XOR Vector Space based S-Box generation and its Application to DES and AES for the Time-Efficient Image Encryption

Arun Upadhyaya[1], Shantharama Rai C.[2] and Ganesh Aithal[3]

[1]Department of Electronics and Communication, Shri Madhwa Vadiraja Institute of Technology and Management, Bantakal, Udupi, Affiliated to Visvesaraya Technological University, India. [2]Department of Electronics and Communication, A J Institute of Engineering and Technology, Kottara, Mangalore, Affiliated to Visvesaraya Technological University, India. [3]Department of Computer Science and Engineering, Shri Madhwa Vadiraja Institute of Technology andManagement, Bantakal, Udupi, Affiliated to Visvesaraya Technological University, 574115, India. arunsse2012@gmail.com, csraicec@gmail.com, ganeshaithal@gmail.com

Abstract: A substitution box (S-Box) is a fundamental component used in DES and AES. This is used to achieve non-linearity in encryption. The use of S-Box also introduces the property of confusion in symmetric cipher systems. A finite field in the multiplicative inverses of GF (2^8) is the basis for the generation of the S-Box in the AES algorithm. This paper presents a novel approach that utilizes the XOR vector space for generating the S-Box. The XOR vector space is constructed using two-dimensional values comprising vectors of size of 2^b. Following this, the resulting S-Box is employed in standard algorithms such as DES and AES to encrypt the image. Furthermore, this paper assesses the security aspects of the proposed algorithm by measuring various parameters. The result is compared with methods that generate various S-Boxes for the DES and AES algorithms. This suggests that the proposed method is relatively better.

Keywords: S-Box; XOR vector space; Image encryption; DES; AES.

1. Introduction

Since the fast and widespread adoption of various communication and connectivity methods, exchanging information over the Internet has been possible. Social media's primary mode of data transmission is sharing digital images. Moreover, images offer valuable and essential information, so they are considered private images [1].

The exchange of this information is crucial in various sectors such as such as banking, healthcare and telecommunications [2]-[3]. The standard encryptions methods like DES and AES are employed to secure the transmission of sensitive data [4]. For a block cipher to be considered safe, the confusion and diffusion properties [5] must be satisfied. Block cipher methods utilize cryptographic components known as substitution boxes (S-Boxes). This method is employed to fulfill the properties of confusion [6]. DES and AES, two renowned encryption standards, are often used for data transfer. Standard encryption algorithms require more time and resources for execution, thus demanding high-performance systems [7].

The S-Box's non-linear nature makes it difficult for differential cryptanalysis to exploit patterns, providing confusion, strength, and resistance [8]. The effectiveness of DES and AES encryption algorithms largely depends on their S-Boxes. These cryptographic methods utilize the static S-Box. When the S-Box is pre-available, potential eavesdroppers could attack with ease. However, by employing a key to introduce randomness in the selection of the S-Box, as discussed in [9]-[10], this approach offers greater substantiation than using a static S-Box. Many researchers are working towards increasing the strength of the S-Box thereby improving the security of encryption algorithms. The authors in [11] proposed a novel method which uses modular operations to generate the dynamic substitution-box. An effective heuristic evolution strategy is used which has the potential to improve the nonlinearity of the S-box. Many publications [12]-[13] use some of the optimization algorithms to generate the S-Box. The drawback of such methods is the computational load in optimization techniques [14]. In addition,

Received: December 12th, 2023. Accepted: May 30th, 2024
DOI: 10.15676/ijeei.2024.16.2.3

timing and side-channel attacks may make optimization algorithms to insecure [15].

The researchers [16] introduced a new technique based on chaotic maps for generating an initial S-Box. This S-Box undergoes dynamic permutation to enhance confusion and bolster security. Consequently, the process of permutation enhances encryption security. In [17], a novel chaos-dependent approach was presented, leveraging matrix rotation and affine transformation. The concept of chaos-based rotation metrics significantly strengthens AES S-Box, thereby enhancing cryptosystem security. The use of dynamic S-Box renders algebraic attacks ineffective. Additionally, the paper outlines an innovative and efficient method for encrypting images through a DNA based conversion process employing a algebraic function which is nonlinear.

The study in [18] presented an innovative heuristic evolution strategy centered on the nonlinearity property of the S-Box. The dynamic selection of S-Box enhances the security of the ciphertext by utilizing a random key. In [19], a polynomial transformation for constructing the S-Box is suggested. This method is straightforward, efficient, and proven to construct a resilient S-Box. Additionally, in [20], a creative technique utilizing linear transformation is proposed to construct a robust S-Box.

Although many researchers use different approaches to generate the S-Box, its strength is improved while the complexity of its generation method also increases. Additionally, generating the S-Box requires more time. This study focuses on the construction of an S-Box using XOR vector space. The proposed method takes comparatively less time [21]-[22] to generate the S-Box. As well as being equally effective against various types of attack.

A value from a S-Box is dynamically selected using a key-dependent approach. In well-known cipher systems such as DES and AES, the selected value is then used for image encryption.

The structure of the remaining portion of this article is as follows: Section 2 clarifies the construction of the XOR vector space. A dynamic S-Box selection technique based on XOR vector space is introduced in Section 3. The results of the experiments and the analysis of the suggested approach are presented in Section 4. The paper concludes with Section 5.

2. Generation of XOR Vector Space

The generation of the XOR vector space involves performing the XOR operation on the elements of the row and column integers. Each row can be generated by performing the XOR operation on the individual columns with a specific row integer. Alternatively, this process can be accomplished by appropriately shifting the integers of the first row according to the binary representation of the selected row integer.

Using the technique of repeated division by 2, every decimal integer can be expressed in binary or radix-2. Reverse order is used for writing the remainders, beginning with the final quotient. For example, the decimal number can be represented using binary as $(99)_{10} = (1100011)_2 = 2^6 + 2^5 + 2^1 + 2^0$. In the next section, it will be clear how this conversion relates to the creation of XOR vector space.

Consider an XOR vector space of $2^3 \times 2^3 = 8 \times 8$, as illustrated in Table 1. The decimal integers in the 0^{th} row range from 0 to 2^3-1. The integer sequence given in the 0^{th} row is permuted in every other row. Two consecutive integers from the 0^{th} row are discovered to be interchanged to form the first row. In the second row, two of the four numbers from the 0^{th} row are consecutively switched. The four subsequent numbers from the 0^{th} row are switched in the 4th row, and so forth.

