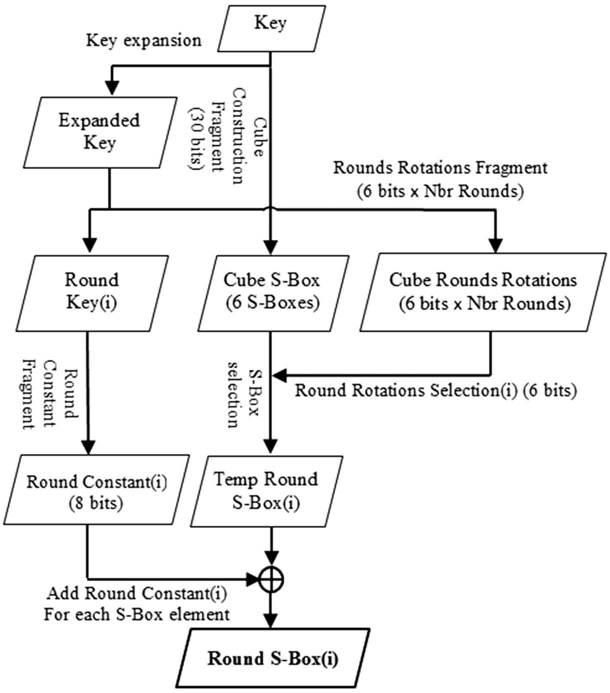

Key
Key expansion
Expanded Key
Cube Construction Fragment (30 bits)
Rounds Rotations Fragment (6 bits x Nbr Rounds)
Round Key(i)
Cube S-Box (6 S-Boxes)
Cube Rounds Rotations (6 bits x Nbr Rounds)
Round Constant Fragment
S-Box selection
Round Rotations Selection(i) (6 bits)
Round Constant(i) (8 bits)
Temp Round S-Box(i)
Add Round Constant(i) For each S-Box element
Round S-Box(i)

Fig. 2 *Round S-Box selection scheme*

```
Data: $Encrypt\text{-}key$, $Irreducible\text{-}polynom\text{-}list$ [30].
Result: six (06) S-Boxes forming the Cube.
begin
    /* (6 distinct values) modulo 30      */
    $Cube\text{-}S\text{-}Boxes\text{-}Indexes \leftarrow Encrypt\_key\_fragment$
    for $i \in Cube\text{-}S\text{-}Boxes\text{-}Indexes$ do
        /* with original affine transform  */
        $ConstructS\text{-}Box\,[i]$
```

Fig. 3 *Algorithm 1: dynamic cube-S-Box construction*

Before obtaining the active round S-Box, three steps are followed.

4.1 Substitution cube construction

The first step as presented by Algorithm 1 (see Fig. 3) consists of the generation by the same method used for the conventional AES S-Box, presented in the previous section, of the six S-Boxes constructing the substitution cube. Such as the multiplicative inverse of these S-Boxes is calculated using distinct irreducible polynomials key dependently selected from the set of 30 irreducible polynomials cited in Table 1. Thus, the selection is guided by six distinct indexes to the irreducible polynomials list taken from the encryption key. So that, every six bits of a 36 bits key-fragment determine the index of an irreducible polynomial selected to construct one S-Box from the substitution cube. Nevertheless, for the initial cube S-Boxes construction, the affine constant that takes a certain hexadecimal value comprise between 01_H and FF_H, is kept the same for the whole cube S-Boxes generation [22]. This step aims to increase the number of potential S-Boxes used from a unique S-Box to six S-Boxes among 30 available S-Boxes $\binom{30}{6}$ with distinct irreducible polynomials.

As soon as the cube is in place, the key-scheduling could be possible, since an S-Box is needed during this process. For that, the first S-Box of the cube is chosen to achieve this task.

