ISSN 2303-4521
Original Research

Periodicals of Engineering and Natural Sciences
Vol. 9, No. 4, October 2021, pp.674-684

A novel of substitution-box design using PLL algorithms in magic cube

Rana M. Zaki[1], Hala Bahjat Abdul wahab[2]
Computer Science Department, University of Technology- Iraq, Baghdad, Iraq

ABSTRACT

In most modern symmetric ciphers, substitution boxes are non-linear core components that provide robust security and confusion. The construction of active S-boxes has been hot object among safely experts. The goal is at build a cryptographically active S-box, number of researchers have built an S-Box using RNA, DNA, chaotic systems, etc. In this article, we used new approach to generate multiple S-Boxes based on the Permutation of Last Layer algorithms (PLL) to replacement the place (permutation) of the pieces generated from the Magic Cube and instead of using static S-Boxes, we can generate dynamic S-Boxes to use various S-Boxes in every round to add to confusion. When comparing the proposed algorithm with other searches and tests on it. We found it has achieved a high percentage of success comparable to the S-Box from chaotic and etc. The proposed design passes S-Box test criteria effectively, these tests**:** invertibility, completeness, avalanche, strict avalanche, and balanced are among S-Box test criteria. The results of the analysis show that the novel S-Box overrun each of these statistical tests, has a good avalanche impact, and thus can protect against a wide range of attacks. Because S-Box and its reverse take but smaller milliseconds to construct it ability be applied in a variety of lightweight applications

Keywords: Security, Block cipher, Substitution Box, Magic Cube, PLL Algorithms

Corresponding Author:

Rana M. Zaki
Computer Science Department
University of Technology- Iraq, Baghdad, Iraq
Rana.M.Zaki@uotechnology.edu.iq

1. Introduction

Individuals and organizations can use cryptography to protect their data. Various asymmetric and symmetric ciphers reign been intended for this purpose. Symmetric ciphers are simpler and more efficient than asymmetric ciphers, and they use fewer computational resources. Stream and block ciphers are the two major types of symmetric ciphers [1]. When compared to stream ciphers, block ciphers are easier to implement, extra public, while cryptographically stronger [2]. The block cipher is regarded accordingly one of the most very applied supply into data security [3]. All of the very recent and widely used symmetric ciphers, such as AES, DES, Blowfish, RC2, RC5, and IDEA, are block ciphers. Substitution and permutation procedures or the Feistel structure are used by the majority of block ciphers. A substitution operation uses a substitution box (S-Box) to swap one set of bits for a different set of bits (S-Box). The position of bits or bytes in the specified input block is changed by the process of flipping. The S-Box is one of the most basic techniques for creating candid confusion. The creation is complex association that should be confirmed between the plaintext and the cipher text is referred to as confusion [4].Many researchers have looked into new and innovative ways to design S-Boxes. AES is a popular block cipher that employs S-Boxes in the encryption and decryption processes. Sah. et al.[5] proposed an improvement to AES's security. It encrypts different plaintext blocks using multiple sub-keys. However, the new cipher is more complex and slower than AES [6-9]. There is a lot of research that indicates the development S-Boxes to AES's security such that [10-13]. DNA computing is another popular area of cryptography that has been suggested as a possible way to solve for resistant cipher styling. Kad. et al. [14] as well as Al-Wat.et al. [15] was using DNA processing to propose effective S-Boxes, examined the validity of the proposed ciphers using different factors, as well as proved that the ciphers passed a test. Several other researchers proposed and created cryptographic techniques using DNA computing. Like [16-18].Ciphers that use S-Boxes rely heavily on their security. M. K and Gn. [19] In 2016, he suggested an S-Box focused on pseudo-random-number-generators and able to share. It proved to be more efficient when the S-box dimensions are used and can be integrated into famous cryptographic techniques. Nas. et al [20] presented S-p-Boxes that are highly nonlinear in 2019; the proposed includes a great number of highly nonlinear S-P-Boxes. Because chaotic systems have the ownership of randomness, chaot cryptographic is one of the more motivating areas in the field of input scurry in the new time [21]. Garg et al. [22] investigated various techniques for designing S-Box as well as indicated that S-Boxes built to use a chaotic method have strong cryptographic properties. When compared to the original chaotic logistic map, Alz. et al [23] introduced a new 1D discrete-chaotic map with both a big chaotic scope and better chaos actions. They constructed efficient substitution-boxes using this improved chaotic map and hill climbing search technique (S-boxes). The test results offer that the created S-box has perfect cryptography force and is set up to be best than other S-boxes ready. Mer. et al [24] S-box generation algorithms established on the series created by the 1D chaotic logistic plan neutrallision . Several other researchers proposed and created block ciphers using DNA mathematics, like [7, 23-27].

In this article, used novel algorithm to produce the S-box established the Permutation of Last Layer algorithms (PLL) generated from the Magic Cube. The first PLL algorithm is responsible for generating the decimal symbol, whilst the second algorithm is taking charge of to generate the hexa token processing this hexa symbol to create the S-Box. When compared to other proposals, the propos S-Box is prepared in that a path that it takes little time to build while also having good statistical tests.

The remainder of this article is as follows: part 2 present Magic Cube and part 3 present the propose way to build S-Box established on PLL algorithms. In part 4 present the test of this suggestion. The conclusion of this article is offered in part 5.

2. Magic cube (Rubik's cube)

Ern Rubik, a Hungarian sculptor and architecture professor, invented the Rubik's cube in 1974. The magic cube has three layers and six faces with different colors, each with nine cells with same color.

$$B_{\text{size}} = (3 * 3)_{\text{Paces}} \times 6_{\text{faces}} = 54 \qquad (1)$$

The magic cube also has a number of rotations (NR) is a possible rotation type. Popularly, in 3 × 3 × 3 cubes ,there are eighteen different rotations are potential [28].

$$NR = (3)_{\text{Layers}} \times 6_{\text{faces}} = 18 \qquad (2)$$

Permutations of the authentic 3 × 3 × 3 Rubik's Cube is like being $8! \times 3^{8} \times (12! \times 2) \times 2^{11}$

This is nearly 43 quintillion. Because of the big number of possibility, solve the Rubik's cube becomes added difficult. As a result, it was intended to be using this difficulty in the suggested encryption system, that could allow obtaining the original message more difficult for hackers.