According to the previous explanation, a pattern of number shifting appears to be involved in order to obtain particular rows of the vector space during the entire procedure. This pattern can be converted into an equivalent process that shifts a set of integers according to the selected row number. The 0^{th} row consists of the sequence of positive integers, namely 0,1,2,3,4,5,6,7.

Table 1. XOR vector space (square matrix) of $2^3 \times 2^3 = 8 \times 8$

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	3	2	5	4	7	6
2	2	3	0	1	6	7	4	5
3	3	2	1	0	7	6	5	4
4	4	5	6	7	0	1	2	3
5	5	4	7	6	1	0	3	2
6	6	7	4	5	2	3	0	1
7	7	6	5	4	3	2	1	0

As a next step, we will consider an operator denoted '#' that shifts integers within row 0. Prefixes associated with the '#' operator are derived from the selected row number in binary format and determine how these integers are arranged.

In general, let's consider an integer sequence from 0 to $2^k - 1$. A vector space is represented by k bits where k is the number of bits required for each digit. Let this sequence be row 0, given by $a_0, a_1, a_2, \ldots (a_{2^k} - 1)$.

Any selected row, let's say $(x)_{10}$ is obtained with the permutation of row 0 with the mentioned procedure. In this case, $(x)_{10}$ is represented by the binary weights as follows:

$$(x)_{10} = i_b 2^b + i_{b-1} 2^{b-1} + i_{b-2} 2^{b-2} + \cdots i_0 2^0 \quad (1)$$

Where 2^b , 2^{b-1} ... are the weights of the corresponding binary integers i_b, $i_{b-1} \ldots i_0$

For example $(6)_{10}$ is equal to $1\times 2^2 + 1\times 2^1 + 0\times 2^0 = (110)_2$

The shift operation can now be used to generate row x as follows

$$(i_b 2^b \#) + (i_{b-1} 2^{b-1} \#) + (i_{b-2} 2^{b-2} \#) + \cdots + (i_0 2^0 \#) \quad (2)$$

The plus sign (+) in this case denotes the subsequent shift operation that must be applied to the resulting string of integers following the preceding shift operation.

According to the operation, row x can be expressed as shifting integers as follows:

Step 1: Convert the selected integer row *x* to binary and represent it as shown in Equation 1.

Step 2: Get the Equation for performing the shifting operation according to Equation 2.

Step 3: Perform the shifting operation as per the prefix # operator for the weights with a binary integer equal to 1 and the binary integer with 0 is not considered for shift operation.

- If the prefix of # is 2^b then 2^b numbers of integers in row 0 are interchanged sequentially.
  For example, Let's consider row 0 – 0,1,2,3,4,5,6,7 will be operated by $2^1\#$ with b = 1. This indicates two consecutive integers in the row 0 are shifted to obtain a new row 2,3,0,1,6,7,4,5.
- In the case of multiple operations with the '+' sign, the shifting of the higher order weight has to be done first i.e., $(i_b 2^b \#)$ if $i_b = 1$ and then $(i_{b-1} 2^{b-1} \#)$ if $i_{b-1} = 1$ and further following in the same order.

  For example, let us consider the generation of the row 5 of Table 1.
  The binary equivalent of $(5)_{10}$ is $(101)_2$. In terms of equivalent weights, this can be written as $(5)_{10} = (1 \times 2^2 + 0 \times 2^2 + 1 \times 2^0)$

  The shifting operation is represented as $(2^2\# + 2^0\#)$ for *i* =1.
  Since $2^2 = 4$, $(2^2\#)$ is shifting four consecutive digits continuously from row 0, which generates *4,5,6,7,0,1,2,3.*

Next, the operation $(2^0\#)$ is performed on the previously obtained sequence i.e. *4,5,6,7,0,1,2,3*. Here 2^0=1 and hence, each consecutive digit is shifted, which generates the final sequence as *5,4,7,6,1,0,3,2*. Therefore the fifth row is represented as 5,4,7,6,1,0,3,2.

This initial sequence $a_0, a_1, a_2, \ldots (a_{2^k} - 1)$ can be used to represent the operation in general. It is divided into groups, so that each group contains 2^b number of integers, as indicated in Equation 3.

$$a_0, a_1, a_2, \ldots \underbrace{\text{1st } 2^b \text{ Integers}}\ldots \underbrace{a_2 \quad \text{2nd } 2^b \text{ Integers}}\ldots \underbrace{a_{3(2^b)-1}, \quad \text{3rd } 2^b \text{ Integers}}$$
$$\underbrace{a_{3(2^b)}, a_{3(2^b)+1}\ldots a_{4(2^b)-1}}, \ldots \underbrace{a_{2^k-3}, a_{2^k-2}, a_{2^k-1}} \tag{3}$$
$$\text{4th } 2^b \text{ Integers} \qquad \text{Final } 2^b \text{ Integers}$$

where $b < k$.

Let us consider a row 2^b. This leads to a shift operation of $2^b\#$. Here the shift operation is applied to the sequence given in equation 3 by shifting 2^b consecutive number of digits. The new sequence is obtained is as given Equation 4.

$$a_{2^b}, a_{2^b+1}, \ldots a_{2(2^b)-1}, a_0, a_1, a_2, \ldots . a_{2^b-1}, a_{3(2^b)}, a_{3(2^b)+1}\ldots, a_{2(2^b)}, a_{2(2^b)+1},$$
$$\ldots, \ldots a_{2^k-3}, a_{2^k-2}, a_{2^k-1} \ldots \tag{4}$$

Applying the subsequent shift i.e., $(2^{b-1}\#)$ to the last sequence obtained involves shifting a consecutive 2^{b-1} digits. Taking into account each appropriate weight with the binary integer i=1, this procedure will continue. Similarly, the subsequent shift operations involving binary numbers with *i* = 1 will be carried out in accordance with the previously described procedure.

The process of constructing a specific row is shown in Algorithm 1.