4.2 Round S-Box selection

In this step, the constructed cube is used to select for each round of the algorithm an S-Box among the six. This selection will be guided by a fragment of 6 bits, extracted from the corresponding round key. This fragment is used to determine the current S-Box, by computing from the previous position of the active S-Box, the direction and the number of moving steps for the horizontal and vertical displacements over the cube faces representing the six S-Boxes.

```
Data: $Cube\text{-}S\text{-}Boxes$, $Expand\text{-}Key\text{-}frag$.
Result: $Rounds\text{-}S\text{-}Boxes\text{-}Index$.
begin
    if $Expand\text{-}Key\text{-}frag\,[0] = 1$ then
        $rotate\_right$
        if $Expand\text{-}Key\text{-}frag\,[1..2] = i$ and $i$ in$[00..11]$ then
            $i\ right\text{-}rotations$
    else if $Expand\text{-}Key\text{-}frag\,[0] = 0$ then
        $rotate\_left$
        if $Expand\text{-}Key\text{-}frag\,[1..2] = i$ and $i$ in$[00..11]$ then
            $i\ left\text{-}rotations$
    if $Expand\text{-}Key\text{-}frag\,[3] = 1$ then
        $rotate\_up$
        if $Expand\text{-}Key\text{-}frag\,[4..5] = i$ and $i$ in$[00..11]$ then
            $i\ up\text{-}rotations$
    else if $Expand\text{-}Key\text{-}frag\,[3] = 0$ then
        $rotate\_down$
        if $Expand\text{-}Key\text{-}frag\,[4..5] = i$ and $i$ in$[00..11]$ then
            $i\ down\text{-}rotations$
```

Fig. 4 *Algorithm 2: cube rotations*

Each displacement on the substitution cube from a given position is defined by three different notions which are: axis, direction, and steps, such as

- *Axis:* defines the vertical or horizontal axis of the displacement.
- *Direction:* up and down for the vertical axis, right and left for the horizontal axis.
- *Steps:* represents the number of displacements to perform in a given axis and following a given direction. The tolerated steps are zero, one, two or three steps.

Concretely, the fragment of 6 bits is divided into two sub-fragments of three bits each, in such a way that the first sub-fragment is used to define the displacement in the horizontal axis, as follows: the first bit of this fragment to calibrate the horizontal displacement direction (0: displacement to the right and 1: displacement to the left direction). The two succeeding bits to calculate the number of horizontal displacement steps (00, 01, 10 or 11 for, respectively, 0, 1, 2 or 3 steps). While the second sub-fragment is used to define the displacement in the vertical axis, as follows: the first bit to select the vertical displacement direction (0: rotation to down direction and 1: rotation to up direction) and the two succeeding bits to define the number of vertical displacement steps (00, 01, 10 or 11 for, respectively, 0, 1, 2 or 3 steps). This step is operated through the pseudo code represented by Algorithm 2 (see Fig. 4). For example if the previous active S-Box is S − Box1 and the extracted fragment guiding the displacement used to select the current active S-Box is: 1 01 0 10, so the displacements will be: displacement or rotation to the left direction (1) with one step (01) to reach S − Box4, then displacement or rotation to down direction (0) with two steps (10) to select S − Box2. knowing that from the S − Box1, S − Box2 is reached through (1) right rotation or (3) left rotations, S − Box3 is reached through (2) right or (2) left rotations and S − Box4 is reached through (3) right or (1) left rotation following the horizontal axis. While S − Box5 is reached through (1) up rotation or (3) down rotations, S − Box3 is reached through (2) up or (2) down rotations and S − Box6 is reached through (3) up or (1) down rotation following the vertical axis.

4.3 Dynamic round S-Box calculation

The effective round S-Box is obtained by applying for each value $\mathrm{S} - \mathrm{Box}[i]_{i = 0, \ldots, 255}$ of the 256 output values of the selected round S-Box a XOR operation with a value (8 bits) called 'RoundConstant';

which is extracted from the round-key, as presented in the pseudo code in Algorithm 3 (see Fig. 5) by

$$\mathrm{S-Box}[i]_{i = 0, \dots, 255} \oplus \mathrm{RoundConstant} \tag{1}$$

. In other words, this step consists of transforming an initial S-Box previously selected from the substitution cube (constructed from a distinct irreducible polynomial and the same affine constant with the other S-Boxes of the cube) to a new S-Box active round S-Box by adding a round constant to the 256 values of the initial S-Box, aiming to use a distinct S-Box for each round, and therefore increases the number of possible active S-Boxes from the initial six S-Boxes forming the cube to a number of S-Boxes equal to (6×256) from one round and $(6 \times 256)^R$ for R rounds. In this way, even though two given rounds randomly choose the same initial S-Box from the substitution Cube, their active round S-Box should be different unless the fragments used as *RoundConstant* for these two rounds have the same value.