Rubik cube can be rotated clockwise or anti-clockwise rotation, it has different layers like upper (UP), upper-inverse (UP'), down (D), down inverse (D'), front (F), front-inverse (F'), back (B), back-inverse (B'), left (L), left-inverse(L'), right (R), right inverse (R') horizontal (H), horizontal-inverse (H'), middle (M), middle-inverse (M'), vertical (V), vertical-inverse (V'), show in figure (1) .

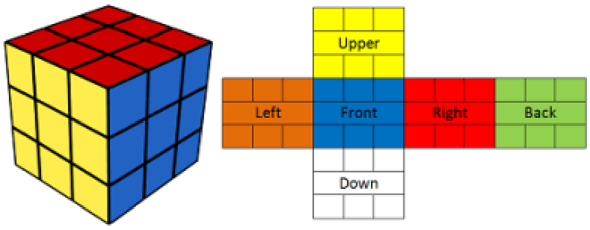


Figure 1. The Rubik's cube and its unfolding structure based on face values [29]

Rotation corner (RC): The corner of rotation of every layer is defined by the rotation corner, which can be 90°, 180°, 270°, or 360°. Because the rotation corner 360° residue constant as the corner begins to rotate, the cubes layer can be rotated at three various corners: 90°, 180°, and 270°. The rotate corner of 270° remains the same as the anti-clockwise rotation angle of 90°[30].

2.1 PLL algorithms (permutation of last layer)

It is an advanced technology of the Magic Cube which was Jessica Fridrich's progress entails memorizing a large number of algorithms, but there is a logical connection between them. The Petrus system and the Fridrich method (or full CFOP), which are used by the vast majority of speed cubes these days, must be mentioned when discussing advanced Rubik's Cube solving techniques. The advanced technique used by Jessica F. divide the puzzle into layers. and that you must help in solving the layer after layer in cube, Using algorithms at every step, without destroying the pieces that are actually see figure(2)[31-33].

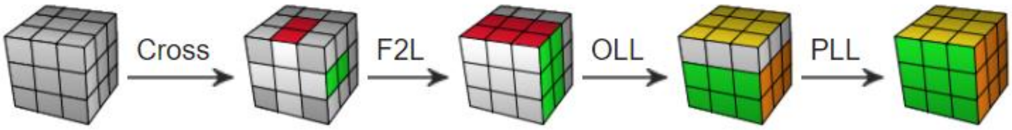


Figure 2. Cross, F2L, OLL and PLL (CFOP) Structure [33]

3. Proposed substitution box

This article describes a new method for building S-Boxes that uses PLL algorithms to modification the positioning (permutation) on the pieces in the last layer without rotary them. After that, the cube would be solved. This step is split into two sections. As well:

– Permutation of edges: It must permute the edges, which means changing their position but not orientation. It is necessary to consider the number of well-positioned edges.

– Permutation of corners: Once the edges have been properly placed, only the corner remains to be solved.

To build S-Box 16 × 16 uses PLL algorithms represented by 3x4 as show in figure (3)

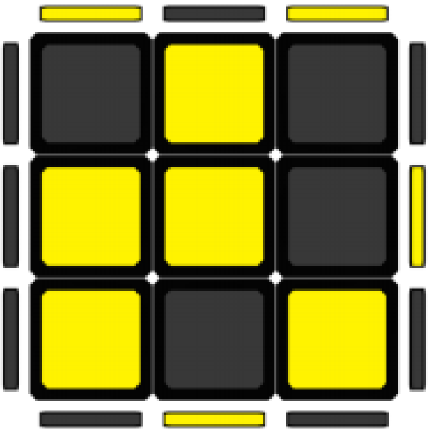


Figure 3. A specific example of last-layer preparation

After each movement, the border limits will be checked of position in orderly to save the output within the wanted, values [0, 255]. The permutation of position to guarantee a satisfying ownership with the aid a 256-value sequence with every item in the S-box is checked from start to finish prevent a certain item from being iterative in the S-Box and ensuring that a values are distributed within a large randomness with 21 algorithms as show figure (4) and to produce novel S-Box as shows in Figure (5).

