IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 6, 2021, accepted January 12, 2021, date of publication January 18, 2021, date of current version January 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052161

New DNA Coded Fuzzy Based (DNAFZ) S-Boxes: Application to Robust Image Encryption Using Hyper Chaotic Maps

AMIRA G. MOHAMED[1], (Student Member, IEEE), NOHA O. KORANY[2], AND SAID E. EL-KHAMY[2], (Life Fellow, IEEE)

[1]Department of Electrical Engineering, Alexandria Higher Institute of Engineering and Technology (AIET), Alexandria 21500, Egypt
[2]Department of Electrical Engineering, Faculty of Engineering, Alexandria University, Alexandria 21500, Egypt

Corresponding author: Amira G. Mohamed (eng_amira90@yahoo.com)

ABSTRACT This article proposes a novel approach of improvising the cryptographic features of substitution-boxes (S-Box) based on the Choquet Fuzzy Integral (CFI) and DNA techniques. First, we propose a strong structure for the construction of four S-Boxes using CFI. The key for generating the CFI based (FZ) S-Boxes consists of two parts, namely, an external secret key and a secret image. Each of these FZ S-boxes is then encoded using DNA techniques, with dynamic rules selection which is dictated by a secret control code. The resultant four S-boxes are designated as DNAFZ S-Boxes. To apply for image encryption, the plain image is, at first 8-bit binary-coded, shuffled by an M-sequence, and down-sampled into four sub-images. Subsequently, the pixel values of each sub-image are replaced with the corresponding values of one of the four DNAFZ S-Boxes. Next, each DNAFZ encoded sub-image is diffused with a different DNA encoded chaotic sequence from Chen's hyper-chaotic map. Finally, the four DNAFZ/Chaotic encoded sub-images are combined to build the final encrypted image. The proposed DNAFZ S-boxes shows excellent statistical properties under majority logic criterions such as correlation, homogeneity, energy, entropy, and contrast. Moreover, numerical simulation is used to examine the efficacy of encrypted images against different attacks. In particular, the values of the pixel correlation coefficient are found to be quite small either horizontally, vertically, or diagonally (between 7.8597e-04 and 0.00527, between 8.7856e-04 and 0.00452, and between 0.00241 and 0.00021, respectively). In addition, the information entropy of the encrypted image is found to be within the range of (7.9965:7.9989) which is very near to the ideal value of 8. As for the UACI and the NPCR, they are in the ranges between 33.46 and 33.32 and between 99.58 and 99.62, respectively. These values are also very close to the optimum ones. The results are compared to those of other encryption algorithms and proved that the proposed encryption method delivers better results than other conventional ones including LSS chaotic map, Arnold transforms, Dynamic Henon map, Hybrid chaotic map optimized substitution, and cubic S-Box.

INDEX TERMS Image encryption, Choquet fuzzy integral, S-Box, DNA technique, hyper chaotic map, DNAFZ S-boxes.

I. INTRODUCTION

A. BACKGROUND

The current availability of various mobile applications and related technologies have made the different types of multimedia files important. The different public channels used to send/receive these files from one person/entity to the next do not provide enough security for the transfer of such files and especially the digital images associated with them. This insecurity has been increasing due to vast leaps in technology over the past few years. Data encryption is used as a method to secure data and cryptography is thus an important technology applied in addressing the security of information. There are different types of algorithms used in cryptographic systems. Among them are AES, which stands for Advanced Encryption Standard and DES or Data Encryption Standard.

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Imran Tariq.

However, images have been difficult to encrypt using such methods, as they contain large amounts of information. This has been the reason behind trials to develop other systems of encryption for specific application on images. These systems include methods that apply fuzzy integral, DNA techniques, and the chaos theory [1]–[3].

DNA cryptography has evolved as a result of the similarities found between biology and computer science. It has been used as a system of data security. DNA or deoxyribose nucleic acid is the material inside every living cell. It holds all the genetic features, carrying them from one generation to the next, through hereditation. DNA is made of four nitrogenous bases, cytosine, thymine, adenine, and guanine. The fact that they can store massive amounts of data and the ideas of complementation and parallelism play an important role in their ability to secure data. DNA-based encryption of photos involves the following steps: using a DNA-encoding rule to encode the image into a DNA matrix, followed by conversions between the four DNA nitrogenous bases in what is known as DNA level permutation and/or DNA level diffusion. This step takes place on the DNA matrix. The final step involves decoding the DNA matrix to obtain the cipher image [4]–[6].

Fuzzy set theory involves various paradigms used for fuzzy neural computations, which are quite interesting. Regarding decision analysis, several operators are employed for aggregation. For X, a finite set, a fuzzy measure is required to base a fuzzy integral upon. This is done as a computational scheme, when dealing with fuzzy sets that have dependent input, in order to nonlinearly integrate all values from individual subsets. This required fuzzy rules to be set in place, where fuzzy measures are the basis for the consequent. These rules gave way to a defuzzified output, which was also a Choquet fuzzy integral (CFI) [7]–[9]. CFI, with its set of benefits, added great value to the area of image encryption. These benefits include that it is easy to implement, simple to compute, effective in reaching high level security, high in speed and sensitivity [10].

Several features characterize chaotic systems. These include, but are not limited to, periodicity, determinacy, and the fact that they are highly sensitive to initial conditions. Chaotic maps produce chaotic sequences, which are pseudo-random, complex in structure and not easy to analyze and predict. Despite the fact that their security is not very high, these sequences possess large key spaces that are very sensitive to any changes that occur in their key. This is the reason for the development of hyper-chaotic systems for use in algorithms that deal with image encryption. To create a four-dimensional hyper-chaotic system, an extra dimension is added. This takes place in accordance with Chen's chaos theory. In recent years, the characteristics of hyper-chaotic systems have been improved by adding several positive Lyapunov exponents, complex and dynamical characteristics, as well as improved sensitivities and bigger key spaces [11]–[13].

To make sure that block ciphers continuously possess confusion properties, S-boxes (substitution boxes) were used in the structure of permutation ciphers to play the role of essential nonlinear elements. Linear and differential cryptanalysis represent different types of attacks that a strong block cipher must be able to resist. Majority logic criteria should be satisfied by S-boxes, in the case of a strong encryption system. There is an availability of choosing this S-box under the control of a key, when encrypting, instead of having a static entry. Since there are certain standards applied in encryption applications, there have been different types of S-boxes suggested for them. This includes random key-dependent, as well as bijective S-boxes. These S-boxes are usually devised depending on different techniques, some of them are related to chaotic maps, while others on algebraic structures [14]–[19].

B. RELATED WORK

It has always been the goal of designers of cryptosystems to make S-boxes that can perform robustly on a cryptographic level. Various methods of cryptographic construction for S-boxes have been suggested by researchers. In [1], a new method using Hilbert fractals was demonstrated. The idea in this article was based on hyper-chaotic systems with fractal keys, as well as S-boxes. A new S-box was constructed using an LSS system (Logistic-Sine System), which is a discrete compound chaotic system proposed in [2]. Then, a single S-box and chaos were used to design an image encryption algorithm. Reference [4] introduced a new encryption approach based on a chaotic system that used an S-box depending on the DNA system. The first step was coding the image into a DNA sequence, followed by the substitution of this sequence using a DNA based S-box. To produce the last cipher, a chaotic sequence was applied to permute the rows and columns. This sequence was generated through the application of a 2D logistic map. To generate an S-box, [16] suggested two methods, an O-shaped path scrambling algorithm, namely the FLSOP and a fractional Lorenz-Duffing chaotic system in 6D. Another research, [17], suggested a different method to produce an S-box, that is on the basis of coset graphs of symmetric groups. They were tested for their suitability to encrypt not only images, but also other multimedia files. Research demonstrated a method to encrypt photos on the basis of dynamic S-boxes, built according to a chaotic system, whose initial states and parameters were produced using an external 256-bit key, as well as the last pixel of the image before processing, to produce the S-box [20]. Researchers proposed a technique to encrypt images that depends on a spatiotemporal chaotic system [21]. In said research, S-boxes were constructed based on random numbers produced from a regular and a piecewise-linear chaotic map. To encrypt images, Arnold transform and S-box scrambling were used for quantum image encryption [22]. Gray-level encryption was carried out using S-box substitution that entailed quantum chaos sequence. This step was performed prior to the permutation of the Arnold transform. An S-box was constructed in [23] based on a chaotic sine map. Afterwards, a cryptosystem was used, applying the 2D discrete domain Hénon map.

In [24], the construction of encrypted image was depended on the high-dimension Lorenz chaotic system and perceptron model within a neural network. Novel confusion and diffusion method for image encryption were proposed in [25]. Arrays produced by PWLCM, Chevy-chevy, and DNA were responsible for generation cipher image. In [26],spatial bit-level permutation and high-dimension chaotic system were applied for producing image encryption. Initially, the binary bits of image is permuted by the scrambling mapping generated by piecewise linear chaotic map (PWLCM). Afterthat, Chen system is utilized to shuffle and diffuse the red, green and blue components simultaneously. A stream-cipher algorithm based on one-time keys and robust chaotic maps were designed to encrypt color image in [27].

C. OUR CONTRIBUTION

To overcome these security shortcomings and design a secure and effective image cryptosystem, this article proposes a new image encryption algorithm based on four S-Boxes generated with a Choquet fuzzy integral (CFI) and DNA technique. The new S-boxes, named DNAFZ S-boxes, are introduced for the first time in this article. In addition, the generation of four S-Boxes depends on the secret image and external key to increase the security levels and randomness. It is tested by applying NIST test analysis. It achieves the best results when compared to other algorithms. Furthermore, an image encryption scheme depends on dynamic features of hyper chaotic systems. The new DNAFZ/chaotic encryption algorithm proposed in this article is illustrated in Figure 1 and can be summarized as follows:

- Generation of DNAFZ S-Boxes:

1) Choquet fuzzy integral (CFI) is applied for generating four random sequences. As explained in our previous work in [28], [29], the initial conditions (key) for the generation of these CFI pseudorandom sequences depends on two components. One of these components is a secret sequence (key) of finite length, while the other component is a secret image. Both components are diffused together to generate the primary conditions and parameters of the CFI.
2) The four fuzzy based S-Boxes (FZ S-boxes) are created by resizing each of the four CFI random sequences as a matrix of size 16×16. Hence, each S-Box is on the form of a matrix consisting of 256 integers of in the range of [0, 255], with each integer represented by 8 bits.
3) DNA's complementary rule is used to encode the four FZ S- boxes. The selection of DNA rule of each FZ S-box is made through a simple control code. The generated DNA-Fuzzy based S-boxes are designated as DNAFZ S-boxes.

- Application of the DNAFZ S-Boxes to image encryption based on Chen's hyper chaotic map:

1) The pixels of the original image are 8-bits encoded and reshaped into a one-dimensional binary sequence. Then, a secret M-sequence is used to shuffle the binary stream.
2) The shuffled image binary stream is down sampled to extract four sub-images.
3) The pixel values of each sub-image are changed by one of the four DNAFZ S-boxes to generate four DNAFZ encoded sub-images.
4) Chen's hyper-chaotic map is applied to generate different random sequences. After that, these sequences are converted into DNA technique. The selection of DNA rule of each chaotic sequence is made according to the remaining four rules.
5) The DNAFZ encoded sub-images are diffused with different hyper chaotic/DNA random sequences, using a control code, to generate four DNAFZ/chaotic sub-images.
6) The four DNAFZ/chaotic sub-images are finally combined to reconstruct the cipher image.

The proposed algorithm has been proven effective and secure through tests and security analyses. The results are compared to those of other encryption algorithms and proved that the proposed new algorithm delivers better results than other recent techniques.

D. STRUCTURE OF ARTICLE

As for the remainder of this article, it is divided as follows: in Section II, DNA sequences are discussed; in Section III, the Choquet fuzzy integral (CFI) is briefly explained. Section IV contains an introduction to hyper-chaotic maps. Section V proposes the algorithm of the S-box used in image encryption. Section VI explains the results of the experiment. As for the evaluation analysis of S-Box, it is discussed in Section VII. The analysis of the security of the proposed image encryption method is discussed in Section VIII. As for Section IX, it holds the conclusion of this article.

TABLE 1. The encoding rules for DNA sequences.