A summary comparison between the original method AES and the proposed method is illustrated in Table 2 and some details of which are given in the Appendix. This comparison is established in terms of functionality, the possible S-Box combinations for a given block cipher round and the cryptographic properties of the offered S-Boxes.

In practice, the additional processing performed by this method compared to the original one consists of the construction of the substitution cube, which includes five additional S-Boxes. However, this extra processing is negligible from the performance perspective, as long as it is performed only once, during the initialisation step of the encryption/decryption process. Moreover, for each round, further treatment is required, for the selection of the active round S-Box through the cube rotation, as well as the active S-Box modification, through the Xor operation of all its elements with a secret value.

```
Data: $Round\text{-}key, Cube\_S\text{-}Box$.
Result: $Round\_S\text{-}Box$.
begin
    $Rounds\_S\text{-}Boxes\_Index\,[\,] \leftarrow Cube\_rot(Cube\_S\text{-}Box, Expand\text{-}Key\text{-}frag)$

    for $i \in Rounds - Number$ do
        $TmpRoundSBox\,[i] \leftarrow Cube\_S\text{-}Box\,[Rounds\_SBoxes\_Index\,[i]]$

        $RoundConst\,[i] \leftarrow get\_Round\_Key\_value$
        for $j\ in\ 256$ do
            $Round\_S\text{-}Box\,[i]\,[j] \leftarrow TmpRoundSBox\,[i]\,[j] \oplus RoundConst\,[i]$
```

Fig. 5 *Algorithm 3: dynamic round S-Box*

It is observed through the study and the analysis of the related works cited in Section 2, that the used S-Boxes during the whole process are based on a previously determined irreducible polynomial. Unlike our method that provides the possibility to use different AES-like-S-Boxes built with different irreducible polynomials and affine constant, randomly chosen. Thereby, apart from the protection afforded against linear and differential attacks, it is estimated that the proposed method gives also a head start on the recent version of the integral cryptanalysis [23], which is applied on AES with an unknown S-Box. This further protection is due to the significant number of secret AES like S-Boxes that can be used during each round.

5 Statistical and cryptographic randomness tests

In this section, the statistical and cryptographic randomness properties of the standard AES and the proposed method are studied. The followed method in this section is inspired by the one introduced in [24], used for the cryptographic randomness testing of block ciphers and hash functions.

5.1 Statistical randomness tests (NIST statistical tests)

In this subsection, the statistical tests recommended by the NIST are applied to verify the randomness of the proposed method output. For that, the tests are conducted with the data and size samples which tend to reflect the same test benchmarks used in the AES election. The setting in question corresponds to the random keys (128 bits) and plaintexts generated by the Blum-Blum-Shub (BBS) pseudo-random generator, as well as the high and low density plaintext [25, 26].

5.1.1 Random 128-bit keys/plaintexts: This test is based on the test of randomness of 128 separated ciphertexts obtained by the encryption of 128 plaintexts which are composed by 8128 block of 128 bits length, generated by the BBS generator, which is considered as a cryptographically secure pseudo-random generator, using the different methods tested, by changing for each plaintext a different 128 bits key also generated with the BBS generator [26].

5.1.2 Low-density plaintext: In this case, the behaviour of the encryption algorithm with respect to an input with a low density

Table 2 Original and proposed methods comparison

		Proposed method	Standard AES
key-dependence		key-dependent cube S-Boxes	key independent S-Box
available S-Boxes		30 × 256 = 7650 S-Boxes	one S-Box
potential available combinations of used S-Boxes for all rounds in one block	10 rounds	$\binom{30}{6} \times (6 \times 256)^{10} = 4.34 \times 10^{37}$	one combination
	12 rounds	$\binom{30}{6} \times (6 \times 256)^{12} = 1.02 \times 10^{44}$	one combination
	14 rounds	$\binom{30}{6} \times (6 \times 256)^{14} = 2.41 \times 10^{50}$	one combination
S-Box properties	*fixed* or *inverse-fixed points* existence	yes	no
	nonlinearity	112	112
	$\Delta - uniformity$	4	4
	algebraic–degree	7	7
	others: AE, BE, IP, SAC etc.	uncontrolled quality	medium quality
additional calculation		- 5 additional S-Boxes construction - cube rotations - XOR computation for each S-Box values of each round.	nothing