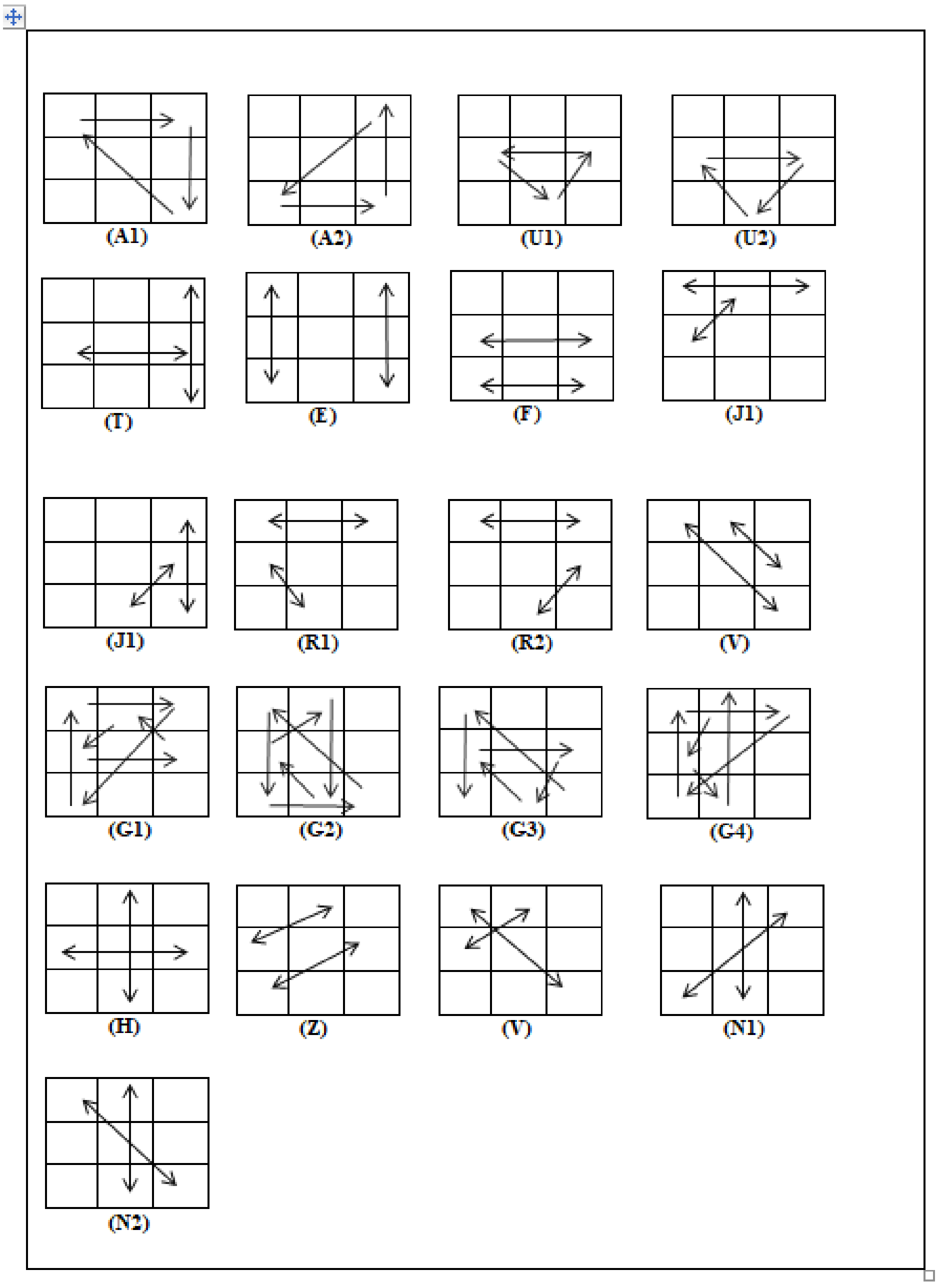


Figure 4. Permutation of last-layer Algorithms

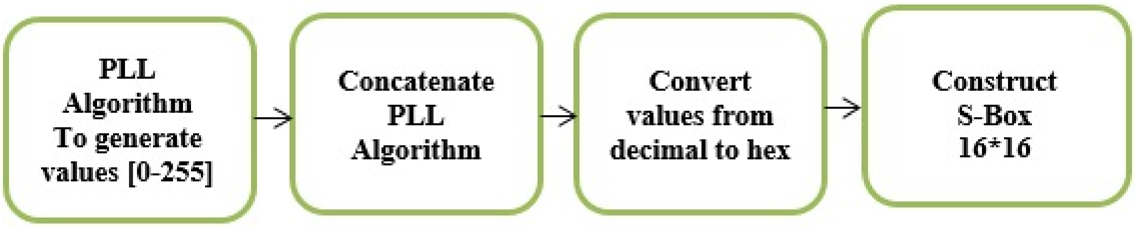

PLL Algorithm To generate values [0-255]
Concatenate PLL Algorithm
Convert values from decimal to hex
Construct S-Box 16*16

Figure 5. Proposed Construct New S-Box

The first and second stages in the above scheme use an algorithm PLL. It depends on solving the problem of different colors in the last layer of the cube by switching locations according to a certain movement and a certain direction as shown in algorithm 1.

Algorithm (1): Creating decimal code

```
Input: PLL Algorithm
Output: Decimal codes

 Begin
        1.Create a number sequence using PLL ( Tperm, Yperm, Hperm, Vperm ,Fperm
        ,Eperm,A1perm,A2perm,R1perm,R2perm,G1perm,G2perm,G3perm,G4perm,N1perm,
        N2perm,U1perm ,U2perm ,J1perm ,J2perm ,Zperm ,Sperm)

        2.Concatenate [Tperm, Yperm, Hperm ,Vperm ,Fperm, Eperm, A1perm, A2perm, R1perm,R2perm,
        G1perm,G2perm,G3perm,G4perm,N1perm, N2perm ,U1perm ,U2perm ,J1perm ,J2perm ,Zperm,
        Sperm ] and save the outcomes in ns11 // ns11 is a collection of one domination with decimal values
        called array
End
```

After creating the decimal code, it will be converted to hex code in order to create a new S-Box table for use in the encrypt process. S-Box building is represented by Algorithm 2. Table 1 shows the results.

Algorithm (2): Create S-Box (16*16)