Bases	Rule 0	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
A	00	00	01	10	10	01	11	11
G	01	10	00	00	11	11	01	10
C	10	01	11	11	00	00	10	01
T	11	11	10	01	01	10	00	00

II. DNA SEQUENCES

Among the basic issues in biology are deoxyribonucleic acid and DNA sequences. This sequencing has been extensively used in the application in various other fields. The DNA sequence includes four basic nucleic acids, adenine, thymine, cytosine, and guanine, or A, T, C, and G, respectively. Bases are paired according to the following rule: the purine A is always paired with the pyrimidine T and the pyrimidine C is always paired with the purine G. On the other hand, according to the binary system, 0 and 1 complement one another. Thus, complementary pairs include 00 and 11, as well as 01 and 10. In Table 1, rules for coding and decoding DNA bases are shown. An example can be given for better understanding,

TABLE 2. DNA XOR operations using Rule 2.

	A	C	G	T
A	A	C	G	T
C	C	A	T	G
G	G	T	A	C
T	T	G	C	A

where 185 is the gray level of a pixel. It can be converted into a DNA sequence, according to the binary format, which is (10111001). According to Table 1, DNA sequence rules are Rule1 (GTGC), Rule2 (CTCG), etc. In recent years, DNA computing has garnered a lot of attention. In this field, some scientists suggested applying Boolean operators, while applying DNA as a form of computation [30]–[32]. The exclusive OR (XOR) operator for each of the four bases of a DNA sequence is listed in Table 2.

III. CHOQUET FUZZY INTEGRALS

A fuzzy integral is used to aggregate information; it is a nonlinear function that depends on the fuzzy measure that gives an alternative computational algorithm. In [10], [28], [29], [33], [34], the features of CFI are described.

A. FUZZY MEASURE

Fuzzy measures have different classes including necessity, belief, probability, which is a subset of classical measures, possibility, and plausibility measures. The fuzzy measure can be determined through the methods seen in the following equations:

$$F(A_1) = g_1 \tag{1}$$

$$F(A_i) = g_i + F(A_{i-1}) + \lambda\, g_i\, F(A_{i-1}),\ 1 < i < n \tag{2}$$

where g_i represents the membership grade. As for $F(A_i)$, it marks the fuzzy measure over the corresponding membership grades.

B. SUGENO λ MEASURE

It is a measure close in similarity to the probability measure. However, it is specific, as it handles the state of probability representing the state $g(A \cup B)$. As for λ, it shows the variance between the Sugeno measure and the state mentioned above. Eq. 3 can be used to calculate the real-valued parameter, $\lambda \in [-1, \infty]$:

$$1 + \lambda = \prod_{i=1}^{n} (1 + \lambda\, g_i) \tag{3}$$

C. CHOQUET FUZZY INTEGRAL

In an effort to calculate the CFI, the Sugeno λ and fuzzy measures are used. However, CFI is a nonlinear figure that is more complex than both previously mentioned measures and can be calculated using the following equation:

$$CFI = \int h\, dg = \sum_{i=1}^{n} [h(x_i) - h(x_{i-1})]\, F(A_i) \tag{4}$$

where h is the initial input of the Choquet fuzzy integral.

IV. CHEN's HYPER CHAOTIC MAPS

Chen's hyper-chaotic system is described as following equations:

$$\dot{x} = a\,(y - x) \tag{5}$$

$$\dot{y} = -xz + d\,x + c\,y - q \tag{6}$$

$$\dot{z} = xy - b\,z \tag{7}$$

$$\dot{q} = x + k \tag{8}$$

As can be seen in above equation, a, b, c, d, k are the system parameters. In the event where a = 36, b = 3, c = 28, d = 16 and $-0.7 \leq k \leq 0.7$, then chaotic state is achieved for Chen's hyper-chaotic system. In this case, the system can produce four chaotic sequences. For the purpose of this research and its suggested encryption scheme, Chen's hyper-chaotic system is used to produce hyper-chaotic sequences used in image diffusion [35], [36].

V. PROPOSED IMAGE ENCRYPTION ALGORITHM BASED ON THE DNAFZ S-BOX

This study proposes a new scheme for image encryption based on the S-box system that depends on the characteristics of hyper-chaotic maps, fuzzy integral, DNA techniques, and the above-mentioned analysis. This new system is highly efficient, has a more complex structure, improved security, and stronger resistance to attacks than other algorithms. This new scheme is developed according to the following steps: Step one, generating the CFI random sequences, as well as the S-box. In this article, they will be referred to by FZ S-boxes. Step two involves using the DNA technique to code the previously described FZ S-boxes. Following this step, the S-boxes are referred to as DNAFZ S-boxes. Step number three is using downsampling to extract four sub-images from the original image, for which the binary bit of each pixel is modified through the use of the M sequence. Step four focuses on making each of the four sub-images more complex, by replacing each pixel's value with the value of one of the four DNAFZ S-boxes. Step five involves the diffusion of the coded sub-images. This takes place by using Chen's hyper-chaotic map, to extract different DNA random sequences and through the application of a control code. The final encrypted image is obtained through the combination of the four sub-images, which were encoded through the DNAFZ/chaotic systems. The proposed algorithm in its block diagram form can be seen in Figure 1. Moreover, Figure 2 explains the algorithm with a 4 × 4 example of image data. The steps to build the encryption technique are as follows:

A. DESIGN OF THE CFI-BASED FOUR S-BOXES GENERATION ALGORITHM (FZ- S Boxes)

In this article, CFI is used as a basis for the generation of four random sequences that are highly secure [28], [29]. To calculate the CFI, an external key and a secret image are needed. Both are used to calculate (h_1, h_2, h_3, h_4), namely, the initial input. Since a key is used to select the secret image, this makes this image variable. To extract the secret image

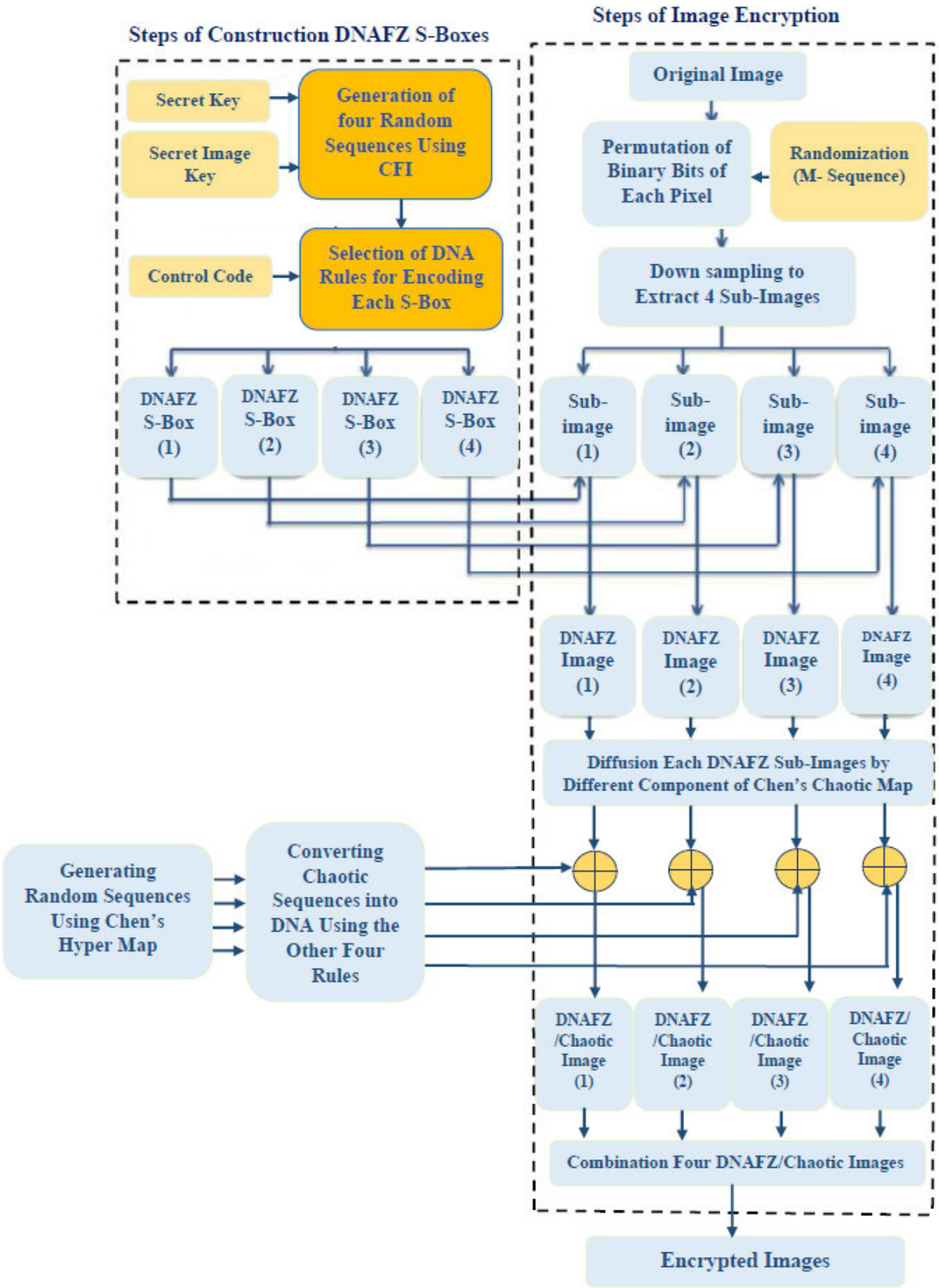


FIGURE 1. Block diagram of the proposed algorithm.

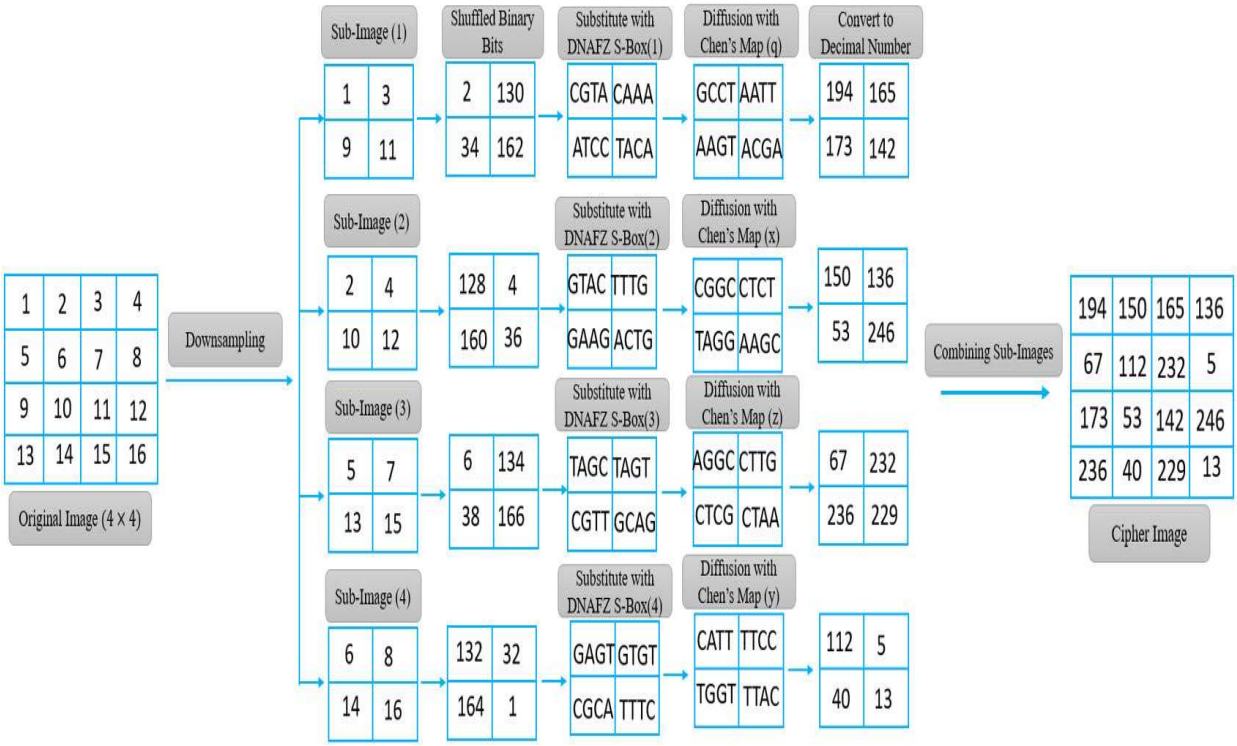


FIGURE 2. A numerical example of the proposed algorithm.

from a group of images, a 6-bit key is used. By applying these parameters, the security level of the process is increased. Figure 3 shows the generation of four S-Boxes by the CFI, along with a brief explanation of how they were generated (Algorithm 1).