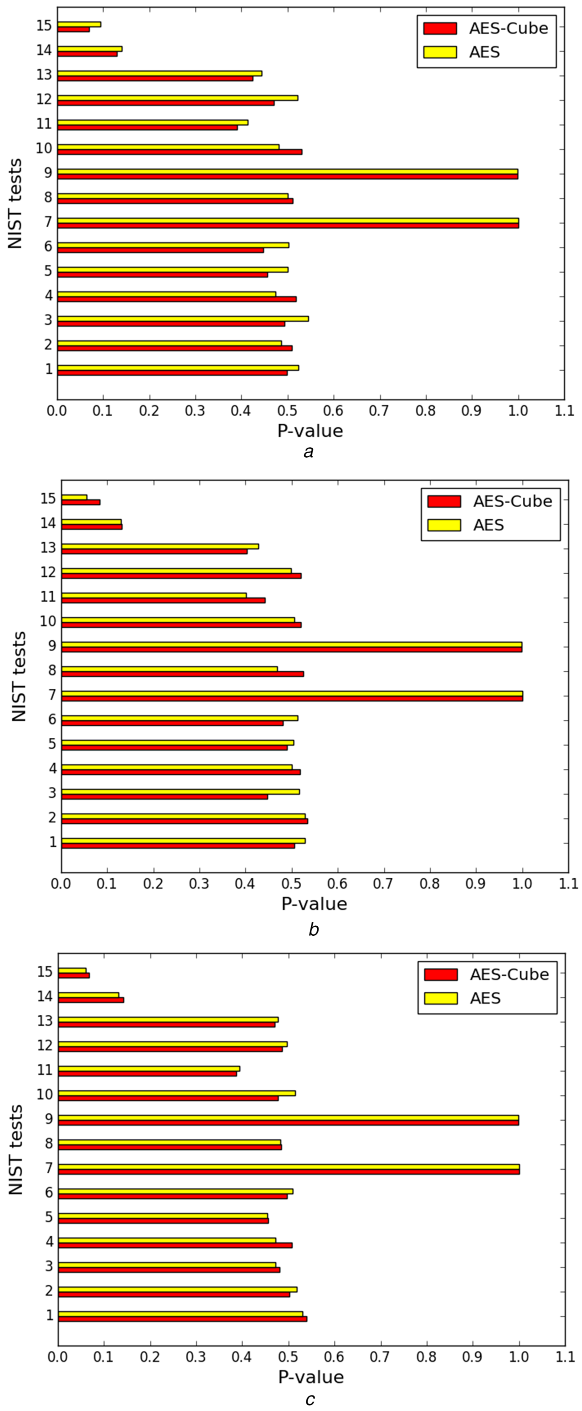


Fig. 6 *NIST randomness tests*
(a) Random 128-bit keys/plaintexts test, ***(b)*** Low-density plaintext test, ***(c)*** High-density plaintext test

(low existence of ones) is studied. This is conducted by starting with the encryption of 128 texts, composed of 8257 blocks obtained as follows:

- The first block is composed only by zeros.
- The blocks between 2 and 219 are obtained by inserting a one on all possible positions of the block and filling the rest with zeros.
- The blocks between 130 and 8257 are obtained by inserting two ones on all possible positions of the block and filling the rest with zeros.

The 128 texts are encrypted with a different random 128 bits key for each text [26].

Table 3 Correlation coefficient results comparison

CCT results characteristic	Value
number of samples	100,000
number of times original AES give better CCT	49,707
number of times proposed method give better CCT	49,220
number of times both methods give same CCT	1073
average CCT for the original AES	0.070844
average CCT for the proposed method	0.070609

5.1.3 High-density plaintext: In this case, the behaviour of the encryption algorithm with respect to an input with a high density (high existence of ones) is studied. This is conducted by starting with the encryption of 128 texts, composed of 8257 blocks obtained as follows:

- The first block is only ones.
- The blocks between 2 and 219 are obtained by inserting a zero on all possible positions of the block and filling the rest with ones.
- The blocks between 130 and 8257 are obtained by inserting two zeros on all possible positions of the block and filling the rest with ones.