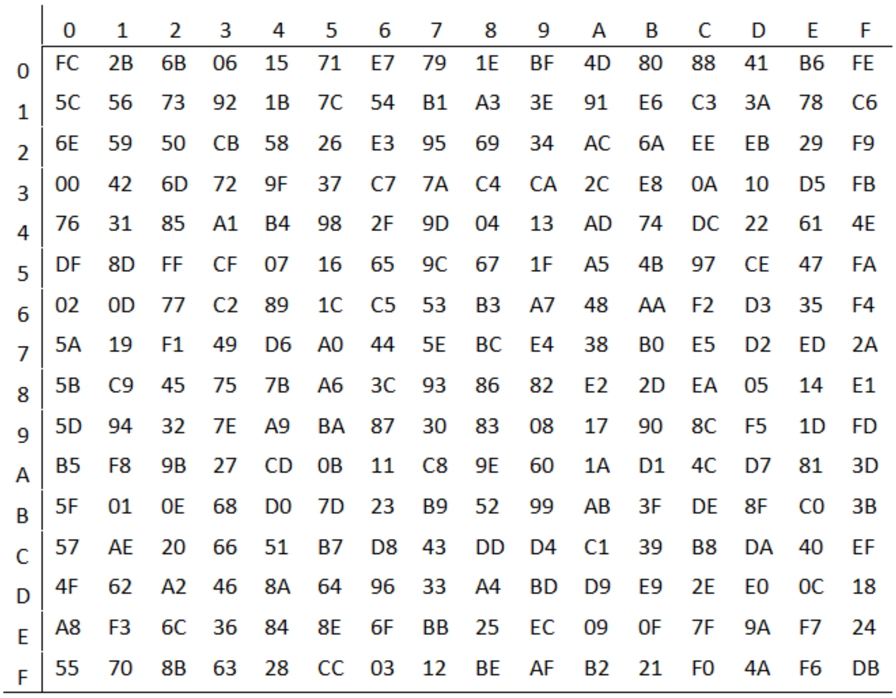
```
Input: decimal code
Output : Novel  S-Box  contain (16 Row and  16 Colomn)

Begin
        1.Convert decimal code to hex code by using dec2hex(ns11)
        2. Reshape from matrix one diminution [0-255] to [16*16] and save the results in dx // dx this
        array of two domination with hex values.
        3. For I =1 to 16
           3.1   delete any space from dx to construct new S-Box (16×16) and store the result in dm

End
```

In order to extract the original data, The S-Box inverse is used in the decryption process; algorithm 3 shows how to design the exact reverse for a novel S-Box. The initial value of the S-Box is implemented as follows: (FC). This value is divided in two numbers (F) to represent the address of the value in opposite S-Box (C). The row 0 and column 0 addresses of (FC) are obtained and combined to recognize the value that will be stored in the opposite S-Box. As a result, the value in row (F) and column (C) of the opposite S-Box is (00). Table (2) shows the results.

Algorithm (3): For the built S Box, use the opposite (Inverse) S -Box.

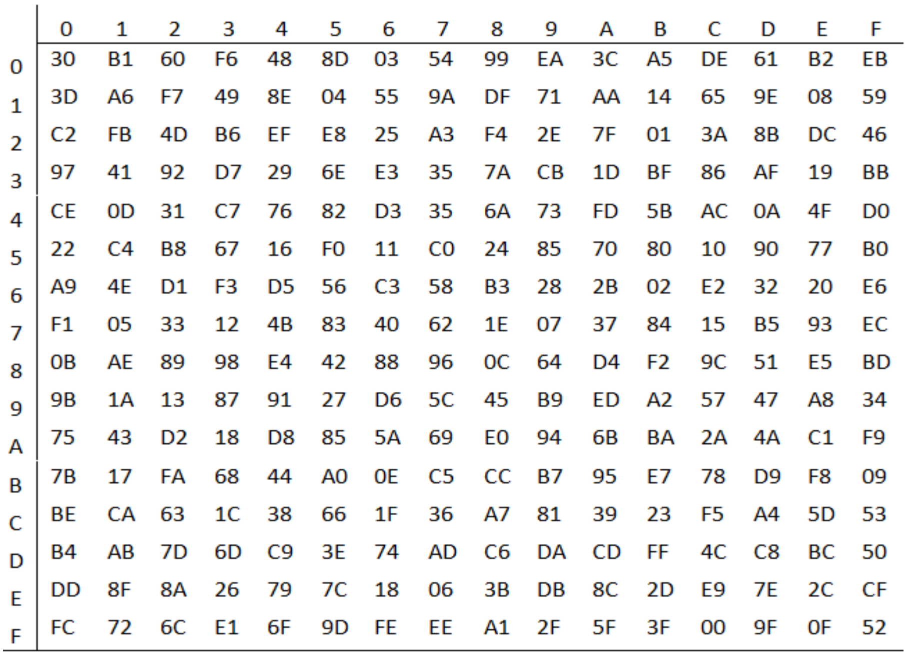
```
Input: S-Box array two dimension 16 Row and 16 Column.
Output: Opposite S-Box (16 ×16).
Begin
        For i=1 to 16
          For j=1 to 16

Each number in S-Box should be separated into two numbers (to appear the address of the value in S-
Box inverse).
The two numbers addresses are too obtained and joined to make the new value that will be stored in S-
Box inverse.
Steps 2 to 3 are repeated until the inverted S-Box is completed successfully.
End
```

Table 1. PLL algorithm to produce S-box



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	FC	2B	6B	06	15	71	E7	79	1E	BF	4D	80	88	41	B6	FE
1	5C	56	73	92	1B	7C	54	B1	A3	3E	91	E6	C3	3A	78	C6
2	6E	59	50	CB	58	26	E3	95	69	34	AC	6A	EE	EB	29	F9
3	00	42	6D	72	9F	37	C7	7A	C4	CA	2C	E8	0A	10	D5	FB
4	76	31	85	A1	B4	98	2F	9D	04	13	AD	74	DC	22	61	4E
5	DF	8D	FF	CF	07	16	65	9C	67	1F	A5	4B	97	CE	47	FA
6	02	0D	77	C2	89	1C	C5	53	B3	A7	48	AA	F2	D3	35	F4
7	5A	19	F1	49	D6	A0	44	5E	BC	E4	38	B0	E5	D2	ED	2A
8	5B	C9	45	75	7B	A6	3C	93	86	82	E2	2D	EA	05	14	E1
9	5D	94	32	7E	A9	BA	87	30	83	08	17	90	8C	F5	1D	FD
A	B5	F8	9B	27	CD	0B	11	C8	9E	60	1A	D1	4C	D7	81	3D
B	5F	01	0E	68	D0	7D	23	B9	52	99	AB	3F	DE	8F	C0	3B
C	57	AE	20	66	51	B7	D8	43	DD	D4	C1	39	B8	DA	40	EF
D	4F	62	A2	46	8A	64	96	33	A4	BD	D9	E9	2E	E0	0C	18
E	A8	F3	6C	36	84	8E	6F	BB	25	EC	09	0F	7F	9A	F7	24
F	55	70	8B	63	28	CC	03	12	BE	AF	B2	21	F0	4A	F6	DB