Algorithm 1: Shift Algorithm (#) to generate specific row of a S-box
Step 1: Input: K, x Initialize Row 0 with integers 0 to $2^K - 1$
Step 2: Represent $(X)_{10}$ in binary as $(x)_{10} = i_b 2^b + i_{b-1}2^{b-1} + i_{b-2}2^{b-2} + \cdots i_0 2^0$
Step 3: Initialize k = 0 for k=0 to b do if $\boldsymbol{i_{b-k} = 1}$ then Perform shift operation according to equation 3 and 4 end if *k = k+1* end
Step 4: output

The time complexity of the suggested method is dependent on the number of shift operations and, in turn, on '*x*', the specific row that is to be shifted. This relationship can be confirmed by showing that the complexity of the shift operation is $O(\log_2(x))$. However, when using simple hardware specifically designed for shift operations, this complexity is effectively reduced to a constant time, denoted as $O(1)$ instead of $O(\log_2(x))$.

The conventional cipher systems, such as DES and AES, employ the use of S-Boxes. However, in this particular paper, the S-Box is replaced by a XOR vector space. Table 2 illustrates a 16×16 matrix representing a XOR vector space. The first row of this matrix is a sequence of numbers from 0 to 15. The subsequent rows are formed by shifting the integers of the first row. This is concatenated twice so as to get a single 32×16 matrix, which is required

in DES. An S-Box of 16×16 size with random numbers from 0 to 255 is required for the AES algorithm. To achieve this, a much larger i.e. 256×256 matrix is constructed using XOR vector space, with the first row of this matrix being the sequence of numbers from 0 to 255. The subsequent rows are generated using the shift operation. From this wide matrix, a single row, which consists of a permutation of numbers from 0 to 255 is selected randomly and transformed into a 16×16 matrix as shown in Table 3. This matrix is selected from the 190^{th} row of the 256×256 matrix obtained from performing shifting operations. The chosen row of the XOR vector space is rearranged into a 16×16 S-Box format. It can be seen that the integers are arranged in random order. The matrices generated using XOR vector space is shown in Table 2 and Table 3. These S-Boxes serve as an important component in the substitution process for both DES and AES encryption algorithms, and the subsequent section provides additional information on their implementation.

3. Implementation of S-Box in DES and AES based on XOR vector space

A. Introduction

A symmetric-key block cipher, the DES, was published on November 23, 1977, as Federal Information Processing Standard (FIPS) 46 by the National Bureau of Standards (NBS). DES, as described in [23], employs a Feistel cipher and consists of 16 rounds of operation. This method has three main operations: the initial and final permutations, the round function, and the key schedule. During the round function stage, DES utilizes eight S-Boxes, each of size 4×16, totaling to 32×16 S-Boxes. The input to the S-Box consists of a 48-bit expansion permutation block. A 48-bit block is transformed into a 32-bit block by the S-Box.

NIST initially established AES in 2001 [24]. The AES encryption algorithm is a symmetric block cipher that operates considering a message block size of 128 bits. Using keys of 128/192/256 bits, these individual blocks are converted. Depending on the chosen key length, the conversion is spread out over 10/12/14 rounds. There are four steps in each round: Add Round Key, Mix-Columns, Sub-Bytes, and Shift-Rows. Using a 16×16 S-Box with 256 total components, AES employs the SubBytes function to randomly organize the values 0 through 255.

B. Proposed Methodologies

This section outlines two distinct approaches that are used in the paper. In the first method, S-Box generated using XOR vector space is used for both the DES and AES. In the second method, the S-Box is obtained by combining the rows of both the standard S-Box and those obtained from the XOR vector space using of random selection of the S-Boxes.

In the first case, the generated XOR vector space, which is a square matrix with a size of 16×16 shown in the table, is concatenated twice in order to generate the 32×16 matrix. This concatenated XOR vector space matrix is used as an S-Box in DES algorithm.

In the second scenario, XOR vector space produces a 256×256 matrix for the AES. A pseudo-random number generator (PRN) is used to randomly select a row from this 256×256 matrix. A linear feedback shift register (LFSR) with an initial state of [1, 1, 1, 1, 0, 0, 0, 0] and tap positions of [1, 8, 5, 8] is used as the PRN generator. In the AES encryption method, a selected row is turned into a square matrix of size 16×16 and employed as an S-Box. Table 3 displays the 16×16 S-Box.

In the third case, encryption is carried out using DES algorithm by constructing the hybrid S-Box. The 32×16 matrix is structured by incorporating four 16×16 S-Box of the standard DES, and the remaining four 16×16 are derived from the XOR vector space..

In the fourth scenario for AES, a 16×16 hybrid S-Box is created by merging an 8×16 matrix from the standard S-Box with another 8×16 matrix selected from the specified row of the 256×256 XOR vector space illustrated in Table 3.

Table 2. XOR Vector Space of dimension *16×16* generated using shift operation

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14
2	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13
3	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12
4	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11
5	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10
6	6	7	4	5	2	3	0	1	14	15	12	13	10	11	8	9
7	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
8	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
9	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6
10	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5
11	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4
12	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3
13	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2
14	14	15	12	13	10	11	8	9	6	7	4	5	2	3	0	1
15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 3. S-Box used in AES based on XOR Vector Space

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

4. Security Analyses

A robust encryption algorithm exhibits desirable attributes capable of withstanding various known attacks. A thorough evaluation of security encompasses several parameters, including the histogram of pixel occurrences, Number of Pixel Change Rate (NPCR), Unified Average Changing Rate (UACI), Mean Square Error (MSE), and Peak Signal-to-Noise Ratio (PSNR). In this analysis, a standard color image of size 256×256 depicting DEBLUR is utilized.

A. Visual Analysis

Figure 1 shows the results obtained for the DES algorithm. Figure 1(a) is the plain image and Figures 1(c), 1(e), and 1(g) shows the encrypted image for DES algorithm using the standard method, XOR vector space, and, hybrid S-Box, respectively. Similarly, Figure 2 shows the results obtained for the AES algorithm. Figures 2(c), 2(e), and 2(g) are the encrypted images for

standard method, XOR vector space, and, hybrid S-Box, respectively. In all these figures, the pixels are randomly scattered with no traces of the plain image.

B. Histogram Analysis

In plain images and encrypted images, an image histogram shows the frequency of pixel values. When the distribution of the encrypted image is uniform, the encrypted image is more resistant to statistical attacks.