Step 1:

Blocks of 8 bits (K_1 K_2,……,K_{16}) comprise the 128-bit secret key. Then, the following equation is used to calculate the key parameters:

$$A = (K_1 \oplus K_2 \oplus K_3 \oplus K_4) \tag{9}$$

$$B = (K_5 \oplus K_6 \oplus K_7 \oplus K_8) \tag{10}$$

$$C = (K_9 \oplus K_{10} \oplus K_{11} \oplus K_{12}) \tag{11}$$

$$D = (K_{13} \oplus K_{14} \oplus K_{15} \oplus K_{16}) \tag{12}$$

$$K = \sum_{i=1}^{i=16} (K_i) \tag{13}$$

Step 2:

The gray-level matrix of a 256 × 256 image is inserted. Dividing this image into 2 × 2 blocks results in four values of $I(s)$, using XOR for each of the gray levels within each block. The equations below provide the initial inputs (h_1, h_2, h_3, h_4):

$$h_1 = ((A + K)\ mod\ 256) \oplus I(1) \tag{14}$$

$$h_2 = ((B + K)\ mod\ 256) \oplus I(2) \tag{15}$$

$$h_3 = ((C + K)\ mod\ 256) \oplus I(3) \tag{16}$$

$$h_4 = ((D + K)\ mod\ 256) \oplus I(4) \tag{17}$$

where $I(1)$,$I(2)$,$I(3)$,and $I(4)$ are each block's diffused values, while employing XOR and K is the summation of blocks for the secret key.

Step 3:

This step involves calculating the initial input (h) from the image, as well as the key. This (h) is then used to calculate membership grades, by applying Eq. 18. Afterwards, Eq. 3 is used to calculate λ. The value obtained is then inserted in Eq. 2 to reach the fuzzy measure.

$$g_i = \frac{1}{1 + h_i} \tag{18}$$

Step 4:

To calculate the CFI, Eq. 4 is applied using both the initial inputs (h_1, h_2, h_3, h_4) and the fuzzy measure $F(A_i)$. The following equation shows the secure PN sequence obtained from the CFI:

$$C_j = (ARS(int\ (CFI\ mod\ 1) \times 10^{14}, S))\ mod\ 256 \tag{19}$$

In Eq.19, the arithmetic right shift of the binary sequence is represented by *ARS*. CFI mod 1 represents the normalized fraction value. Four different random sequences (C_1, C_2, C_3, C_4) can be obtained as S takes up values from 0 to 7.

Step 5:

Resizing the four random sequences obtained from the CFI matrix results in four FZ S-boxes. Each FZ S-box is 16×16 in size. Each consists of 256 integers of 8 bits in the range of [0, 255].

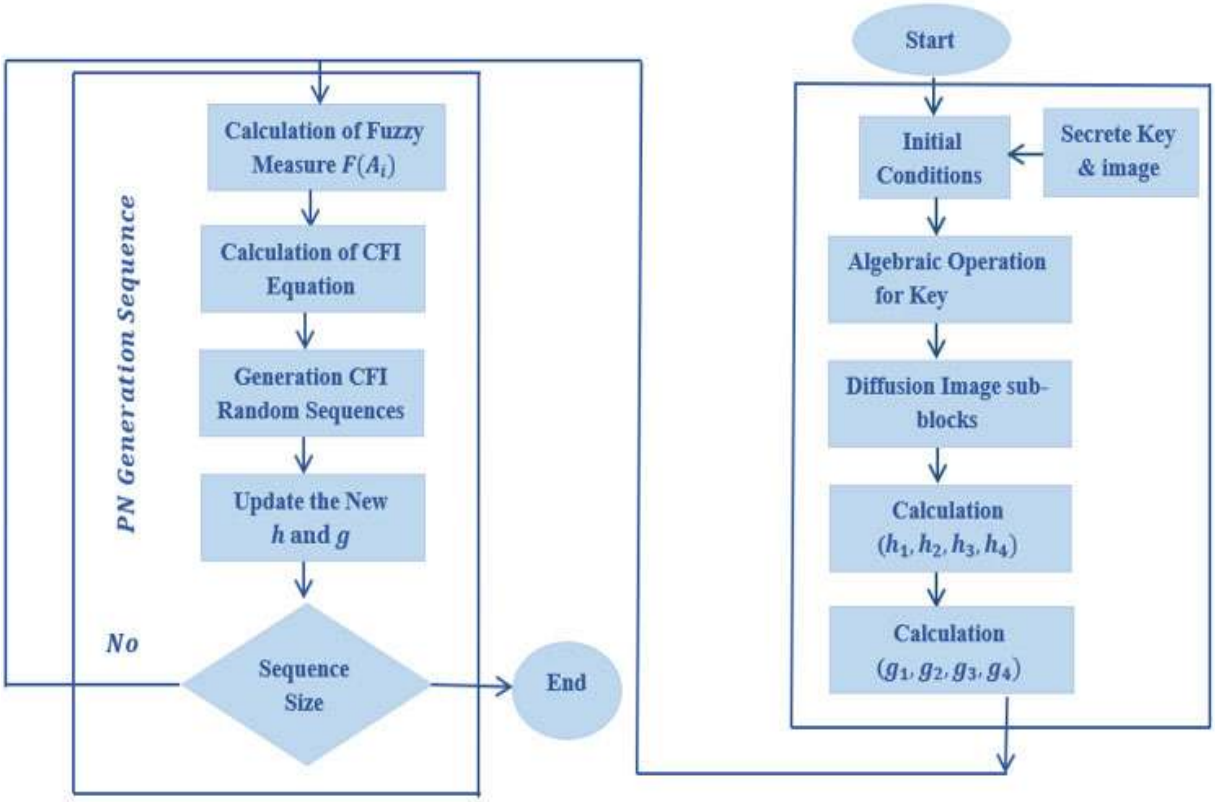


FIGURE 3. Four S-boxes generation.

Algorithm 1 Sequence Generation

Input: External Key with size 128-bit (K), Secret Image $I(M \times N)$
Output: Random sequences C_j
1: Divide K into four blocks (K_1, K_2,.....,K_{16})
2: Calculate key parameters (A,B, C, D)
3: # *Eq*. (9, 10, 11, 12)
4: Divide secret image into 2 × 2 blocks
5: XOR Operation for each of the gray levels within each block
6: Calculate initial inputs (h_1, h_2, h_3, h_4)
7: # *Eq*. (14, 15, 16, 17)
8: Calculate Membership Grades (g_1, g_2, g_3, g_4)
9:# *Eq*. (18)
10: Calculate λ based on *Eq*. (3)
11: Calculate fuzzy measures ($F(A_i)$)
12: # *Eq*. (2)
13: Use *Eq*. (4) to calculate *CFI*
14: Generate random sequences C_j based on *Eq*. (19)
15: **return** C_j

B. CONVERTING THE FOUR S-BOXES INTO DNA (DNAFZ S-BOXES)

- For each of the FZ S-boxes, every element is transformed into an 8-bit binary equivalent, which is then converted into DNA nitrogenous bases, but not using the same rule.

Algorithm 2 DNAFZ-S Boxes Generation

Input: Four Random Sequences (C_1, C_2,C_3,C_4)
Output: DNAFZ-S Boxes
1:Reshape random sequences into matrix with size 16 × 16 (FZ S-Box).
2: Convert each FZ S-Box into 8-bits binary
3: Generate M sequence (control code) to select used 4 bases of DNA
4: Encode each FZ S-Boxes into DNA
5: **return** DNAFZ S-Boxes

- As per Table 1, the DNA coding technique has eight bases; the four S-boxes are encoded using four of these bases, via the simple M Sequence (Control Code).
- For example, rule (1) is used to encode FZ S-box (1), rule (7) is used to encode FZ S-box (2), etc. Figure 4 shows the output of the four different DNAFZ S-boxes.

C. IMAGE SUBSTITUTION BASED ON THE FOUR DNA S-BOXES

The first step involves inserting the original image (size 256 × 256). This is followed by an extraction of the four sub-images (size 64 × 64) from the original image, through downsampling. During the step to follow, the values of the four DNAFZ S-boxes replace the pixels of each sub-image. As an example, DNAFZ S-box (1) produces the pixel value of

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	'CATT'	'TCCT'	'CGTA'	'CAAG'	'CCTT'	'TCCG'	'CGCG'	'TTAA'	'GTGG'	'CATA'	'TACA'	'TTTA'	'TGTC'	'GTGT'	'GTGG'	'TATA'
1	'CCAG'	'GGGT'	'AGAC'	'TTAC'	'TTCC'	'GTCG'	'AACT'	'GTCG'	'TCAT'	'CTTT'	'AGAA'	'ATAC'	'GACA'	'CATT'	'CGGC'	'CAAA'
2	'CAGG'	'GCCC'	'ATCC'	'TACC'	'ACTG'	'GGGA'	'AAAC'	'CAAG'	'TAAT'	'CGTT'	'AGTT'	'CCAG'	'GATT'	'AACT'	'TCTG'	'TAGT'
3	'GAGT'	'GGGT'	'TCAT'	'GACG'	'TGTA'	'TATT'	'TCAG'	'TCTA'	'CCGT'	'TTCT'	'GCAG'	'CAAC'	'GTAC'	'GTCG'	'TTGT'	'AGGA'
4	'CTAG'	'CTTC'	'GCTC'	'GGTG'	'GCGC'	'GGTC'	'GAGT'	'GCTG'	'AACC'	'CATA'	'AAGG'	'TTTG'	'TGGT'	'TATC'	'TCGG'	'ATGG'
5	'TGTC'	'TAGC'	'AATT'	'GGGT'	'CCGT'	'TCTC'	'GTTG'	'GTAT'	'ACTC'	'CTGA'	'CATC'	'ACTT'	'ATGT'	'CTTA'	'GACA'	'GCTT'
6	'AGGT'	'AATG'	'GACT'	'TTGG'	'TCAC'	'AGTA'	'AGCT'	'TTAC'	'AGCC'	'TCCG'	'TATT'	'GGCT'	'TGGA'	'CCGA'	'TGCG'	'TGAC'
7	'AAGA'	'CATC'	'ATAG'	'CCAT'	'ATTG'	'CTGC'	'GAGC'	'GTTC'	'GAAT'	'CGCT'	'GCCC'	'TCAG'	'TTAC'	'TGAC'	'CCGG'	'ACCC'
8	'CGAA'	'CGCT'	'CAAA'	'AGTT'	'CCTA'	'TGTG'	'CAGG'	'TTCG'	'CGTA'	'TCGG'	'TTGT'	'ATAA'	'ACCT'	'CATC'	'GCAA'	'GCCG'
9	'TCCA'	'GGAG'	'CTCC'	'CATT'	'TCGC'	'ACTT'	'CACA'	'TCAG'	'TACG'	'GATC'	'AGGC'	'CTAC'	'GGTC'	'TTGT'	'TTGA'	'GGAC'
10	'GGTA'	'CTGG'	'TACA'	'ACCT'	'GGGC'	'ACCC'	'TCAG'	'GGAT'	'TTTT'	'GACC'	'AAGC'	'CCTC'	'TACC'	'CAAC'	'ACTT'	'CCCA'
11	'GATG'	'GTTG'	'TGCG'	'CTGG'	'GGTC'	'CGTT'	'CTAA'	'GCGT'	'TTTG'	'TAGC'	'GATT'	'GGAG'	'TCAT'	'CAAT'	'TGCC'	'GGGC'
12	'TCCT'	'AAAA'	'TTTT'	'TGTA'	'GCCA'	'TCCA'	'CACA'	'TCGC'	'TCCT'	'CGTT'	'GCGA'	'GTTG'	'GTAC'	'TACA'	'CTCA'	'GAGC'
13	'CACA'	'GGAA'	'GATA'	'TACC'	'TTTG'	'TTAA'	'TAGC'	'TAAC'	'AAAA'	'CGCC'	'CATC'	'CCGC'	'CAGT'	'CTCG'	'TATT'	'GACA'
14	'TACG'	'GGTG'	'TCTT'	'CTGC'	'GGCG'	'TGGG'	'ACCA'	'GCAC'	'GTCG'	'AACG'	'CGCA'	'AAAC'	'AAGT'	'ACCT'	'CGAC'	'AAAG'
15	'GGCC'	'GGGT'	'CCAC'	'TCGT'	'AACA'	'ATAC'	'TACC'	'GCGT'	'TCGA'	'CCAG'	'CTCA'	'CATA'	'TTTT'	'TAAA'	'ATTG'	'ATGT'