Table 2. The proposed Inverse algorithm to produce S-box



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	30	B1	60	F6	48	8D	03	54	99	EA	3C	A5	DE	61	B2	EB
1	3D	A6	F7	49	8E	04	55	9A	DF	71	AA	14	65	9E	08	59
2	C2	FB	4D	B6	EF	E8	25	A3	F4	2E	7F	01	3A	8B	DC	46
3	97	41	92	D7	29	6E	E3	35	7A	CB	1D	BF	86	AF	19	BB
4	CE	0D	31	C7	76	82	D3	35	6A	73	FD	5B	AC	0A	4F	D0
5	22	C4	B8	67	16	F0	11	C0	24	85	70	80	10	90	77	B0
6	A9	4E	D1	F3	D5	56	C3	58	B3	28	2B	02	E2	32	20	E6
7	F1	05	33	12	4B	83	40	62	1E	07	37	84	15	B5	93	EC
8	0B	AE	89	98	E4	42	88	96	0C	64	D4	F2	9C	51	E5	BD
9	9B	1A	13	87	91	27	D6	5C	45	B9	ED	A2	57	47	A8	34
A	75	43	D2	18	D8	85	5A	69	E0	94	6B	BA	2A	4A	C1	F9
B	7B	17	FA	68	44	A0	0E	C5	CC	B7	95	E7	78	D9	F8	09
C	BE	CA	63	1C	38	66	1F	36	A7	81	39	23	F5	A4	5D	53
D	B4	AB	7D	6D	C9	3E	74	AD	C6	DA	CD	FF	4C	C8	BC	50
E	DD	8F	8A	26	79	7C	18	06	3B	DB	8C	2D	E9	7E	2C	CF
F	FC	72	6C	E1	6F	9D	FE	EE	A1	2F	5F	3F	00	9F	0F	52

4. Performance comparison of the novel S-box

During this study, the S-Box was designed using the last layer permutation algorithms, and to prove the success of this proposal, it must pass the S-box criteria like avalanche, balanced, completeness, strict avalanche, invertability, that are compared to other S-boxes in related research and then shown below.

4.1. Bijection

Every S-box input value is set it to a single output value, essentially making the S-box a one-to-one function. Function. Such a property is required for the Inverse S-Box to correctly recover (back substitute) substitute values. Such as the letter C = 97 in ASCII code and letter C =43 in hexa cod, then the same latter is equal in proposed S-Box =A1, when, use the S-box inverse on A1, the 43 is output. Other words, the letter must return to the same value in the s-box, and this is one of the important factors to consider when designing the S-box: its ability to be reversible in require obtaining the main data.

4.2. Balanced criteria (BC)

Checking the diffusion of 0s and 1s in the output sequences is one of the most important S-box test criteria; this distribution should be balanced. [4,22]. The novel S-box is as seen in Table (3), that compares the BC test into two words used a novel S-box to different related research, it is balanced after this test in order to it has an equal or nearly equal number of 0s and 1s.

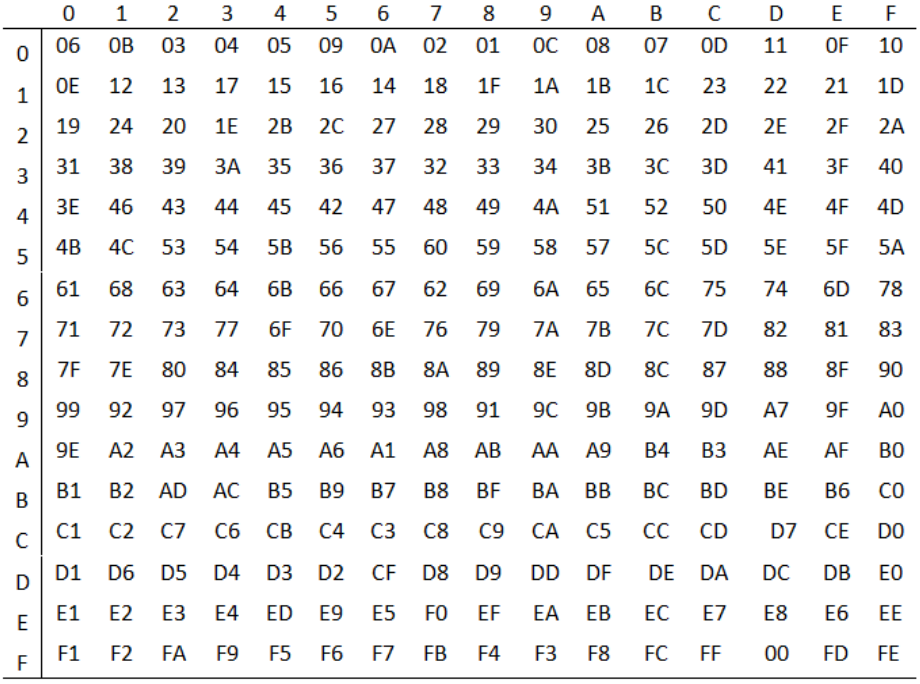
Table 3. The calculate the BC by the proposed method

Method	**Word1=Computer**		**Word2=ABMNOPQR**	
	no. 0's	**no. 1's**	**no. 0's**	**no. 1's**
Ref [1]	39	25	30	34
Ref [4]	33	31	31	33
Ref [5]	34	30	27	37
Ref [10]	35	29	28	36
Novel S-Box	**32**	**32**	**33**	**31**