The 128 texts are encrypted with a different random 128 bits key for each text [26].

The three randomness experiments show that both methods pass the 15 tests proposed by NIST suite, and give almost the same results (see Fig. 6). Despite the randomness character validation of the proposed method, through the NIST suite tests is necessary, however, this work does not replace the cryptanalysis attacks results. For this reason, it is important to reinforce this study by evaluating the cryptographic property of the proposed method results.

5.2 Cryptographic randomness tests

In addition to the randomness tests of this block cipher output, presented in the previous subsection, certain cryptographic properties of this method, such as the diffusion and confusion should be satisfied. Since, these two properties are the most important, when the block cipher is considered. This is achieved through the correlation coefficient test, as well as two tests extracted from the cryptographic randomness tests, introduced in [24]: the SAC, which reflects the quality of the diffusion property. Also, the linear span test (LST), which reflects the quality of the confusion property. The selected tests give information about the linear dependence between the input and the output of a considered block cipher.

5.2.1 Correlation coefficient test (CCT): CCT is part of the essential tests set, for the encryption algorithms audit. Since it reflects the dependency rate; delimited by −1 and 1 (with a good CCT value close to the zero), between every two bits, located at the same position in the plain and the ciphertext of the cryptographic function studied.

Table 3 and Fig. 7 show the experimental results of 100,000 samples composed of a random plaintext and key and the resulting ciphertext with 128 bits, tested by the Pearson method of the CCT.

5.2.2 Strict avalanche criterion (SAC): This test consists of the combination of two distinct concepts. The completeness concept: meaning that any output from a cryptographic transformation, depends on all its input bits. Also, the AE concept, wherein the change rate in the output is expected to reach one half by altering one input bit.

The test is to pass the SAC, only if, regardless of the input bit complemented, then the output bits should change with a probability around one half. Note that the original version of this test was applied to the S-Boxes [27], and later, it was adapted to the block cipher algorithms [28].

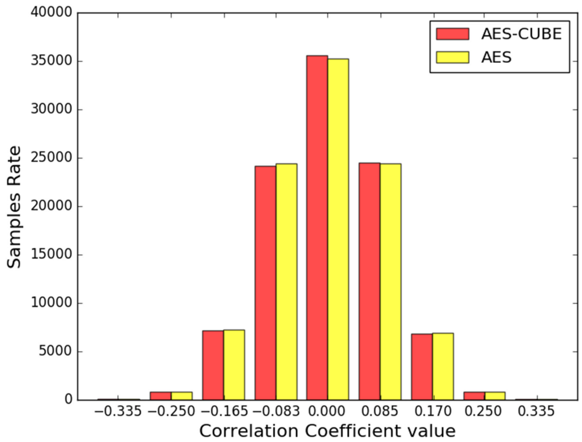
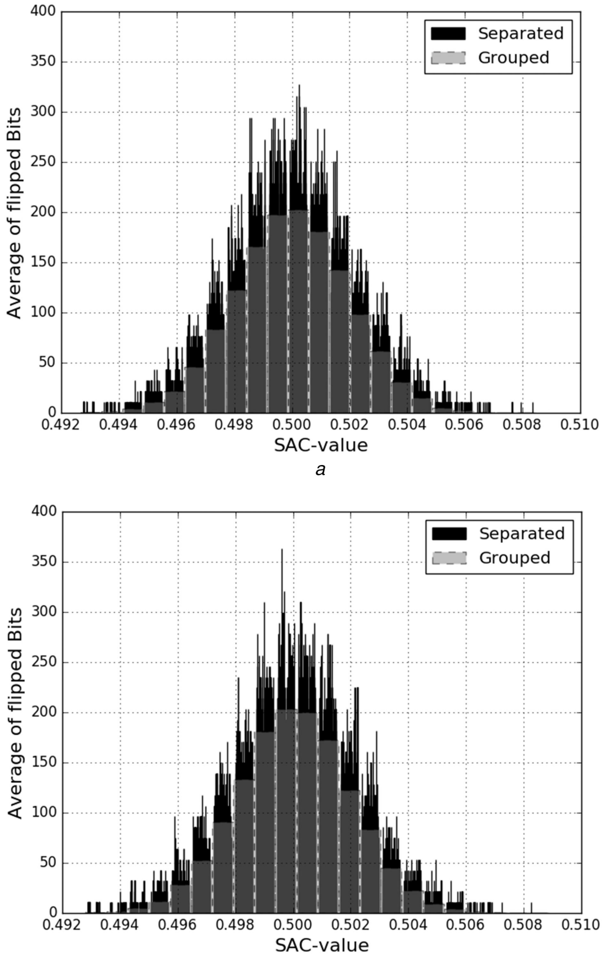