Figure 1(b) illustrates the occurrence number of pixels of the original image shown in Figure 1(a). The histograms representing the distribution of pixel counts in encrypted images Figures 1(d), 1(f)) and 1(h) are depicted in Figures 1(c), 1(e)) and 1(g) respectively.

Figures 2(d), 2(f), and 2(h) shows the histograms of Figures 2(c), 2(e), and 2(g). The analysis shows that the distribution of pixels in the histograms of encrypted images is uniform, indicating resistance to statistical attacks. Consequently, the XOR vector space encryption demonstrates equal strength against such attacks.

Figure 2(h) displays the histogram of the fourth method, which uses a hybrid S-Box in AES. The distribution is not uniform due to the repetition of integers in the S-Box. S-Boxes created by combining standard AES and XOR vector spaces contain repeated values instead of having numbers from 0 to 255 in random order.

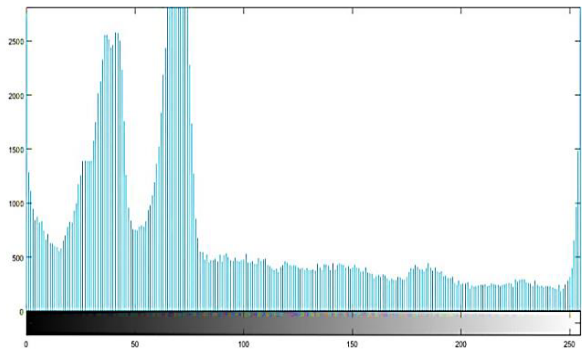
C. Entropy Analysis

A metric for quantifying the randomness or uncertainty in encrypted data is entropy [26]. This is expressed as

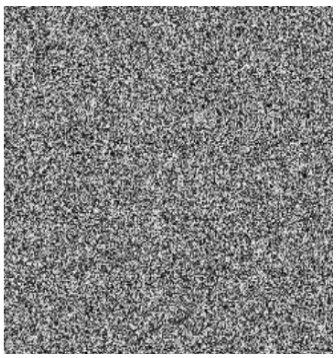
$$H(s) = -\sum_{0}^{255} P(s_i) log_2 P(s_i) \quad (5)$$

Given a set S of pixel values in an image, where $si \in S$, the probability of *si* occurring in the image is denoted as *P*(*si*). In the case of an 8-bit grayscale image with pixels ranging from 0 to 255, the image has a bit count of eight and a maximum entropy of 8 [27]. The entropy is determined through this approach after transforming a color image into grayscale. Table 4 compares the entropy of the proposed approach with that of conventional methods and several recent works. he results clearly indicate that the proposed method is close to achieving maximum entropy, demonstrating its comparative strength.

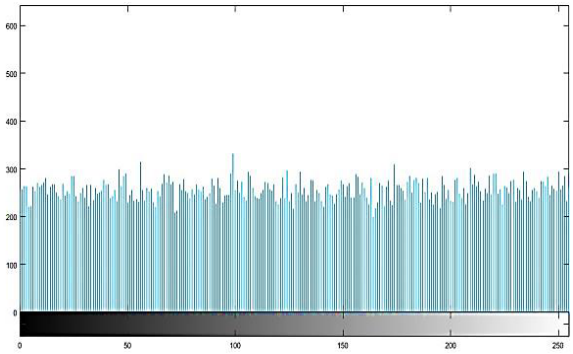
(a)



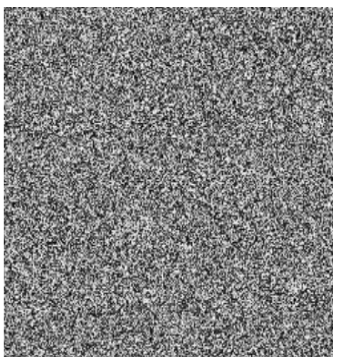
(b)



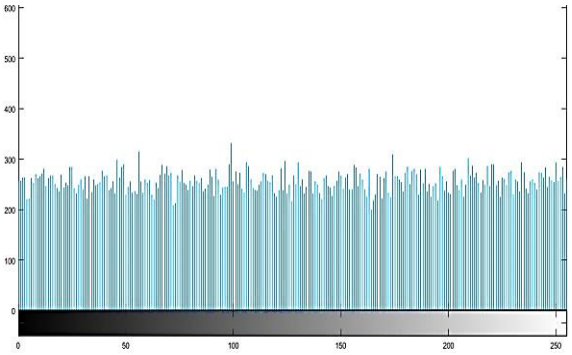
(c)



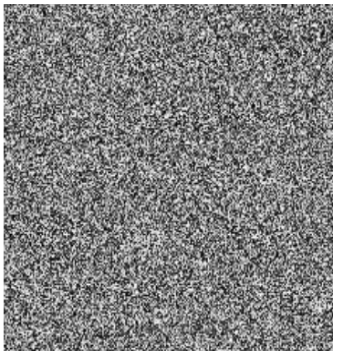
(d)



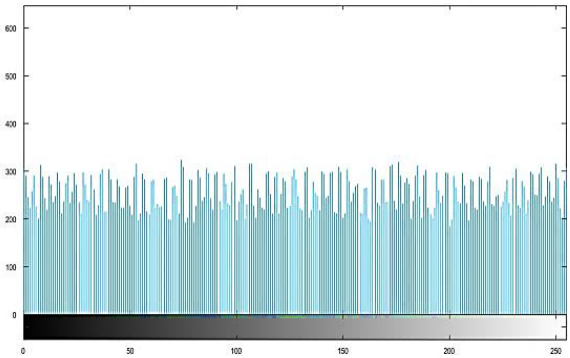
(e)



(f)



(g)



(h)

Figure 2. Image encryption and Histogram of the number of occurrences of pixels of an image with the DES Algorithm (a) Test image (b) Histogram of the test image (c) Cipher image of 1(a) using the standard method of DES (d) Histogram of 1(c) using the standard method of DES (e) Cipher image of 1(a) using the proposed XOR vector space (f) Histogram of 1(e) using the proposed XOR vector space (g) Cipher image of 1(a) using the hybrid S-Box (h) Histogram of 1(g) using the hybrid S-Box.