4.3. Completeness criteria (CC)

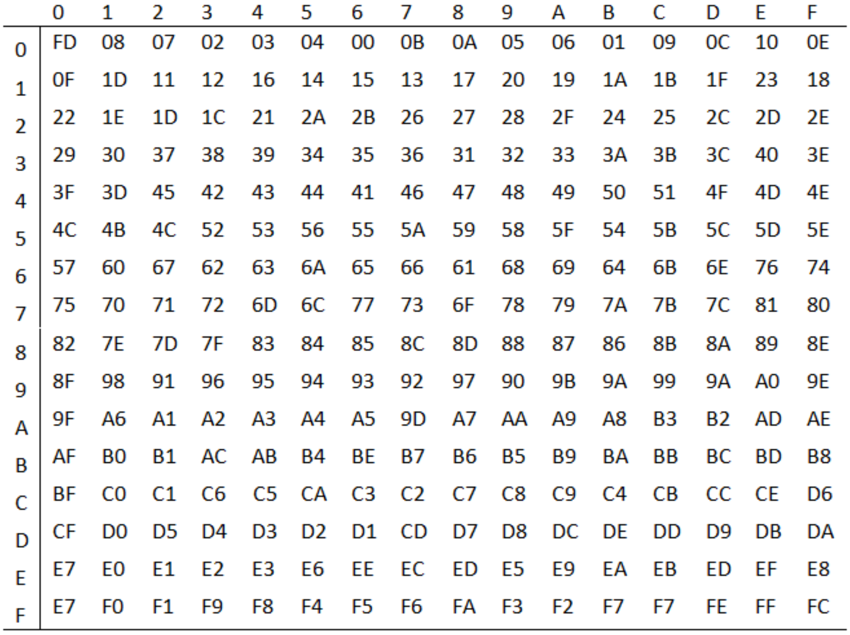
This is a measure; The input bits determine all of the output bits.. In the proposed algorithm, it depends on switching locations according to the use of 21 algorithms, where if an algorithm is used to change certain locations with other algorithms, I will get a new S-box, but if the locations of 21 algorithms are changed, I get a new S -box 100%100. Tables 3 and 4 show the results of these two examples.

Table 4. The S-box produced by new permutation of position



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	06	0B	03	04	05	09	0A	02	01	0C	08	07	0D	11	0F	10
1	0E	12	13	17	15	16	14	18	1F	1A	1B	1C	23	22	21	1D
2	19	24	20	1E	2B	2C	27	28	29	30	25	26	2D	2E	2F	2A
3	31	38	39	3A	35	36	37	32	33	34	3B	3C	3D	41	3F	40
4	3E	46	43	44	45	42	47	48	49	4A	51	52	50	4E	4F	4D
5	4B	4C	53	54	5B	56	55	60	59	58	57	5C	5D	5E	5F	5A
6	61	68	63	64	6B	66	67	62	69	6A	65	6C	75	74	6D	78
7	71	72	73	77	6F	70	6E	76	79	7A	7B	7C	7D	82	81	83
8	7F	7E	80	84	85	86	8B	8A	89	8E	8D	8C	87	88	8F	90
9	99	92	97	96	95	94	93	98	91	9C	9B	9A	9D	A7	9F	A0
A	9E	A2	A3	A4	A5	A6	A1	A8	AB	AA	A9	B4	B3	AE	AF	B0
B	B1	B2	AD	AC	B5	B9	B7	B8	BF	BA	BB	BC	BD	BE	B6	C0
C	C1	C2	C7	C6	CB	C4	C3	C8	C9	CA	C5	CC	CD	D7	CE	D0
D	D1	D6	D5	D4	D3	D2	CF	D8	D9	DD	DF	DE	DA	DC	DB	E0
E	E1	E2	E3	E4	ED	E9	E5	F0	EF	EA	EB	EC	E7	E8	E6	EE
F	F1	F2	FA	F9	F5	F6	F7	FB	F4	F3	F8	FC	FF	00	FD	FE

Table 5. The proposed Inverse algorithm to produce S-box



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	FD	08	07	02	03	04	00	0B	0A	05	06	01	09	0C	10	0E
1	0F	1D	11	12	16	14	15	13	17	20	19	1A	1B	1F	23	18
2	22	1E	1D	1C	21	2A	2B	26	27	28	2F	24	25	2C	2D	2E
3	29	30	37	38	39	34	35	36	31	32	33	3A	3B	3C	40	3E
4	3F	3D	45	42	43	44	41	46	47	48	49	50	51	4F	4D	4E
5	4C	4B	4C	52	53	56	55	5A	59	58	5F	54	5B	5C	5D	5E
6	57	60	67	62	63	6A	65	66	61	68	69	64	6B	6E	76	74
7	75	70	71	72	6D	6C	77	73	6F	78	79	7A	7B	7C	81	80
8	82	7E	7D	7F	83	84	85	8C	8D	88	87	86	8B	8A	89	8E
9	8F	98	91	96	95	94	93	92	97	90	9B	9A	99	9A	A0	9E
A	9F	A6	A1	A2	A3	A4	A5	9D	A7	AA	A9	A8	B3	B2	AD	AE
B	AF	B0	B1	AC	AB	B4	BE	B7	B6	B5	B9	BA	BB	BC	BD	B8
C	BF	C0	C1	C6	C5	CA	C3	C2	C7	C8	C9	C4	CB	CC	CE	D6
D	CF	D0	D5	D4	D3	D2	D1	CD	D7	D8	DC	DE	DD	D9	DB	DA
E	E7	E0	E1	E2	E3	E6	EE	EC	ED	E5	E9	EA	EB	ED	EF	E8
F	E7	F0	F1	F9	F8	F4	F5	F6	FA	F3	F2	F7	F7	FE	FF	FC

4.4. Avalanche criteria (AC)

A key standard in block ciphers is the avalanche property AC, which describes how well a small variation in the input bits tends for lead to a great (avalanche) variation at the value. Because of the result related to dispersion mathematics, this criterion, with such an optimal system of 0.5, is a likable feature to block cipher techniques. When design the block cipher, should commonly called result avalanche, which occurs when a single change in odd bit of input results in a fully various output. Compares the proposed method's AC value to the methods of Wan. et al. [5] and Bal. et al. [18] in table (4).

$$\text{AE} = \frac{\text{NO. of Flip. Bits in (output)Cipher Text}}{\text{NO. of every Bits in (output) Cipher Text}} \qquad (3)$$

The elements in the S-box algorithm most have a natural allocation between 0 and 1. This outcome is determined by the wherever you are, you must have a password. So according Equation, the letters should be distributed so at start of the algorithm (3). By calculating the above value for each letter A through Z and measuring ratio, this criterion is validated for our proposed method. Within case of a one-bit variation for every enter, Table (4) shows the example of how to evaluate the AC to use the proposed method and the methods that were compared.