Fig. 7 *Correlation coefficient test*



a

b

Fig. 8 *Strict avalanche criterion (SAC)*
(a) AES-cube, ***(b)*** AES

The test is achieved by constructing a matrix called SAC matrix as follows:

A set of random inputs P is encrypted to obtain a set of cipher outputs C. For each input $P_K \in \{P_1, P_2, \ldots, P_N\}$, every bit i is flipped separately, and the result is encrypted to obtain a new cipher output C'_{Ki}, which is *XORed* with the original cipher output C_K. Afterwards, the result is added to the whole jth elements in the ith row of the SAC matrix, where $i, j \in [1, 128]$. This process is repeated for each element of the inputs set. The set length is recommended to be as large as possible (2^{20}), as mentioned in [24]. However, in this study, this test was accomplished with a value equal to 2^{16}. In the end, the results in the matrix are divided by the set length. The results given in Fig. 8 represent an inputs set of 2^{16} (128-bits) plaintexts, the average of the SAC matrix elements having a given SAC value, with a separated and grouped matrix lines views. This result shows that both methods satisfy the SAC, with a slight advantage to the proposed one.

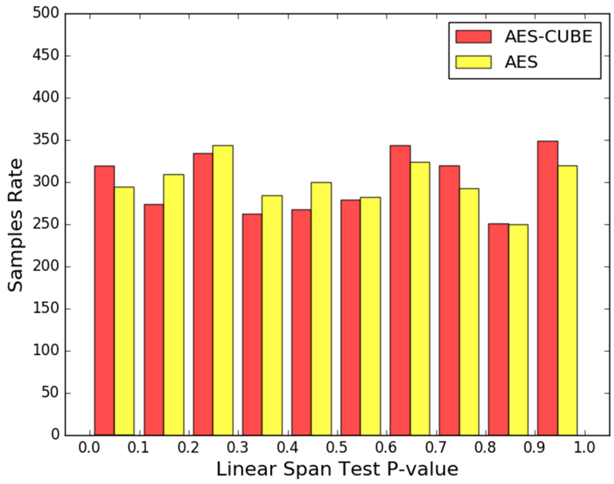


Fig. 9 *Linear span test*

Table 4 Linear span results comparison

LST results characteristic	Value
number of samples	3000
number of times original AES give better LST	1471
number of times proposed method give better LST	1510
number of times both methods give same LST	19
average LST for the original AES	0.49489
average LST for the proposed method	0.50630

5.2.3 Linear span test (LST): LST is based on the well-known binary matrix rank test [25]. It belongs to NIST suite tests, previously introduced, which evaluate the randomness characteristic of any input sequence, given in this case by the block cipher encryption result. This test has great importance for the linear independence validation between the input and the output of the proposed method, as long as this method is mainly based on the continual change of the active S-Box values. Knowing that the S-Box represents the main part that ensures the non-linearity of the algorithm, thereby, the non-linearity quality of the used S-Boxes should be measured through this test.

Similarly to the LST proposed for stream ciphers [29], and the one proposed for the block cipher [24], N sets of all linear combinations, generated from m linearly independent plaintexts of 2^m bits are processed, by applying the binary matrix rank test for the N which is $(2^m \times 2^m)$ bits matrices formed by the encryption result of the N linear combinations. This process is repeated for a considerable amount of samples. In this experimentation, 3000 samples are analysed with AES and the proposed method, using a different encryption key for each sample. The results shown in Fig. 9 and those presented in Table 4 of LST results comparison between the original AES and the proposed method, attests that our method has a slight advantage over the original AES.