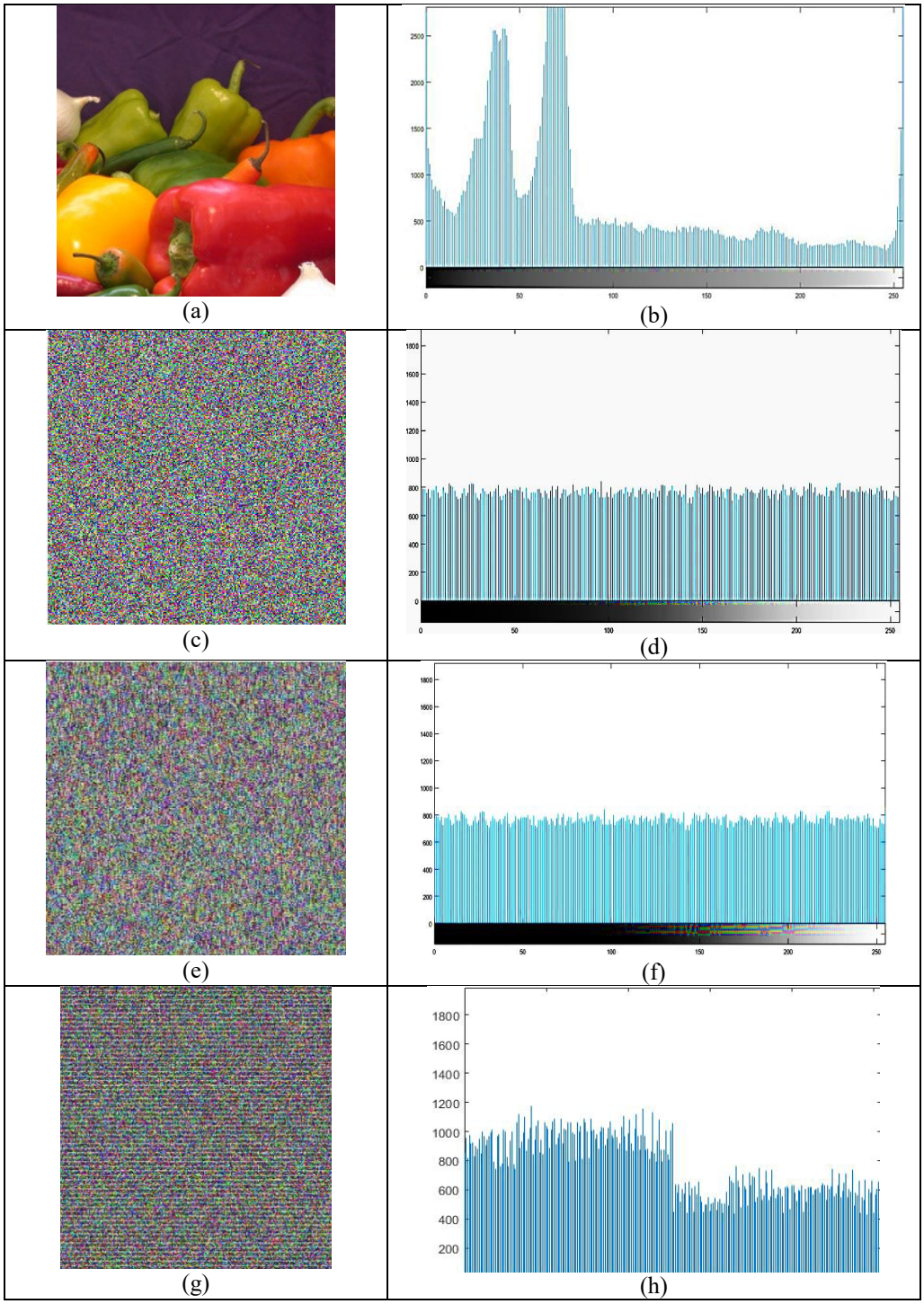


Figure 2. Image encryption and Histogram of the number of occurrences of pixels of an image with the DES Algorithm (a) Test image (b) Histogram of the test image (c) Cipher image of 2(a) using the standard method of DES (d) Histogram of 2(c) using the standard method of DES (e) Cipher image of 2(a) using the proposed XOR vector space (f) Histogram of 2(e) using the proposed XOR vector space (g) Cipher image of 2(a) using the hybrid S-Box (h) Histogram of 2(g) using the hybrid S-Box.

D. Differential Analysis

NPCR and UACI, both cryptoanalysis methods, assess the resistance of encryption methods to differential attacks by changing a single bit of the encryption key. Differential analysis evaluates the resilience of the proposed method [28]. The comparison involves analyzing the cipher image generated with the original key and the resulting image after changing one bit in the key to determine if substantial differences in pixel values occur. The ciphertext images that result are denoted by C_1 and C_2. These resulting ciphertext images are denoted as I_1 and I_2, respectively. The parameters are evaluated as:

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(x, y) \times 100\% \quad (6)$$

Where

$$D(x, y) = \begin{cases} 0, if\ C_1(x, y) = C_2(x, y) \\ 1, if\ C_1(x, y) \neq C_2(x, y) \end{cases}$$

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(x,y) - C_2(x,y)|}{255} \times 100\% \quad (7)$$

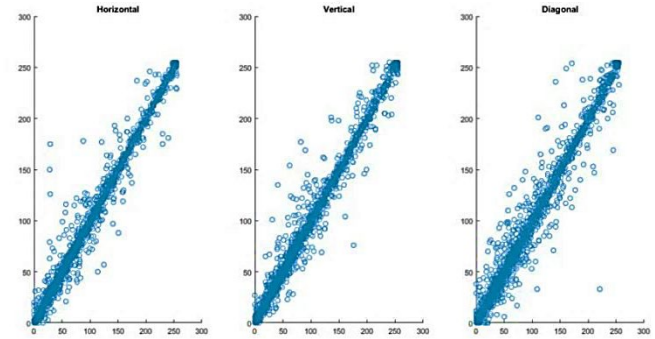
where $C_1(x, y)$ and $C_2(x, y)$ are the pixel values at the point (x,y) in the two ciphertext images respectively. Here, $M \times N$ indicates the dimensions of the image. The ideal values for NPCR and UACI are 99.61% and 33.46%, respectively [29]. Table 4 compares the parameters generated by the proposed technique to those obtained using standard methods that employ a static S-Box. The obtained parameters are also compared with references. It is evident that the proposed method exhibits a strong resistance to differential attacks and is very close to the ideal value.