(a) The Output of DNA S-Box (1)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	'GTCG'	'TGTT'	'GTAG'	'ATCC'	'TTTG'	'CGTT'	'GTAT'	'AGGC'	'TAGA'	'TTCG'	'AGCT'	'CGGG'	'AGAG'	'AAGA'	'TAGA'	'AGCG'
1	'CTTC'	'AAAA'	'TCAC'	'CGGC'	'CGGT'	'GAGT'	'TCCT'	'GAGT'	'TGTC'	'ATGG'	'TCAC'	'GCGC'	'AACT'	'GTCG'	'GTAA'	'CTCC'
2	'CTCA'	'CATT'	'TCGT'	'GGCT'	'ACTG'	'TAAA'	'TCCC'	'TTCC'	'AGCC'	'CTAG'	'ACAG'	'TTTC'	'TACG'	'CCCT'	'TGTG'	'GGCA'
3	'CACA'	'TAAA'	'GGTC'	'AACT'	'TGAG'	'CGCG'	'GGTC'	'AGTG'	'GTTA'	'AGGT'	'AATC'	'CTCC'	'AAGC'	'AAGT'	'TGGA'	'TCAA'
4	'TTGC'	'TTGG'	'GATG'	'CAAG'	'GATA'	'CAAG'	'CACA'	'GATG'	'CCCT'	'GTCG'	'ACCA'	'CGGG'	'AGAA'	'TGCG'	'GGTA'	'GCGA'
5	'AGAG'	'AGCA'	'GCCG'	'TAAA'	'TTTA'	'GGTG'	'AAGG'	'TAGC'	'TCTG'	'TTGA'	'CTCG'	'ACTG'	'TCGA'	'TTGG'	'AACT'	'GATG'
6	'ACAA'	'ACCG'	'GACT'	'CGGA'	'AGTC'	'CCAG'	'CCAT'	'CGGC'	'ACAT'	'TGTT'	'AGCG'	'TAAT'	'TGAA'	'GTTA'	'AGAT'	'TGAC'
7	'ACCA'	'ATCG'	'CCGC'	'ATTC'	'CCGG'	'ATGA'	'GACA'	'CAGG'	'AACC'	'TTAT'	'GATT'	'TGTC'	'AGGC'	'GGAC'	'CTTA'	'TCTT'
8	'GTAC'	'ATAT'	'GTCC'	'ACAG'	'CTTG'	'TGAG'	'GTCA'	'GGGT'	'TTAG'	'GGTA'	'TGGA'	'ACGC'	'GCTT'	'TTCG'	'GATC'	'AATT'
9	'TGTT'	'CAAC'	'ATGT'	'ATCG'	'GGTA'	'TCTG'	'ATCT'	'AGTC'	'CGCT'	'CACG'	'TCAA'	'ATGC'	'CAAG'	'CGGA'	'CGGA'	'CAAC'
10	'GAAG'	'ATGA'	'GGCT'	'GCTT'	'GAAA'	'ACTT'	'AGTC'	'AAAC'	'TGGG'	'TACT'	'CCCA'	'ATTG'	'GGCT'	'CTCC'	'GCTG'	'ATTT'
11	'CACG'	'GAGG'	'CGAT'	'CTGA'	'CAAG'	'ATAG'	'CTGC'	'CATA'	'TGGG'	'AGCA'	'CACG'	'TAAC'	'CGTC'	'CTCC'	'TGAT'	'AAAA'
12	'GGTT'	'GCCC'	'CGGG'	'TGAG'	'CATT'	'CGTT'	'CTCT'	'GGTA'	'AGTT'	'ATAG'	'GATA'	'CAGG'	'TAGC'	'GGCT'	'GTGT'	'CACA'
13	'CTCT'	'TAAC'	'CACG'	'CGCT'	'CGGG'	'AGGC'	'GGCA'	'GGCC'	'CCCC'	'TTAT'	'GTCG'	'TTTA'	'CTCA'	'GTGT'	'AGCG'	'CACT'
14	'CGCT'	'GAAG'	'GGTG'	'GTGA'	'AAAT'	'CGAA'	'ACTT'	'GATC'	'CAGT'	'ACCT'	'GTAT'	'GCCC'	'TCCA'	'CCTT'	'CTAC'	'CCCC'
15	'CAAT'	'GAAA'	'CTTC'	'AGTA'	'GCCT'	'CCGC'	'AGCT'	'CATA'	'TGTA'	'TTTC'	'ATGT'	'GTCG'	'GGGG'	'CGCC'	'TCGG'	'ACGA'

(b) The Output of DNA S-Box (2)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	'CAGT'	'GGAG'	'TAGC'	'GCGT'	'CGGG'	'ATAG'	'TAGC'	'GCAA'	'TGCA'	'GGGT'	'ACAT'	'GTAA'	'GCAC'	'GCCA'	'AGCA'	'TCAT'
1	'TTGG'	'GCCC'	'CGTC'	'CTAA'	'GTAA'	'GACA'	'TGTT'	'AACA'	'TGAG'	'ACGA'	'TGTC'	'CATA'	'ACCT'	'TAGT'	'GAGC'	'TTGT'
2	'GTGT'	'ATCG'	'CGTA'	'GAAT'	'TCTG'	'CGCC'	'CGTT'	'TGGT'	'GCAT'	'GTGC'	'GCTC'	'TGGG'	'AGCT'	'CTTT'	'TGAG'	'CAAT'
3	'CTCT'	'GGCC'	'CAAG'	'TCCT'	'GGAC'	'CTAT'	'AAAG'	'ACAG'	'TAGG'	'ACAA'	'ACCG'	'TTGT'	'TCCA'	'TCCA'	'AGAA'	'TGTC'
4	'AGGA'	'AGGA'	'GACG'	'ATCC'	'GACG'	'GTCC'	'CTCT'	'TACG'	'CTTT'	'CAGT'	'CCTT'	'CTAA'	'CCAC'	'CGAT'	'AAAG'	'AATA'
5	'GCAC'	'CCAT'	'CATT'	'GGCC'	'AGGG'	'TAAG'	'TCCA'	'TGCA'	'CGTG'	'AGGA'	'GTGT'	'TCTG'	'GGTA'	'AGGA'	'ACCT'	'AACG'
6	'TCTC'	'GCTT'	'AACT'	'ATAA'	'GCAG'	'ATTC'	'TTTC'	'ATAA'	'ACTC'	'CGAG'	'TCAT'	'CGCC'	'GGAC'	'TAGG'	'ACAC'	'TGAC'
7	'GCTT'	'GCGT'	'GTTA'	'CCGG'	'TTTA'	'CCGA'	'TACT'	'CTCA'	'GCCT'	'TGGC'	'CACG'	'GGAG'	'GCAA'	'GAAC'	'CTGG'	'CGTG'
8	'CAGC'	'GCGC'	'GAGT'	'ACTC'	'CTGG'	'GGAC'	'TAGT'	'GAAA'	'GGGC'	'TAAG'	'GGAA'	'GCTA'	'GATG'	'TGGT'	'GACG'	'TCCG'
9	'GGAG'	'ATCC'	'TCGA'	'GCGT'	'GAAG'	'TGTG'	'TCGT'	'TCAG'	'ATAT'	'ATCT'	'GGTC'	'TCGA'	'CTCC'	'GTAA'	'CTAA'	'CTCC'
10	'GACC'	'GCGA'	'GAAT'	'TATG'	'TACC'	'GCTG'	'GCAG'	'TCCC'	'AGAA'	'AGCT'	'GTTT'	'CCGG'	'TAAT'	'TTGT'	'CATG'	'CCGG'
11	'CTCT'	'AACA'	'ATAC'	'GTGA'	'ATCC'	'ACGC'	'TTGA'	'ATCG'	'TGAA'	'CCAT'	'CTCT'	'AGCC'	'ATAG'	'ATGT'	'GGAC'	'CCCC'
12	'TAAG'	'TATT'	'CTAA'	'CGAC'	'ATCG'	'GTAG'	'CTGT'	'TAAG'	'CCAG'	'CCGC'	'AACG'	'CTCA'	'GGCA'	'CAAT'	'AAGA'	'GTCT'
13	'CTGT'	'TGCC'	'CTCT'	'ATAT'	'CTAA'	'CCAA'	'TAAT'	'TAAT'	'ATTT'	'AGGC'	'TAGT'	'TGGG'	'ATGT'	'AAGA'	'CCAT'	'CTCT'
14	'CTAT'	'CACC'	'AAAG'	'GAGA'	'ACCC'	'ATAC'	'TCTG'	'TACG'	'CTCA'	'CCTT'	'CAGC'	'GATT'	'GGTT'	'GTTG'	'GTGC'	'GTTT'
15	'TTCC'	'CACC'	'TTGG'	'GCAG'	'CATT'	'TTTA'	'GCAT'	'TTCG'	'GGAG'	'GGGG'	'ACGA'	'AAGT'	'GAAA'	'ATAT'	'CGTA'	'TCTA'

(c) The Output of DNA S-Box (3)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	'GACT'	'TTTC'	'AGCT'	'TTAT'	'AATT'	'CCGC'	'GGCT'	'ATAC'	'CGTA'	'TTTT'	'GCAC'	'CTGC'	'ATAC'	'TTAA'	'TCTA'	'CGAC'
1	'GGGT'	'CTAA'	'AATG'	'GAGC'	'GTGC'	'ATCA'	'CGTG'	'ACCA'	'TGTC'	'GCAT'	'TGTG'	'CACG'	'CCAA'	'GGCT'	'TTCT'	'TGGT'
2	'GTGT'	'ACGA'	'CATG'	'TTCC'	'AGAG'	'TATA'	'GATG'	'GGTT'	'ATAC'	'ATGT'	'ATAG'	'TGTT'	'GCTA'	'AAGG'	'CGTC'	'GACC'
3	'CAGA'	'TTTA'	'AACC'	'GGAA'	'GTTC'	'GAGC'	'ACCC'	'TCAC'	'TGCT'	'CCAC'	'GCAA'	'AGGT'	'AGAA'	'AGAA'	'GCTC'	'GGTG'
4	'GCTT'	'CCTT'	'GTCA'	'CCGA'	'GTCA'	'TTGA'	'TAGA'	'AGCA'	'GAGG'	'CACT'	'TAAG'	'CAGC'	'GAAC'	'GATC'	'CCCC'	'ACCG'
5	'CTAC'	'TAAC'	'AACG'	'GTTA'	'TCTT'	'GGCC'	'GGAA'	'GGTA'	'TATG'	'GCTT'	'ATGT'	'AGAG'	'GTTG'	'GCTT'	'TCAA'	'GCCA'
6	'GGAG'	'ATAG'	'ACCA'	'ACGC'	'CTAC'	'GCGG'	'CGGG'	'ACGC'	'CCAG'	'CATC'	'TGAC'	'GATA'	'TTTC'	'CGCT'	'GCAC'	'CGTC'
7	'GTAG'	'CTAT'	'TTGG'	'GAAT'	'AGGG'	'TAAT'	'CGCA'	'GAGA'	'CTAA'	'AGTT'	'TACA'	'ATTC'	'CTAC'	'TTCC'	'AAGT'	'GATG'
8	'TACT'	'GTAT'	'CTCT'	'GCAG'	'GAGT'	'ATTC'	'AGCT'	'TTCC'	'ATTT'	'TGCC'	'ATTC'	'CTAG'	'ATCG'	'CGTT'	'GTCA'	'CGAA'
9	'TTTC'	'CCGA'	'CGAT'	'CTAT'	'CTCC'	'GGTG'	'TGAT'	'TGAC'	'GCGC'	'TCGA'	'CTTG'	'AGAT'	'TAGA'	'CTGC'	'CAGC'	'TAGA'
10	'TTCA'	'TTAT'	'GTCC'	'CGCG'	'CGCA'	'CTAG'	'CTAC'	'TGAA'	'ACTC'	'CCTA'	'ATGG'	'CAAT'	'TGCC'	'AGGT'	'GACG'	'GAAT'
11	'AAGA'	'ACCA'	'TCGC'	'TTGT'	'TCGA'	'ACAT'	'CGGT'	'CCGA'	'CGTC'	'TAAC'	'AAGA'	'CCTA'	'GCGC'	'ACGT'	'CTTC'	'GAAA'
12	'GGCC'	'CGCG'	'AAGC'	'CATC'	'GCGA'	'TTGC'	'GAGT'	'AGCC'	'AAAC'	'TAAT'	'TCCA'	'GAGA'	'GTTA'	'AACC'	'CCCT'	'GTGA'
13	'TAGT'	'CGTA'	'TAGA'	'ACGC'	'CAGC'	'TAAC'	'TGCC'	'CGCC'	'ACGG'	'ACTT'	'TGCT'	'TGTT'	'ACGT'	'ACCT'	'AAAC'	'TAGA'
14	'GAGC'	'GACA'	'GCCC'	'CTCT'	'ACAA'	'CCGC'	'TGAG'	'AGCA'	'CAGA'	'AAAG'	'TACT'	'ATCG'	'GTTG'	'TTGG'	'CTGT'	'CTGG'
15	'CGGA'	'TACA'	'TGGT'	'GTAC'	'TACG'	'CGGG'	'ATAC'	'AGGA'	'ATTC'	'GTTT'	'CCAT'	'GCCT'	'CTCC'	'CCGC'	'AATG'	'GGAG'