6 Conclusion and future work

In this study, a new AES version is proposed, based on key-dependent S-Box cube, in which six S-Boxes are constructed with distinct irreducible polynomials key dependently selected. These S-Boxes are used to select for each round, through the movement of the Cube guided by a fragment from the round key, an initial S-Box processed by a round constant to obtain an active round S-Box. This new version increases considerably the number of available S-Boxes and their potential combinations over the block encryption rounds, which thwarts the linear and differential cryptanalysis attacks assumptions based on known S-Boxes. Furthermore, this enhancement does not affect the quality of the used S-Boxes, by constructing only the AES-like S-Boxes, through different irreducible polynomials.

From the derived results given by the randomness and the cryptographic properties analysis, it can be noted that the proposed method gives a good result compared with the original AES.

In future work, the effect of decreasing the rounds iterations number of this method will be studied. In order to improve the processing time, by making up the additional treatment performed by our method. In addition, the possibility of using only S-Boxes with good cryptographic characteristics in order to enhance the encryption quality, somehow reflected by the S-Box used.

7 References

[1] Daemen, J., Rijmen, V.: '*The design of rijndael. Information security and cryptography*' (Springer-Verlag New York, Inc., NJ, USA, 2002)

[2] 197, F.I.P.S.P.: 'Announcing the ADVANCED ENCRYPTION STANDARD (AES)', Federal Information Processing Standards Publication, 2001

[3] Courtois, N.T.: 'Is AES a secure cipher?'. Available at http://www.cryptosystem.net/aes/, accessed 04 February 2018

[4] Matsui, M.: 'Linear cryptanalysis method for DES cipher'. Advances in Cryptology – EUROCRYPT'93, Lofthus, Norway, May 1993 (LNCS, **765**), pp. 386–397

[5] Biham, E., Shamir, A.: 'Differential cryptanalysis of DES-like cryptosystems', *J. Cryptol.*, 1991, **4**, (1), pp. 3–72

[6] Rohiem, A.E., Elagooz, S., Dahshan, H.: 'A novel approach for designing S-Box of advanced encryption standard algorithm (AES) using chaotic map'. Twenty Second National Radio Science Conf., Cairo, Egypt, March 2005, pp. 455–464

[7] Fahmy, A., Shaarawy, M., El-Hadad, K., *et al.*: 'A proposal for a key-dependent AES'. 3rd Int. Conf.: Sciences of Electronic, Technologies of Information and Telecommunications: SETIT, Tunisia, Susa, March 2005

[8] Schneier, B.: '*Foundations – applied cryptography*' (1996, 2nd edn., 20th anniversary edition), pp. 1–18

[9] Mohammad, F.Y., Rohiem, A.E., Elbayoumy, A.D.: 'A novel S-Box of AES algorithm using variable mapping technique'. Proc. 13th Int. Conf. on Aerospace Sciences and Aviation Technology, ASAT- 13, Cairo, Egypt, May 2009, pp. 1–10

[10] Juremi, J., Mahmod, R., Sulaiman, S., *et al.*: 'Enhancing advanced encryption standard S-Box generation based on round key', *Int. J. Cyber-Secur. Digit. Forensics*, 2012, **1**, (3), pp. 183–188

[11] Hosseinkhani, R., Javadi, H.H.S.: 'Using cipher key to generate dynamic S-Box in AES cipher system', *Int. J. Comput. Sci. Secur.*, 2012, **6**, (1), pp. 19–28

[12] Partheeban, P., Nityanandam, P.N.: 'Generation of dynamic S-BOX using irreduceable polynomial and the secret key used', *Int. J. Res. Eng. Sci.*, 2013, **1**, (5), pp. 24–27

[13] Alabaichi, A., Salih, A.I.: 'Enhance security of advance encryption standard algorithm based on key-dependent S-Box'. 2015 5th Int. Conf. on Digital Information Processing and Communications, ICDIPC 2015, Sierre, Switzerland, October 2015, pp. 44–53

[14] Kazlauskas, K., Vaicekauskas, G., Smaliukas, R.: 'An algorithm for key-dependent S-Box generation in block cipher system', *Informatica*, 2015, **26**, (1), pp. 51–65