Table 4. Comparison of the parameters with the standard DES, AES

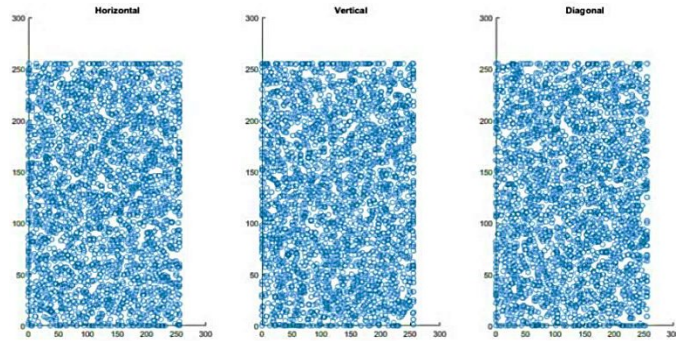
Parameter	DES (Standard S-Box)	DES (XOR S-Box)	DES (Hybrid S-Box)	AES (Standard S-Box)	AES (XOR- S-Box)	AES (Hybrid S-Box)	[17] (Lena)	[32] (Lena)	[33] (crowd)	[34] (Lena)	[35] (Lena)
Entropy	7.947	7.954	7.9664	7.99144	7.99128	7.9384	7.9969	7.9992	7,95667	7.9998	7.9972
NPCR	99.61	99.61	99.673	99.61	99.99	99.45	100	99.56	99,62987	99.623	99.6124
UACI	33.56	33.56	33.63	33.42	34.53	29.3822	33.544	33.48	31,83459	33.47	33.4591

E. Correlation analysis

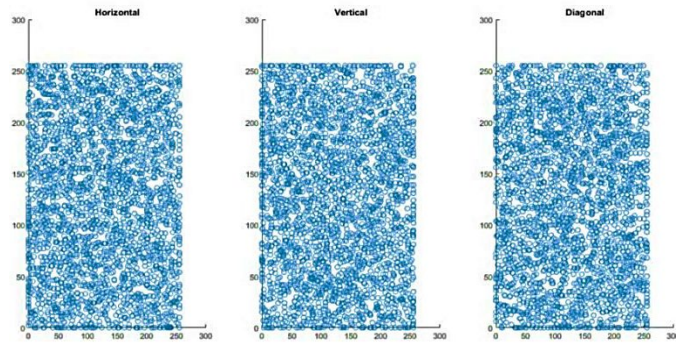
The correlation assesses the degree of relationship between adjacent pixel values within encrypted image. Typically, pixel values in a plaintext image are distributed linearly, whereas those in an encrypted image should exhibit a nonlinear distribution. Figure 3(a) illustrates that the pixel values in the original image are distributed linearly along the horizontal, vertical, and diagonal directions. Figure 3(b) displays the pixel correlation of the standard DES method using a static S-Box, demonstrating complete randomness. Figure 3(c) illustrates the correlation for the proposed method utilizing the XOR vector space, while Figure 3(d) represents the correlation for the hybrid S-Box. The correlation of pixels for the standard AES method using the original S-Box is shown in Figure 4(b). Figure 4(c) shows the correlation using the XOR vector space. The encrypted image displays an exceptionally minimal pixel-to-pixel correlation, showcasing the robust confusion and diffusion properties of the proposed method [36]. Yet, in AES using the hybrid S-Box, the correlation result is less favorable compared to other methods, attributed to repetitions in integers within the S-Box, as demonstrated in Figure 4(d).



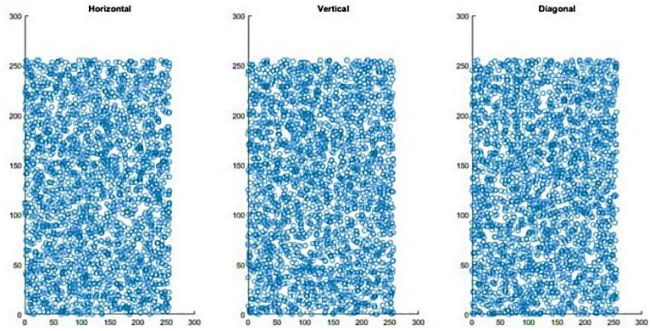
(a)



(b)

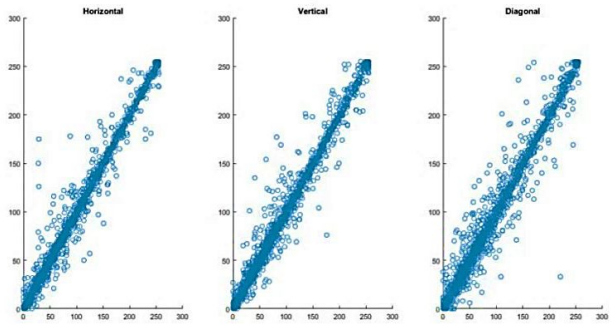


(c)

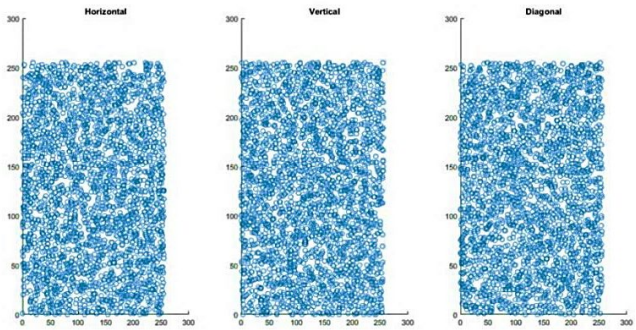


(d)

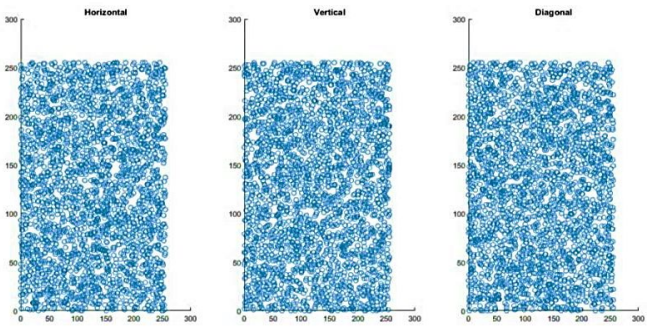
Figure 3. Correlation of pixels in the test image and cipher image using DES (a) correlation in the test image (b) correlation using the original S-Box (c) correlation using the XOR vector space (d) correlation using the Hybrid S-Box.