(d) The Output of DNA S-Box (4)

FIGURE 4. The output of different four DNAFZ S-boxes.

Algorithm 3 Substitution of Plaintext Image With DNAFZ S-Boxes

Input: Plaintext Image, P, its row, col, Four DNAFZ S-Boxes
Output: Four Substitution Images
1: Extract four images by applying downsampling for $P(row, col)$
2: Substitute of four images (P_1,P_2,P_3,P_4) with DNAFZ S-Boxes
3:for i=1 to 4
4: $Image = P(i)$
5: t = 1
6: for ii= 1 to row
7: for jj = 1 to col
8: $a = dec2bin(Image(ii, jj))$
9: b = 4 *MSB* of a
10: c = 4 *LSB* of a
11: $d = bin2dec\ (b)$
12: $e = bin2dec\ (c)$
13: $Image(ii, jj)$ = DNAFZ S-Box (d, e)
14: $t = t + 1$
15: end for
16: end for
17: end for
18:**return** Four Substitution Images

sub-image (1); DNAFZ S-box (2) produces the pixel value of sub-image (2), etc. The substitution steps are applied through following steps:

1) A binary code composed of 8 bits functions as the substitution input. This code represents numbers from 0 to 255, which are the pixel values of the sub-images.
2) A 16×16 matrix with columns and rows numbered from 0 to 15 marks the DNAFZ S-box.
3) To perform the value substitution for each pixel, the 8 input bits are assigned to rows and columns.
4) Thus, the first four bits are assigned to rows, while the second four are assigned to columns.
5) Each pixel's new substitution value is obtained by finding its location in the S-box, marked by the calculation of the number of rows and columns its bits are housed in. For an example on the calculation of the S-box and its transformation, see Figures 5.

D. FINAL CIPHER IMAGE BASED ON CHEN's HYPER MAP

To begin with, there is a brief explanation of how pseudo-random sequences are obtained from a hyper-chaotic system, in order to cipher an image. Said system has a secret key represented by its control parameter, as well as its initial conditions, including Chen's hyper-chaotic system's variables of x, y, z, q described in Section IV. These initial variables were set to $x_0 = 0.2243$, $y_0 = 0.5876$, $z_0 = 0.9855$, $a = 36$, $b = 3$, $c = 28$, $d = 16$, $k = 0.2$, leading to random sequences used in the encryption key. The four sequences obtained

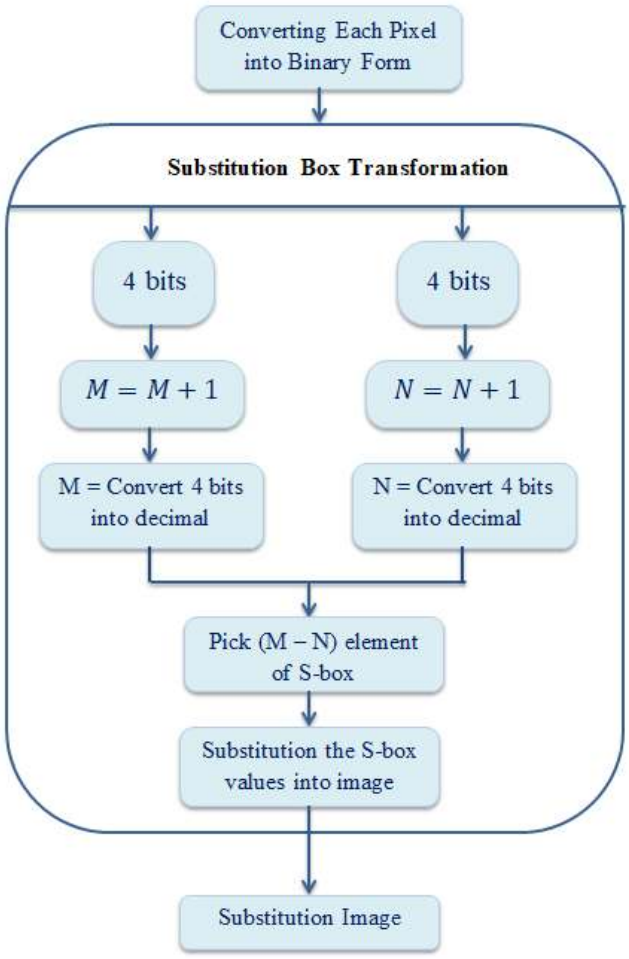


FIGURE 5. Example of substitution box transformation.

were transformed using the DNA technique. Each DNA rule applied was chosen as per the four remaining rules.

As per Table 2, each DNAFZ sub-image was XORed using a different random sequence obtained after DNA conversion of the value obtained from the chaotic system. These sequences were chosen by a control code. To explain this further, the control code yielded four sequences, one sequence per sub-image. Whereas DNAFZ sub-image (1) was encrypted by (q), DNAFZ sub-image (2) was encrypted by (x), DNAFZ sub-image (3) was encrypted by (z), and finally DNAFZ sub-image (4) was encrypted by (y). Following this step,four sub-images (DNAFZ/Chaotic) were changed to the binary system, according to Table 1 and each of its elements were converted into a decimal number. In the end, the four sub-images were combined to yield the final encrypted image.

VI. EXPERIMENTAL RESULTS

For an encryption algorithm to work efficiently, it needs to be able to encrypt an image, so that statistically it will have major differences in comparison to its original version. Results of the experiments tested in this article showed that

Algorithm 4 Final Cipher Image Based on Chen's Hyper Map

Input: Initial Variables (x_o, y_o, z_o, a, b, c, d, k), DNAFZ Sub-Images
Output: Final Cipher Image
1: Insert initial inputs
2: $x_o = 0.2243$, $y_o = 0.5876$, $z_o = 0.9855$
3: $a = 36$, $b = 3$, $c = 28$, $d = 16$, $k = 0.2$
4: Generate four random sequences (x, y, z, q)
5: # *Eq*. (5, 6, 7, 8)
6: $x = mod\ ((\ x \times 10^4), 256\ )\ )$
7: $y = mod\ ((\ y \times 10^4), 256\ )\ )$
8: $z = mod\ ((\ z \times 10^4), 256\ )\ )$
9: $q = mod\ ((\ q \times 10^4), 256\ )\ )$
10: Convert four random sequences into DNA
11: Use random sequences to apply XOR for each DNAFZ sub-images.
12: Convert each sub-images into decimal number
13: Combine four sub-images to produce final cipher image
14: **return** Final Cipher Image

the suggested algorithm is robust and has improved security features. The MATLAB software was used to carry out the experiments on grayscale images whose size was 256 × 256. Figure 6 marks the original, as well as encrypted images. From this figure, it can be seen that both images are completely different, proving the effectiveness of the encryption algorithm.

VII. EVALUATION ANALYSIS OF S-BOX

The majority logic criterion (MLC) evaluates the S-box's statistical capability for the different encryption methods. To measure the ability of S-boxes to confuse images, several techniques are available, whether analytical or statistical. Statistical analysis is quite beneficial, as it shows the ability of encryption to achieve distortion in an image. This analysis aids in achieving S-boxes with better encryption performance. Table 3 gives the results of the tested S-boxes versus other ones, in tests related to contrast, correlation, homogeneity, energy, and entropy. This later one tests the amount of randomness in an encrypted image. Homogeneity and energy analyses describe different characteristics of the encrypted image, while correlation evaluates the relation between the initial image and its encrypted version. On the other hand, contrast analysis estimates the loss of brightness in the host image. All data are evaluated using MLC, for the purpose of producing the best S-box for encryption [18], [19], [37], [38].

A. ENTROPY

Information obtained from entropy has two purposes, the first is to check the distribution of the gray value within the image and the second is to demonstrate the uncertainty in the image's information. Since the maximum value for entropy is eight, while the minimum is zero, an encryption method that pushes the value of entropy close to eight is optimum. To calculate entropy, Eq. 20 is used:

$$E = \sum_{i=1}^{N-1} P(X_i)\ log_2\ P(X_i) \tag{20}$$

where N is the total number of symbol (X) and $P(X_i)$ is the probability that symbol(X_i) will occur. An increasing value of entropy shows that pixels' gray values are more uniformly distributed. The security of an encrypted image decreases as its entropy value is much less than 8, marking an opportunity of predictability.

B. CONTRAST

Contrast has to do with the difference in the brightness of an object. To test contrast, a contrast analysis is performed enabling users to see objects and identify information beneath. To improve viewing, as well as visual effects, an adjustment of the contrast and brightness is performed, while processing an image. When encrypting an image, its contrast is directly proportional to its randomness, as the mapping of the S-box is not linear. Mathematically, it can be computed as:

$$Contrast = \sum |\ i - j\ |^2\ P(i, j) \tag{21}$$

where, $p(i, j)$ is the position of pixels in the gray-level co-occurrence matrix (GLCM).

C. CORRELATION

Correlation measures the degree of similarity between neighboring pixels. In regular images, neighboring pixels have a strong correlation. However, when an image is encrypted, the process tends to reduce this correlation. As such, an encrypted image that has negligible correlation over an insecure channel is robust. Correlation has the following formula:

$$Correlation = \sum (\frac{(i - \mu i)(j - \mu j)}{\sigma_i \sigma_j} \tag{22}$$

where, $p(i, j)$ indicates the value of the pixel. For a digital image, i is the value of the position of the row and j is the value of the position of the column. As for μ, its marks the variance, while σ is the standard deviation.

D. ENERGY

In an energy analysis, a calculation of the sum of the squared members of gray-level co-occurrence is made. Energy value is high, when high value pixels are found in certain areas of the plain image in the gray-level co-occurrence matrix. The energy of the encrypted image, as opposed to the plain image, is small, as these values are distributed. A computation of the associated value of an image is also performed in an energy analysis. It can be calculated as follows:

$$Energy = \sum P(i, j)^2 \tag{23}$$

where the number of GLCM matrices is represented by $p(i, j)$.