[15] Rahaman, Z., Corraya, A.D., Sumi, M.A., *et al.*: 'A novel structure of advance encryption standard with 3-dimensional dynamic S-Box and key generation matrix', *Int. J. Adv. Comput. Sci. Appl.*, 2017, **8**, (2), pp. 314–320

[16] Daemen, J., Rijmen, V.: '*The design of Rijndael – AES – the advanced encryption standard*' (Springer Science & Business Media, USA, 2013)

[17] Wang, D., Sun, S.L.: 'Replacement and structure of S-Boxes in Rijndael'. Int. Conf. on Computer Science and Software Engineering (CSSE), Hubei, China, December 2008, vol. 3, pp. 782–784

[18] Xian, Z.H., Sun, S.L.: 'Study on test for structure of S-Boxes in Rijndael'. 2nd Int. Workshop on Education Technology and Computer Science, ETCS, Wuhan, China, March 2010, vol. 3, pp. 84–86

[19] Li, X., Chen, J., Liu, W., *et al.*: 'An improved AES encryption algorithm'. IET Int. Communication Conf. on Wireless Mobile & Computing, (CCWMC), Shanghai, China, December 2009, pp. 694–698

[20] Cui, J., Huang, L., Zhong, H., *et al.*: 'An improved AES S-Box and its performance analysis', *Int. J. Innov. Comput. Inf. Control*, 2011, **7**, (5), pp. 2291–2302

[21] Church, R.: 'Tables of irreducible polynomials for the first four prime moduli', *Ann. Math.*, 1935, **36**, (1), pp. 198–209

[22] Das, S., Zaman, J.K.M.S.U., Ghosh, R.: 'Generation of AES S-Boxes with various modulus and additive constant polynomials and testing their randomization', *Procedia Technol.*, 2013, **10**, pp. 957–962

[23] Tiessen, T., Knudsen, L.R., Kölbl, S., *et al.*: 'Security of the AES with a secret S-Box'. Int. Workshop on Fast Software Encryption, FSE, Istanbul, Turkey, March 2015, pp. 175–189

[24] Doganaksoy, A., Ege, B., Koçak, O., *et al.*: 'Cryptographic randomness testing of block ciphers and hash functions', *IACR Cryptology ePrint Archive*, 2010, **2010**, p. 564

[25] Rukhin, A., Soto, J., Nechvatal, J., *et al.*: '*A statistical test suite for random and pseudorandom number generators for cryptographic applications*' (Booz-Allen and Hamilton Inc., Mclean, VA, 2001)

[26] Soto, J., Bassham, L.: '*Randomness testing of the advanced encryption standard finalist candidates*' (Booz-Allen and Hamilton Inc., Mclean, VA, 2000)

[27] Webster, A.F., Tavares, S.E.: 'On the design of S-Boxes'. Conf. on the Theory and Application of Cryptographic Techniques, Berlin, Heidelberg, 1986, pp. 523–534

[28] Turan, M.S., Çalik, Ç., Saran, N.B., *et al.*: 'New distinguishers based on random mappings against stream ciphers'. 5th Int. Conf. on Sequences and Their Applications – SETA 2008, Berlin, Heidelberg, September 2008, (LNC, **5203**), pp. 30–41

[29] Turan, M.S.: 'On statistical analysis of synchronous stream ciphers'. PhD thesis, Middle East Technical University, Ankara, Turkey, 2008

8 Appendix

The formulas used to compute the potentially available combinations of the used S-Boxes per rounds in one block for the proposed method are calculated as follows (Table 2):

- $\binom{30}{6}$ represents the possible cases of choosing six distinct irreducible polynomials from the 30 available irreducible polynomials to construct the six S-Boxes of the cube.
- (6×256) represents the number of active S-Boxes that could be constructed with Algorithm 3 (Fig. 5), by applying 256 possible transformations (add round constant with XOR operation for each S-Box element) with one of the six (6) S-Boxes forming the cube (for one round).
- $(6 \times 256)^R$ with $R = \{10, 12, 14\}$ represents the possible combinations for R rounds between the possible active S-Boxes that could be obtained for each round.