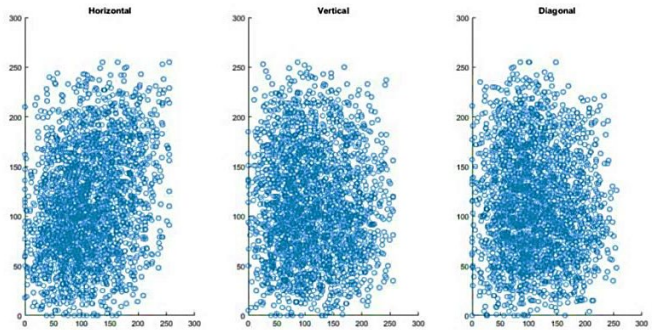
(a)



(b)



(c)



(d)

Figure 4. Correlation of pixels in the test image and cipher image using AES (a) correlation in the original image (b) correlation using the original S-Box (c) correlation using the XOR vector space S-Box (d) correlation using the Hybrid S-Box.

F. MSE and PSNR analysis

MSE (Mean Squared Error) and PSNR (Peak Signal-to-Noise Ratio) serve as metrics to evaluate the quality of both the plaintext and cipher images. PSNR measures the peak error, whereas MSE quantifies the squared differences between the original and cipher images. Greater distortion corresponds to smaller PSNR values. When the Mean Squared Error value is less, the error [30] is minimized. The following equation [31] calculates MSE mathematically:

$$MSE = \sum_{x=1}^{M} \sum_{y=1}^{N} \frac{|P(x,y) - C(x,y)|}{M \times N} \quad (8)$$

and PSNR is calculated by

$$PSNR = 20\, log_{10} \frac{255 \times 255}{\sqrt{MSE}} \text{ dB} \quad (9)$$

Table 5. Comparison of MSE and PSNR

Parameter	DES (Standard S-Box)	DES (XOR S-Box) f	DES (Hybrid S-Box)	AES (Standard S-Box)	AES (XOR-S-Box)	AES (Hybrid S-Box)	[17] (Lena)	[18] (Lena)	[32] (Lena)	[36] (Lena)
MSE	9091.54	8731.92	8965	3568.68	3563.38	3705.72	8952.413	7757.1	7771.88	7835.4
PSNR	8.58	8.75	8.64	11.69	11.67	12.57	8.6114	28.23	8.7793	9.1902

In this context, $P(x, y)$ and $C(x, y)$ represent the pixel values of the plaintext and cipher image at coordinates (x, y) respectively. Table 5 presents a comparison of the PSNR and MSE achieved using the XOR vector space method with those of the standard method and other references.

5. Conclusion

The study presents a new approach for constructing a substitution box (S-Box) using a newly developed XOR vector space generation technique.This approach involves shifting of integers based on a particular rule, leading to an initial time complexity of $O(\log_2(x))$. However, with the execution of suitable hardware, the complexity can be further brought down to $O(1)$. This XOR vector space generation method has been applied to four different cases within the DES and AES cryptographic systems. In each instance, the resultant cipher images exhibit no noticeable traces. The histograms of all encrypted images display uniform scattering, with the exception of the fourth case, which employs a hybrid matrix combining standard and XOR vector spaces for the S-Box in AES. The entropy levels of all methods approach eight, while the NPCR and UACI values are near ideal, indicating robustness against differential attacks. Additionally, the PSNR values are lower compared to existing methods. Overall, the security analysis demonstrates that this innovative approach is competitive with standard and other hybrid methods in ensuring cryptographic security.

6. References

[1] Chen, Zhuozhao, and Guodong Ye. "An asymmetric image encryption scheme based on hash SHA-3, RSA, and compressive sensing." *Optik* 267 (2022): 169676.

[2] Amina, Souyah, and Faraoun Kamel Mohamed. "An efficient and secure chaotic cipher algorithm for image content preservation." *Communications in Nonlinear Science and Numerical Simulation* 60 (2018): 12-32.

[3] Yahi, Amina, et al. "A color image encryption scheme based on 1D cubic map." *Optik* 249 (2022): 168290.

[4] Valandar, Milad Yousefi, Milad Jafari Barani, and Peyman Ayubi. "A fast color image encryption technique based on three-dimensional chaotic maps." *Optik* 193 (2019): 162921.

[5] Paar, Christof, and Jan Pelzl. Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media, 2009.

[6] Knudsen, Lars R., and Matthew Robshaw. The block cipher companion. Springer Science & Business Media, 2011.

[7] Akhshani A, Behnia S, Akhavan A, Abu Hassan H, Hassan Z. A novel scheme for image encryption based on 2D piecewise chaotic maps. Opt. Commun. 2010;283:3259–66.

[8] Manzoor, Atif, Amjad Hussain Zahid, and Malik Tahir Hassan. "A new dynamic substitution box for data security using an innovative chaotic map." *IEEE Access* 10 (2022): 74164-74174.

[9] Liu, Hongjun, et al. "Chaos-based adaptive double-image encryption scheme using a hash function and S-Boxes." *Multimedia Tools and Applications* 77 (2018): 1391-1407.

[10] Çavuşoğlu, Ünal, et al. "A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system." *Nonlinear dynamics* 87 (2017): 1081-1094.

[11] Zahid, Amjad Hussain, et al. "A novel construction of dynamic S-box with high nonlinearity using heuristic evolution." IEEE Access 9 (2021): 67797-67812.

[12] Hematpour, N., Ahadpour, S. Execution examination of chaotic S-Box dependent on improved PSO algorithm. Neural Comput & Applic 33, 5111–5133 (2021). https://doi.org/10.1007/s00521-020-05304-9.

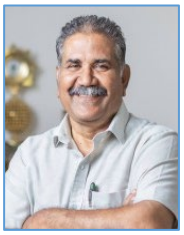
[13] M. Ahmad, E. Al-Solami, A. M. Alghamdi and M. A. Yousaf, "Bijective S-Boxes Method Using Improved Chaotic Map-Based Heuristic Search and Algebraic Group Structures," in IEEE Access, vol. 8, pp. 110397-110411, 2020, doi: 10.1109/ACCESS.2020.3001868.