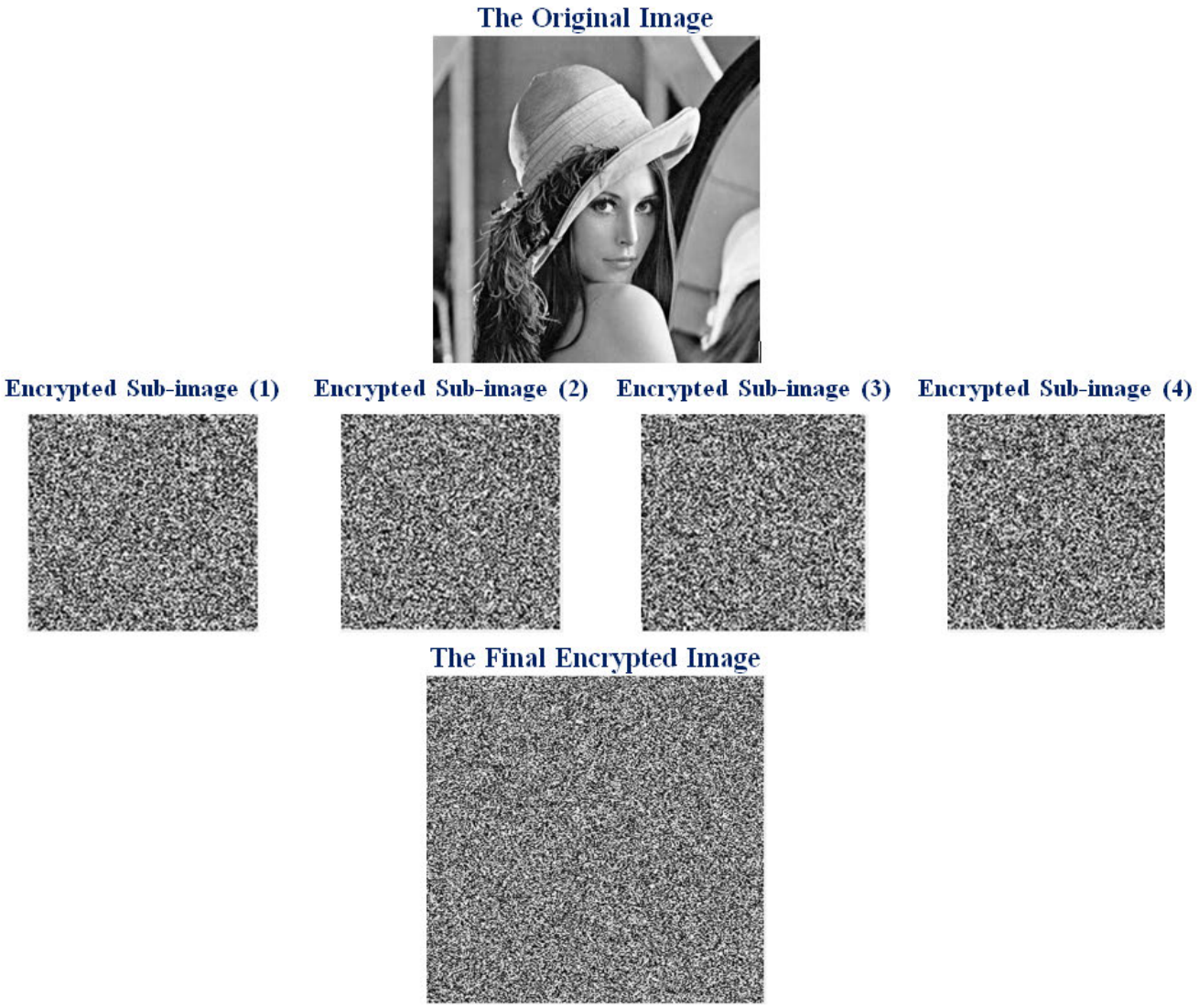


FIGURE 6. The original and final encrypted image.

E. HOMOGENEITY

In the GLCM, the intimacy of element distribution in the diagonal direction is known as homogeneity and its value essentially depends on the components existing on the diagonal of such a matrix. It is well-known that the smaller the value of homogeneity, the stronger the encryption algorithm. Homogeneity is calculated as per the following:

$$Homogeneity = \sum \frac{P(i,j)}{1+ \mid i-j \mid} \tag{24}$$

For the purpose of this article, an MLC encryption analysis was performed based on two regular plain images. This was done in order to show that the suggested S-boxes can be used to encrypt multimedia files. Table 3 shows the result of the analysis using the suggested S-boxes, as well those following the use of other types of S-boxes. Results of the MLC analysis proved that the proposed S-boxes are strong compared to others, as well as their adequacy for use in encryption applications and in algorithms meant for secure communication of multimedia files between two parties.

VIII. SECURITY ANALYSIS OF IMAGE ENCRYPTION

There are different types of attacks on multimedia files, including chosen cipher-text, statistical, known-plaintext, differential, as well as several brute-force attacks. A good cryptosystem must be able to resist all of them. The various types of analyses suggested above were conducted on the proposed algorithm, including key sensitivity analysis, NIST test, histogram analysis, Chi-square, information entropy, correlation between two adjacent pixels, PSNR, contrast, classical types of attack, and robustness test. Other types of differential analyses carried out were the unified average intensity change and the change in the number of pixel rate [42], [43].

TABLE 3. The comparative results of the proposed S-box with the other works when S-box is employed in encryption algorithms. It can be visualized that the proposed S-box has outperformed the other works.

S-Boxes	Entropy	Contrast	Correlation	Energy	Homogeneity
Baboon Image					
DNAFZ S-Box	7.9829	10.6423	4.2586e-04	0.0157	0.3859
Ref [17]	7.9817	10.4391	0.0128	0.0157	0.3889
Ref [39]	7.9775	10.5427	0.0006	0.0158	0.3905
Ref [40]	-	10.4227	0.0064	0.01566	0.398483
Ref [18]	7.9553	10.3466	-0.0087	0.0157	0.3895
Ref [19]	7.9248	7.5601	0.0348	0.0204	0.4649
Ref [38]	7.9820	8.6488	0.0060	0.0174	0.4062
Ref [15]	7.9822	8.5823	-0.0064	0.0175	0.4071
Ref [41]	7.9551	8.5267	4.4609e-04	0.0174	0.4088
Peppers Image					
DNAFZ S-Box	7.9857	10.7258	3.0881e-04	0.0157	0.3862
Ref [17]	7.9545	10.5377	-0.0061	0.0157	0.3864
Ref [18]	7.9566	10.3042	-0.0075	0.0157	0.3899
Ref [19]	7.9245	7.5598	0.0357	0.0201	0.4658
Ref [38]	7.9842	8.6969	-0.0134	0.0174	0.4045
Ref [15]	7.9823	8.6727	-0.0043	0.0173	0.4076
Ref [41]	7.9532	8.5155	0.0079	0.0174	0.4100

A. KEY SENSITIVITY ANALYSIS

In order to prevent a brute-force attack, the cryptosystem has to be sensitive to a secret key. To test this issue in this research, a plain image was first encrypted using a key, then decrypted with a key that was subjected to a slight change from the initial one [1], [44], [45]. This initial key is CFI's initial parameters, consisting of the external key and the secret image. The specificity of the generation code can be changed through a change in the initial parameters of the CFI for the specific system. The first step involves changing one bit of the external key, which changes the key itself. The variation percentage was 99.6074428 %, which proves the high sensitivity of the secret key. The second step calls for applying Gaussian noise to the secret image. Thus, four new S-boxes are produced and The difference percentage was 99.6554982 %, which proves the high sensitivity of the secret image. Similarly, hyper-chaotic maps are very sensitive to parameters related to the experiment's initial conditions. A small change in the key can lead to a decrypted image that is different from the encrypted one and the difference was 99.5682241%. Thus, it was concluded that the encryption algorithm is sensitive to the main key, as a slight change will lead to an incorrect image and incorrect results. Results of this experiment can be seen in Figure 7. This figure shows six images, (a) the key sensitivity with the change secret key, (c) the key sensitivity with the add noise to secret image key,(e) the key sensitivity with the change initial condition of hyper map, and (b), (e), (f) the difference between decrypted image with main key and decrypted image with modified key.

B. STATISTICAL ANALYSIS

1) NIST TEST SUITE ANALYSIS

NIST test is excuted to check the randomness of the binary sequence. It is a statical group consists of 15 tests. If the P-value for each test is greater than 0.01, the sequence has to be random [46], [47]. Table 4 shows the NIST test results of four FZ S-Boxes as compared to S-Boxes based on two types of chaotic maps. In addition, the test results of cipher image are presented in Table 5. It can be seen that all the tests have passed successfully.

2) HISTOGRAM ANALYSIS

This section marks the histogram analysis of both original and encrypted images. There does not seem to be any statistical resemblance between both. Original images' histograms are tilted and have long spikes. As for ciphered images, their histograms are flat and uniform. Figure 8 and 9 mark the 3D and 2D histograms of the original and encrypted images, respectively. The comparison between images in the two figures shows that the ciphered images are random like [44], [48]. To evaluate uniformity of the cipher images, the histogram's variance x determined as follows is applied for quantity analysis [49], [50]. The x_i and x_j represent the pixel's number with gray value equal to i and j, respectively.

$$Var(x) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(x_i - x_j)^2 \tag{25}$$

The smaller value of variances denotes the higher uniformity of ciphered images. The variance values of cipher images are shown in Table 6, from which can be seen that the simulation results indicate that the histograms of encrypted images are uniformly distributed. Moreover, the comparison is applied to other schemes, it can be seen that the variance values of the proposed algorithm are lower than other schemes. Therefore, the proposed scheme is more robust to resist the statistical attacks.

3) CHI-SQUARE TEST ANALYSIS

Chi-square test analysis is performed to measure the quantitatively of uniformity the gray value of pixel. Smaller the value of Chi square indicates the higher uniformity of grayscale [54], [55]. Chi-square test can be calculated

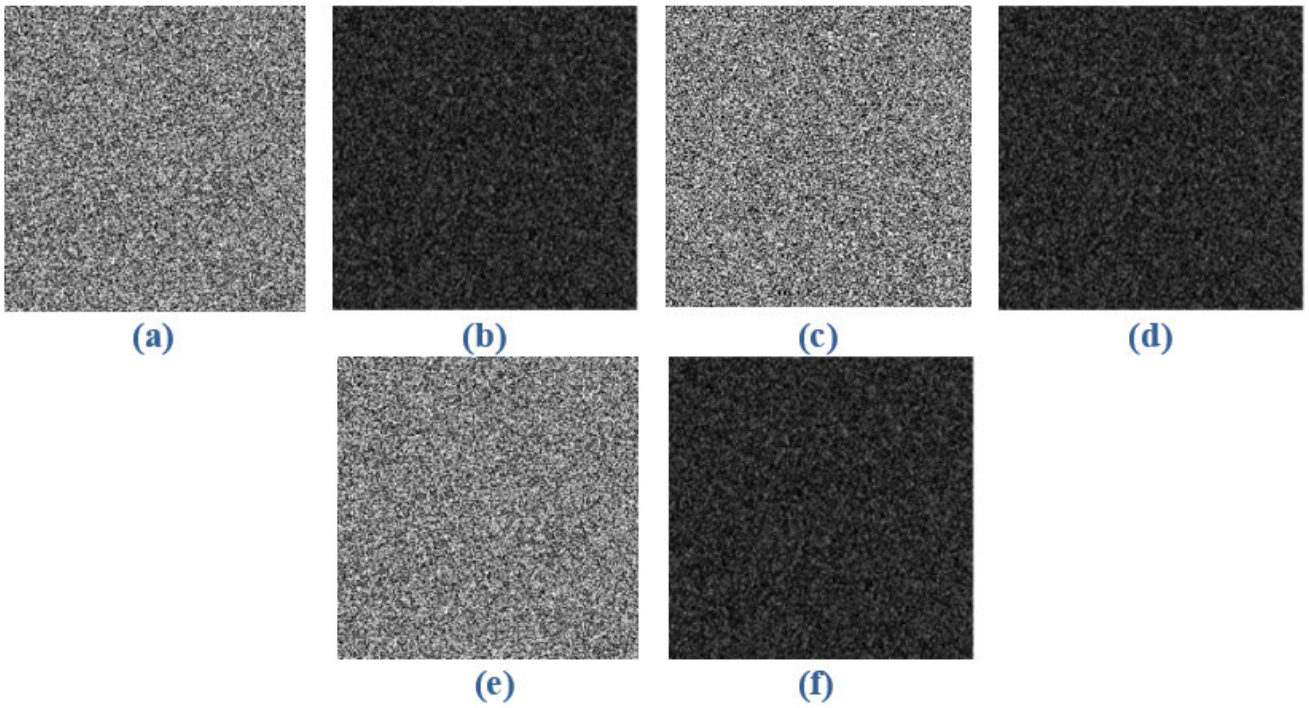


FIGURE 7. Key sensitivity analysis; (a) Key sensitivity with change one bit in secret key;(c) Key sensitivity with add noise in secret image; (f) Key sensitivity with change the key of hyper map; (b), (d), and (e) The difference between decrypted image with main key and decrypted image with modified key.