Table 6. Calculate AC by the method proposed

Methods	**actual Infor.**	**Ascii**	**hex**	**binary**	**Change in S-Box**	**binary output**	**AC**
[10]	L	76	4C	0100 1100	BE	1011 1110	
change bit	M	77	4D	0100 1101	7F	0111 1111	3/8 =0.375
[22]	L	76	4C	0100 1100	14	0001 0100	
change bit	M	77	4D	0100 1101	83	1000 0011	5/8 =0.625
novel S-Box	L	76	4C	0100 1100	DC	1101 1100	
change bit	M	77	4D	0100 1101	22	0010 0010	7/8 =0.875

4.5. Strict avalanche criteria (SAC)

If one bit in the input changes half of the output bits changes, the S-Box fulfills the strict avalanche set of criteria. [16,34-38] is a phrase that can be used to describe a group of people In other words,SAC was achieved if both the completeness and avalanche criteria were met. The SAC is also achieved because the proposed technique meets these requirements. Table (5) shows an example of how to calculate the SAC using the proposed method and the compared methods if each entrant has a one-bit variation.

Table 7. SAC performance comparison

S-Box	Actual information	Average SAC
Ref [32]	A. . .Z (65. . .90)	0.4958
Ref [33]	A. . .Z (65. . .90)	0.4973
Ref [13]	A. . .Z (65. . .90)	0.4978
Ref [34]	A. . .Z (65. . .90)	0.4990
Ref [35]	A. . .Z (65. . .90)	0.4980
Novel S-Box	A. . .Z (65. . .90)	0.5288

5. Conclusion

To encrypt data in our daily lives, the world of cryptography has become an important aspect of our daily lives, so a number of researchers have resorted to developing encryption algorithms Stream and Block cipher, as well as the development of the construction of the S-box used in the encryption algorithm. Where we discussed in this research the method of the Permutation last layer to build the S-box. The build the S-Box and its opposite in this proposal take only a few milliseconds. S-box tests such that savalanche, balance, strict avalanche, completeness, and invertablity were used to evaluate the novel S-box. These statistical exam were too compare to different studies in the field and get the dynamic S-box when change in the permutation of position by use 2^{21} permutation. These findings suggest the novel S-box has very good encryption ownership, allowing to secure communication, and that due to the speed with which S-box and its inverse are constructed, this S-box is appropriate into used in lightweight cryptography for appropriate devices. In the future work, we can use different Algorithm such first 2 layers to construct the S-box to growth the security level.

Reference

[1] Rana M Zaki, Hala Bahjat Abdul Wahab," 4G Network Security Algorithms: Overview”, International Journal of Interactive Mobile Technologies(iJIM),.Vol. 15 Issue 16, pp.127-143,2021.

[2] Rana M Zaki, Teaba WA Khairi, Akbas Ezaldeen Ali," Secure Data Sharing Based on Linear Congruetial Method in Cloud Computing", Springer, Singapore, Next Generation of Internet of Things,pp.129-140,2021

[3] B. Jana, M. Chakraborty, T. Mandal, and M. Kule, "An Overview on Security Issues in Modern Cryptographic Techniques," in Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT), 2018, pp. 26-27.

[4] M. M. Lauridsen, "Design and Analysis of Symmetric Primitives," 2016.

[5] S. Sahmoud, W. Elmasry, and S. Abudalfa, "Enhancement the Security of AES Against Modern Attacks by Using Variable Key Block Cipher," Int. Arab. J. e Technol., vol. 3, no. 1, pp. 17-26, 2013.

[6] H. H. Salih, A. T. Sadiq, and A. K. Farhan, "Proposal of New Block Cipher Algorithm," Engineering and Technology Journal, vol. 28, no. 10, 2010.

[7] A. K. Farhan, S. M. Kadhem, N. Monem, and D. Saad, "New Approach for Modifying DES Algorithm by Using Multiple Keys Depend on Heuristic Search Algorithm," Engineering and Technology Journal, vol. 31, no. 1 Part (B) Scientific, 2013.

[8] H. B. A. Wahab and S. F. Amir, "Efficient Digital Watermark key Generation Using Hexagonal Structure and parametric Lagrange Curve," Eng. & Tech. Journal, vol. 33, no. 2, pp. 192-203, 2015.

[9] H. Alrikabi, and H. Tauma "Enhanced Data Security of Communication System using Combined Encryption and Steganography," International Journal of Interactive Mobile Technologies, vol. 15, no. 16, pp. 144-157, 2021.

[10] J. Juremi, R. Mahmod, and S. Sulaiman, "A proposal for improving AES S-box with rotation and key-dependent," in Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012: IEEE, pp. 38-42.

[11] H. Wang, H. Zheng, B. Hu, and H. Tang, "Improved lightweight encryption algorithm based on optimized S-box," in 2013 International Conference on Computational and Information Sciences, 2013: IEEE, pp. 734-737.