[14] M. Ş. Açikkapi and F. Özkaynak, "A Method to Determine the Most Suitable Initial Conditions of Chaotic Map in Statistical Randomness Applications," in IEEE Access, vol. 9, pp. 1482-1494, 2021, doi: 10.1109/ACCESS.2020.3046470.
[15] Cusick, Thomas W., and Pantelimon Stanica. “Cryptographic Boolean functions and applications”. Academic Press, 2017.
[16] Malik, Muhammad Sarmad Mahmood, et al. "Generation of highly nonlinear and dynamic AES substitution-boxes (S-Boxes) using chaos-based rotational matrices." *IEEE Access* 8 (2020): 35682-35695.
[17] Maolood, Abeer Tariq, et al. "Fast Novel Efficient S-Boxes with Expanded DNA Codes." *Security and Communication Networks* 2023 (2023).
[18] Zahid, A. H., Iliyasu, A. M., Ahmad, M., Shaban, M. M. U., Arshad, M. J., Alhadawi, H. S., & Abd El-Latif, A. A. (2021). A novel construction of dynamic S-Box with high nonlinearity using heuristic evolution. *IEEE Access*, *9*, 67797-67812.
[19] Zahid, A. H., & Arshad, M. J. (2019). An innovative design of substitution-boxes using cubic polynomial mapping. *Symmetry*, *11*(3), 437.
[20] Zahid, A. H., Al-Solami, E., & Ahmad, M. (2020). A novel modular approach based substitution-box design for image encryption. *IEEE Access*, *8*, 150326-150340.
[21] Dodmane, R., Aithal, G., & Shetty, S. (2022). Construction of vector space and its application to facilitate bitwise XOR–Free operation to minimize the time complexity. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 9836-9843.
[22] Dodmane, R., Aithal, G., & Shetty, S. (2019). Time Complexity Reduction for the Application of Stream Cipher System Based XOR Free Operation. *International Journal of Recent Technology and Engineering*, *8*(3), 5402-5408.
[23] *Data Encryption Standard*, Federal Information Processing Standards Publication (FIPS PUB) 46, National Bureau of Standards, Washington, DC (1977).
[24] Rijmen, Vincent, and Joan Daemen. "Advanced encryption standard." Proceedings of federal information processing standards publications, national institute of standards and technology 19 (2001): 22.
[25] K. Mandal, C. Parakash and A. Tiwari, "Performance evaluation of cryptographic algorithms: DES and AES," 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, India, 2012, pp. 1-5, doi: 10.1109/SCEECS.2012.6184991.
[26] Shannon, C.E. Communication theory of secrecy systems. Bell Syst. Tech. J. 1949, 28, 656–715. [CrossRef].
[27] Artuğer, Fırat, and Fatih Özkaynak. "A method for generation of substitution box based on random selection." Egyptian Informatics Journal 23.1 (2022): 127-135.
[28] Zhang, X., Zhou, Z., Niu, Y., 2018. An image encryption method based on the Feistel network and dynamic DNA encoding. IEEE Photonics J. 10 (4), 1–14.
[29] Zhu, Congxu, Guojun Wang, and Kehui Sun. "Cryptanalysis and improvement on an image encryption algorithm design using a novel chaos based S-box." Symmetry 10, no. 9 (2018): 399.
[30] Zakaria, Abdul Alif, Mehdi Hussain, Ainuddin Wahid Abdul Wahab, Mohd Yamani Idna Idris, Norli Anida Abdullah, and Ki-Hyun Jung. "High-capacity image steganography with minimum modified bits based on data mapping and LSB substitution." Applied Sciences 8, no. 11 (2018): 2199.
[31] Man, Zhenlong, et al. "Double image encryption algorithm based on neural network and chaos." Chaos, solitons & fractals 152 (2021): 111318.
[32] Zhang, Ying-Qian, Jun-Ling Hao, and Xing-Yuan Wang. "An efficient image encryption scheme based on S-boxes and fractional-order differential logistic map." IEEE Access 8 (2020): 54175-54188.
[33] Çavuşoğlu, Ünal, et al. "Secure image encryption algorithm design using a novel chaos based S-Box." Chaos, Solitons & Fractals 95 (2017): 92-101.

[34] Aslam, Mazzamal, et al. "A strong construction of S-box using Mandelbrot set an image encryption scheme." PeerJ Computer Science 8 (2022): e892.
[35] Khan, Majid, Tariq Shah, and Syeda Iram Batool. "Construction of S-box based on chaotic Boolean functions and its application in image encryption." Neural Computing and Applications 27 (2016): 677-685.
[36] Man, Zhenlong, et al. "Double image encryption algorithm based on neural network and chaos." Chaos, solitons & fractals 152 (2021): 111318.

Arun Upadhyaya received the AMIE degree in E&C Engineering from Institution of Engineers (Kolkata), in the year 2008 and Masters in Digital Electronics and Communication from NMAMIT, Nitte, in the year 2013. He is working as Assistant professor (Selection Grade) in the department of Electronics and Communication Engineering in SMVITM, Bantakal, Udupi. He is currently pursuing the Ph.D. degree at Visvesvaraya Technological University (VTU) in Karnataka, India.

Shantharama Rai C completed his Engineering (B.E) from Mangalore University, M.Tech from N.I.T.K. Surathkal and PhD from V.T.U Belgaum with specialization of power electronics and control system. He has served in different institutions in various capacities for more than 24 years. He is presently working as the Principal at AJEIT, Managluru. He has published /presented more than 50 papers in international/national journals /conferences in the field of engineering. Presently he is guiding 3 PhD Students and 5 have completed their research under his guidence.



Ganesh Aithal completed his Engineering (B.E) from Mysore University, M.Tech from Coachin University of Science and Technology, Coachin and PhD from N.I.T.K. Surathkal with specialization of cryptography. He has served in different institutions in various capacities for more than 35 years. He is presently working as the Vice Principal at SMVITM, Bantakal, Udupi. He has published/presented more than 35 papers in international/national journals /conferences in the field of engineering. Dr. Aithal has successfully guided 5 research scholars.