TABLE 4. NIST test results of four FZ S-boxes as compared to S-boxes based on two types of chaotic maps.

Test Name	P- Value (Four FZ S-Boxes)				(Tent- Logistic) Map [46]	Logistic Map [47]
	S-Box (1)	S-Box (2)	S-Box (3)	S-Box (4)		
Frequency	0.970882	0.950524	0.930915	0.9248898	0.911413	0.508261
Frequency within Blocks	0.972482	0.951224	0.949125	0.924227	0.897763	0.881134
Run Test	0.906377	0.974875	0.843235	0.8673774	0.202268	0.644401
Longest Run of Ones	0.915852	0.899832	0.845854	0.905992	0.897763	0.014259
Non-overlapping Template	0.943375	0.980529	0.53946	0.560907	0.045675	0.245325
Linear Complexity	0.673582	0.362203	0.484698	0.318547	0.090936	0.861830
Serial Test	0.985612	0.923178	0.953487	0.985612	0.935716	0.214860
Approximation Entropy	0.985612	0.92263	0.892495	0.768207	0.574903	0.935081
Cumulative Sum	0.696321	0.983913	0.755262	0.827715	0.637119	0.577018
Cumulative Sum Reverse	0.908149	0.927421	0.882196	0.986574	0.779188	0.622087
Random Excursions	0.627532	0.6470998	0.888136	0.986574	0.554420	0.579367
Random Excursions Variant	0.800908	0.911731	0.986493	0.943345	0.474986	0.44762
Rank Test	0.741908	0.550428	0.681248	0.576146	0.401199	0.673806

TABLE 5. NIST test of cipher image.

Test Name	P- Value (Cipher Image)	Status
Frequency	0.823668	SUCCESS
Frequency within Blocks	0.723251	SUCCESS
Run Test	0.561244	SUCCESS
Longest Run of Ones	0.652398	SUCCESS
Non-overlapping Template	0.550263	SUCCESS
Linear Complexity	0.432266	SUCCESS
Serial Test	0.826714	SUCCESS
Approximation Entropy	0.545477	SUCCESS
Cumulative Sum	0.961524	SUCCESS
Random Excursions	0.592214	SUCCESS
Random Excursions Variant	0.842119	SUCCESS
Rank Test	0.420566	SUCCESS

as follows:

$$X^2_{test} = \sum_{i=0}^{255} \frac{(obs_i - exp_i)^2}{exp_i} \tag{26}$$

where, exp_i and obs_i are the expected and observed frequencies, respectively. The expression for exp_i is:

$$exp_i = \frac{M \times N}{256} \tag{27}$$

where, $M \times N$ is the image size. The Chi-square test results of cipher images are as displayed in Table 7.

4) INFORMATION ENTROPY ANALYSIS

Information entropy serves two purposes: it shows uncertainty in the image's information and evaluates the distribution of its gray value. The smallest value for entropy is zero, while the maximum is eight. So, a good encryption algorithm needs to be able drive the information entropy up, as close as possible to 8 [1], [2]. The information entropy can be

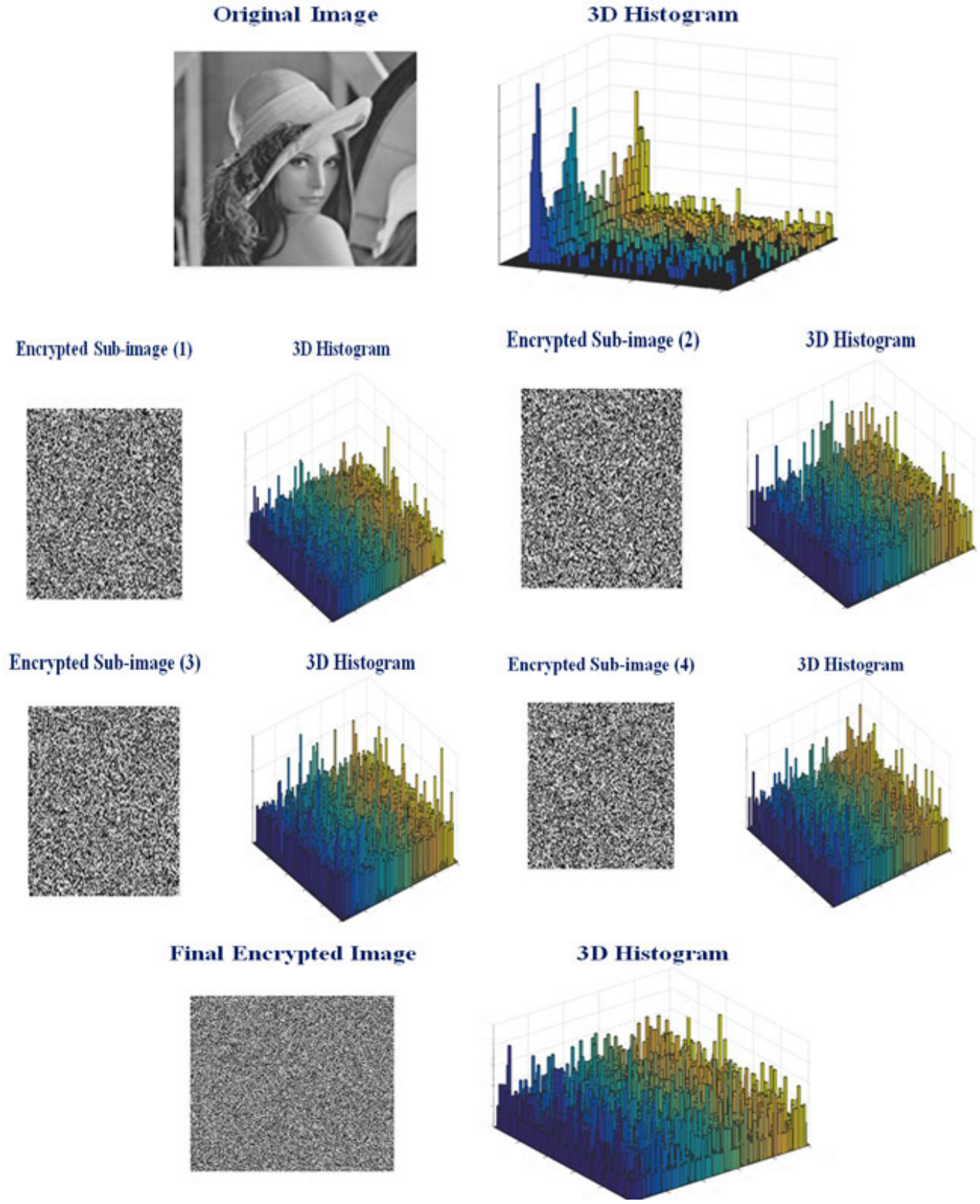


FIGURE 8. 3D histogram analysis.

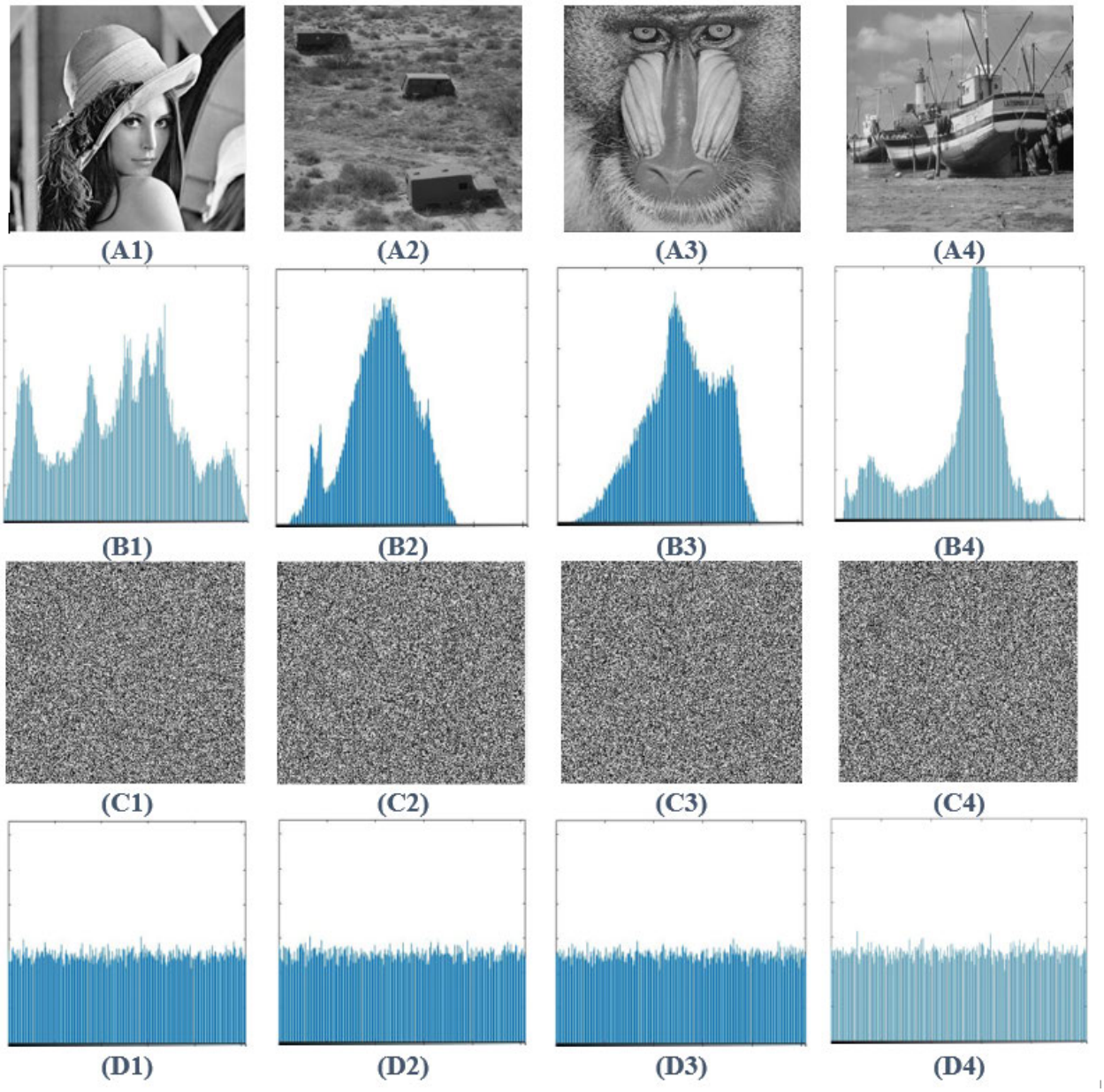


FIGURE 9. Histogram of original and encrypted images; (A1) - (A4) The original images; (B1) - (B4) The histogram of original images; (C1) - (C4) The encrypted images; (D1) - (D4) The histogram of encrypted images.

calculated according to the following equation:

$$E = \sum_{i=1}^{N-1} P(X_i)\, log_2\, P(X_i) \tag{28}$$

where N is the total number of symbol (X) and $P(X_i)$ is the probability that symbol (X_i) will occur. As previously stated, entropy needs to be close to 8 in the encrypted image. Table 8 marks a comparison between the entropy information of a cipher image encrypted with the proposed algorithm and those encrypted with other algorithms. It can be seen that the entropy of the cipher image encrypted with the suggested scheme is close to 8 (the ideal value). Our proposed scheme realizes better 4 results in overall, better 2 results in [22], [56]. This indicates that the suggested scheme achieves the best results, i.e., images encrypted using this scheme have random-pixel-value distribution.

5) CORRELATION COEFFICIENT

To determine the resemblance between two images, a correlational analysis is applied. This is especially done in encryption applications. Adjacent pixels' values are usually close, making them highly similar, with a higher correlation. An effective cryptosystem needs to be able to disrupt this correlation between adjacent pixels. Thus, if the correlation

TABLE 6. Variances of histograms for cipher images.