[12] H. T. Salim., and N. A. Jasim, "Design and Implementation of Smart City Applications Based on the Internet of Things," International Journal of Interactive Mobile Technologies (iJIM), vol. 15, no. 13, pp. 4-15, 2021

[13] A. Al-zubidi, R. K. Hasoun, S. H. Hashim, and Haider Th, "Mobile Application to Detect Covid-19 pandemic by using Classification Techniques: Proposed System," International Journal of Interactive Mobile Technologies, vol. 15, no. 16, pp. 34-51, 2021.

[14] O. B. Sahoo, D. K. Kole, and H. Rahaman, "An optimized S-box for advanced encryption standard (AES) design," in 2012 International Conference on Advances in Computing and Communications, 2012: IEEE, pp. 154-157.

[15] N. H. UbaidurRahman, C. Balamurugan, and R. Mariappan, "A novel DNA computing based encryption and decryption algorithm," Procedia Computer Science, vol. 46, pp. 463-475, 2015.

[16] A. K. Farhan, R. S. Ali, H. Natiq, and N. M. Al-Saidi, "A new S-box generation algorithm based on multistability behavior of a plasma perturbation model," IEEE Access, vol. 7, pp. 124914-124924, 2019.

[17] M. K. Balajee and J. Gnanasekar, "Evaluation of key dependent S-box based data security algorithm using Hamming distance and balanced output," Tem Journal, vol. 5, no. 1, p. 67, 2016.

[18] Y. Naseer, T. Shah, D. Shah, and S. Hussain, "A novel algorithm of constructing highly nonlinear sp-boxes," Cryptography, vol. 3, no. 1, p. 6, 2019.

[19] C.-M. Ou, "Design of block ciphers by simple chaotic functions," IEEE computational intelligence magazine, vol. 3, no. 2, pp. 54-59, 2008.

[20] S. Garg and D. Upadhyay, "S-Box design approaches: critical analysis and future directions," Int J Adv Res Comput Sci Electron Eng IJARCSEE, vol. 2, 2013.

[21] A. A. Alzaidi, M. Ahmad, M. N. Doja, E. Al Solami, and M. S. Beg, "A new 1D chaotic map and $\beta $-hill climbing for generating substitution-boxes," IEEE Access, vol. 6, pp. 55405-55418, 2018.

[22] Q. Lu, C. Zhu, and G. Wang, "A novel S-box design algorithm based on a new compound chaotic system," Entropy, vol. 21, no. 10, p. 1004, 2019.

[23] M. Ahmad, N. Mittal, P. Garg, and M. M. Khan, "Efficient cryptographic substitution box design using travelling salesman problem and chaos," Perspectives in Science, vol. 8, pp. 465-468, 2016.

[24] H. A. Ahmed, M. F. Zolkipli, and M. Ahmad, "A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map," Neural Computing and Applications, vol. 31, no. 11, pp. 7201-7210, 2019.

[25] E. Al Solami, M. Ahmad, C. Volos, M. N. Doja, and M. M. S. Beg, "A new hyperchaotic system-based design for efficient bijective substitution-boxes," Entropy, vol. 20, no. 7, p. 525, 2018.

[26] A. A. Alzaidi, M. Ahmad, H. S. Ahmed, and E. A. Solami, "Sine-cosine optimization-based bijective substitution-boxes construction using enhanced dynamics of chaotic map," Complexity, vol. 2018, 2018.

[27] D. Rajavel and S. Shantharajah, "Scrambling algorithm for encryption of text using cube rotation artificial intelligence technique," 2016.

[28] R. Dhandabani, S. S. Periyasamy, P. Theagarajan, and A. K. Sangaiah, "Six-face cubical key encryption and decryption based on product cipher using hybridisation and Rubik's cubes," IET Networks, vol. 7, no. 5, pp. 313-320, 2018.

[29] O. A. Dawood, A. M. S. Rahma, and A. M. J. A. Hossen, "Generalized method for constructing magic cube by folded magic squares," International Journal of Intelligent Systems and Applications, vol. 8, no. 1, p. 1, 2016.

[30] C. A. Steinparz, A. P. Hinterreiter, H. Stitz, and M. Streit, "Visualization of Rubik's Cube Solution Algorithms," in EuroVA@ EuroVis, 2019, pp. 19-23.

[31] S. Pochmann, "Analyzing Human Solving Methods for Rubik's Cube and similar Puzzles," 2008.

[32] A. Y. Rahardian, R. Munir, and H. Harlili, "Rubikstega: A Novel Noiseless Steganography Method in Rubik's Cube," in International Conference on Information Technology, Engineering, Science & its Applications, 2018.

[33] V. Dan, G. Harja, and I. Naşcu, "Advanced Rubik's Cube Algorithmic Solver," in 2021 7th International Conference on Automation, Robotics and Applications (ICARA), 2021: IEEE, pp. 90-94.

[34] A. A. Abd el-Latif, B. Abd-el-Atty, M. Amin, and A. M. Iliyasu, "Quantum-inspired cascaded discrete-time quantum walks with induced chaotic dynamics and cryptographic applications," Scientific reports, vol. 10, no. 1, pp. 1-16, 2020.

[35] A. A. Abd El-Latif, B. Abd-El-Atty, W. Mazurczyk, C. Fung, and S. E. Venegas-Andraca, "Secure data encryption based on quantum walks for 5G Internet of Things scenario," IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 118-131, 2020.

[36] W. Gao, B. Idrees, S. Zafar, and T. Rashid, "Construction of Nonlinear Component of Block Cipher by Action of Modular Group PSL (2, Z) on Projective Line PL (GF (2 8))," IEEE Access, vol. 8, pp. 136736-136749, 2020.

[37] A. H. Zahid, E. Al-Solami, and M. Ahmad, "A novel modular approach based substitution-box design for image encryption," IEEE Access, vol. 8, pp. 150326-150340, 2020.

[38] T. W. A. Khairi , Rana M. Zaki and W. A. Mahmood, "Stock Price Prediction using Technical, Fundamental and News based Approach", IEEE Conference of Computer Sciences (SCCS), University of Technology – Iraq, pp. 177–181, 8852599,2019.