Algorithms	Images	Histogram Variance
Ours	Lena	243.3569
	Baboon	234.1804
	Cameraman	237.8039
	Boat	246.5020
	Truck	238.7294
	Peppers	247.9922
	Tree	245.3882
	Airplane	217.7804
	All Black	232.8549
	Tree (512 × 512)	262.7243
	Lena (512 × 512)	250.5214
	Airplane (512 × 512)	235.1847
Ref [2]	tree	270.5859
Ref [51]	Lena	262.46
Ref [52]	Peppers	950.9
Ref [53]	All Black	256.430

TABLE 7. The chi-square test of cipher images.

Algorithms	Images	The Chi-square Test
Ours	Lena	232.4063
	Baboon	223.2656
	Cameraman	226.8750
	Boat	235.3828
	Truck	230.6406
	Peppers	236.8672
	Tree	234.2734
	Airplane	216.9297
	Tree (512 × 512)	234.5233
	Lena (512 × 512)	233.1297
	Airplane (512 × 512)	240.2256

TABLE 8. Information entropy analysis.

Algorithms	Images	Information Entropy Analysis
Ours	Lena	7.9989
	Baboon	7.9965
	Cameraman	7.9986
	Boat	7.9969
	Truck	7.9984
	Peppers	7.9975
	Tree	7.9985
	Airplane	7.9976
	All Black	7.9971
	Tree (512 × 512)	7.9991
	Lena (512 × 512)	7.9995
	Airplane (512 × 512)	7.9989
Ref [2]	Lena	7.9971
Ref [22]	Cameraman	7.999354
Ref [23]	Boat	7.9968
Ref [57]	Airplane	7.996213
Ref [58]	Peppers	7.997275
Ref [56]	Baboon	7.9992859

coefficient is close to zero, then the encryption algorithm is effective [22], [57]. To calculate the correlation coefficient in either the vertical, horizontal, or diagonal directions, Eq. 29 is applied:

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x)} * \sqrt{D(y)}} \tag{29}$$

In the previous equation, r_{xy} is the correlation coefficient and COV is the covariance between pixels (x) and (y). (x) is the value of a pixel on the plaintext image, while (y) is the value of the same pixel on the cipher-text one and they are both gray-scale values. D(x) is the variance and E(x) is the mean. On Figure 10, one can see the correlation between two side by side pixels on the plain image, as well as on the cipher image, in the three directions (vertical, horizontal, and diagonal). It shows that the correlation between these pixels has significantly decreased on the cipher image. Table 9 depicts the correlation distribution of various images encrypted with the suggested algorithm, as well as the results of other images encoded with competing algorithms. The correlation in the original image was seen to be closer to 1, while that of the encrypted image was closer to 0 (better than after applying other algorithms). This means that the suggested algorithm can cause a decrease in the correlation between two side by side pixels in the encrypted image.

C. DIFFERENTIAL ATTACKS

To check on the effect of one-bit difference between the original image and the ciphered one, two indicators are used, for the analysis of a differential attack [56]–[59]. They are NPCR and UACI or number of pixels change rate and unified average changing intensity, respectively. They are calculated as follows:

$$NPCR = \frac{\sum_{i,j} D(i, j)}{W * H} * 100\ \% \tag{30}$$

$$UACI = \frac{\sum_{i,j} C_1(i, j) - C_2(i, j)}{255} * 100\ \% \tag{31}$$

W and H are the width and height of the cipher image, respectively. $C_1(i, j)$ is the encrypted image before change in one pixel for the plain image and $C_2(i, j)$ is the image after the change. Table 10 shows values of both UACI and NPCR, which are both close to the optimum values. Their values were 99.6094% for NPCR and 33.4635% for UACI. This proves that the algorithm exhibits high sensitivity towards changes in the original image, even if they are quite small. This means that it can resist differential attacks.

D. PEAK SIGNAL-TO-NOISE RATIO (PSNR)

The peak signal-to-noise ratio is utilized to measure the quality of the encryption algorithm. PSNR displays the variance in the value of pixel between the original image and the cipher image. PSNR can be measured as the following formula:

$$PSNR = 10\ log \left[\frac{M * N * 255^2}{\sum_{i=1}^{N} \sum_{i=1}^{M} [P(i, j) - C(i, j)]^2} \right] \tag{32}$$

where M and N represent the length and width of the image,respectively. $P(i, j)$ represents the pixel value of an original image, and $C(i, j)$ represents the pixel value of the cipher image. The encryption quality is best when the PSNR is small value. Table 11 displays the result of PSNR between original and encrypted images.

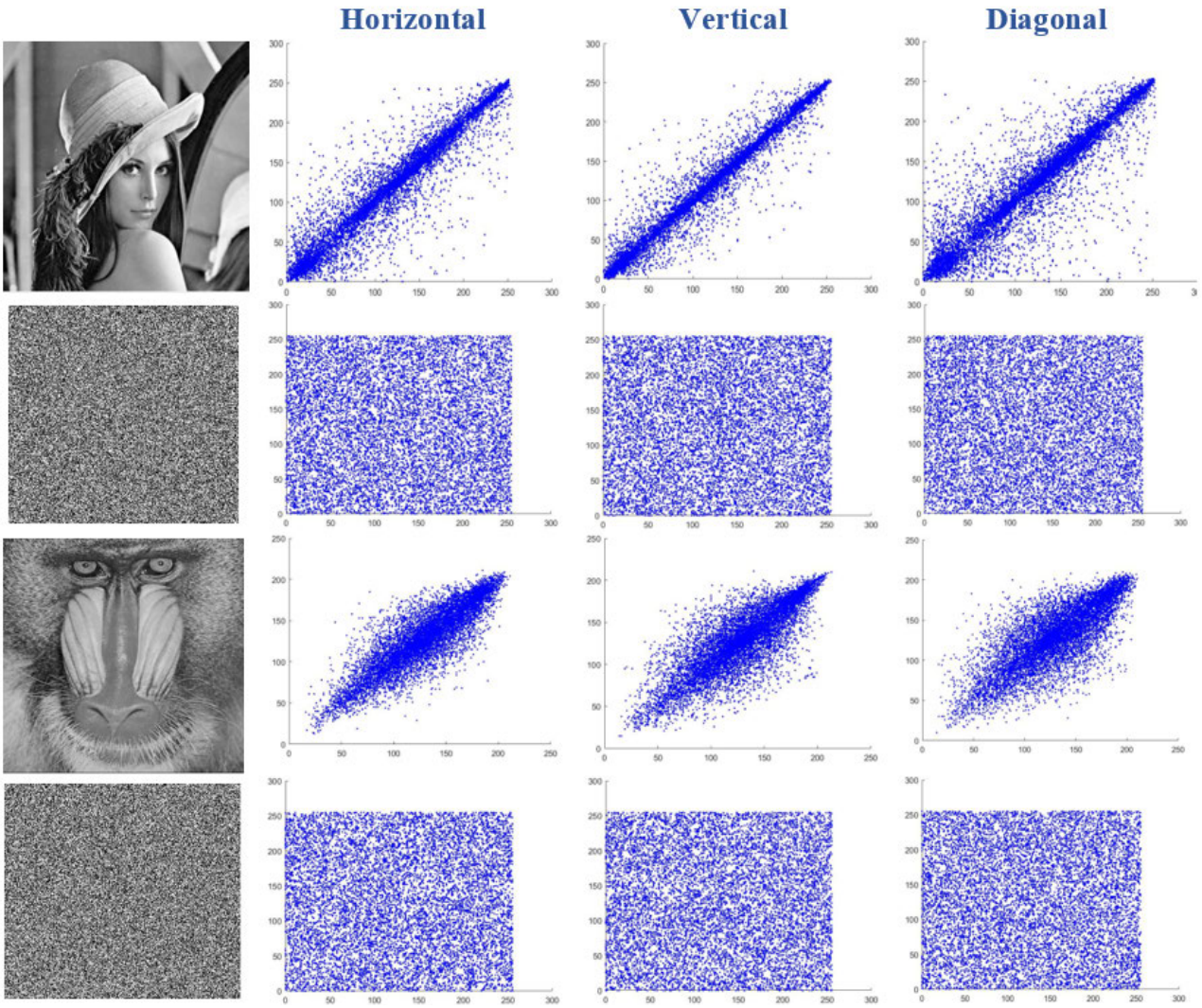


FIGURE 10. Correlation of two adjacent pixels of the original image Lena and its encrypted image.

E. CONTRAST

The intensity variation between pixels and their neighboring pixels can be measured through the contrast. The higher value of contrast indicates the superiority of encryption scheme [60], [61]. Mathematically, it can be calculated as:

$$Contrast = \sum | i - j |^2 P(i, j) \tag{33}$$

where, $p(i, j)$ is the position of pixels in the gray-level co-occurrence matrix (GLCM). Table 12 lists the contrast analysis of the proposed algorithm. It has the best results when compared to other algorithms.

F. SPEED ANALYSIS

The system used in the tests has the following specifications, a 3.0 GHz processor with 8GB RAM, running Windows 10 operating system and using MATLAB R2017b. This mark and dictates the speed of the algorithm. Table 13 demonstrates the encryption time of different images. As for Table 14, it shows the decryption time.

G. CLASSICAL TYPES OF ATTACKS

Four well-known attacks are utilized by attackers to break any cryptosystem which can be termed as ciphertext only attack, chosen ciphertext attack, known-plaintext attack, and chosen-plaintext attack [63], [64].

(1) Ciphertext only Attack: The contender has access to the string of ciphertext.

(2) Known Plain-text Attack: The contender has access to the strings of both plain and cipher-text.

(3) Chosen Ciphertext Attack: The contender can choose the ciphertext string and obtains the corresponding plaintext string.

(4) Chosen Plaintext Attack: The contender can choose the plaintext string and obtains the corresponding ciphertext string.

TABLE 9. Correlation coefficient between two adjacent pixels in the original and encrypted images.

Algorithms	Images	Correlation Coefficient					
		Horizontal		Vertical		Diagonal	
		Plain	Encrypted	Plain	Encrypted	Plain	Encrypted
Ours	Lena	0.9729	0.00022	0.9463	0.00045	0.9402	0.00241
	Baboon	0.8736	0.00020	0.8261	3.5871e-04	0.8132	0.00256
	Cameraman	0.9592	0.00017	0.9335	0.00011	0.9452	0.00251
	Boat	0.9268	7.8597e-04	0.9452	0.00452	0.9152	0.00215
	Truck	0.9477	0.00132	0.8955	6.070e-04	0.8745	0.00118
	Peppers	0.9634	0.00072	0.9704	0.00045	0.9014	0.00129
	Tree	0.9682	0.00021	0.9451	8.7856e-04	0.9312	0.00098
	Airplane	0.9166	0.00015	0.9318	4.3704e-04	0.9621	0.00021
	All Black	NA	0.0012	NA	0.0014	NA	0.0022
	Tree (512 × 512)	0.9325	7.7125e-04	0.9623	0.00014	0.9722	0.00115
	Lena (512 × 512)	0.9267	4.9225e-04	0.9327	8.3201e-04	0.9489	0.00032
	Airplane (512 × 512)	0.9244	0.00024	0.9458	0.00029	0.9528	0.00011
Ref [2]	Tree	0.967700	.000756	0.944820	0.00233	0.92933	0.003625
Ref [22]	Lena	0.938596	0.000455	0.966742	0.002175	0.916518	0.003801
Ref [23]	Baboon	0.8033	0.0044	0.8609	0.0009	0.85827	0.0030
Ref [57]	Airplane	0.9559	.013205	0.9310	.019524	0.8706	0.001245
Ref [58]	Peppers	0.9416	0.01597	0.9511	0.035035	0.9022	0.009679
Ref [56]	Lena	0.984537	-0.009448	0.976635	0.024064	0.953663	-0.004117

TABLE 10. NPCR and UACI for different images.

Algorithms	Images	NPCR	UACI
Ours	Lena	99.62	33.44
	Baboon	99.60	3.46
	Cameraman	99.60	33.44
	Boat	99.60	33.45
	Truck	99.62	33.44
	Peppers	99.61	33.46
	Tree	99.58	33.42
	Airplane	99.60	33.45
	All Black	99.55	33.40
	Tree (512 × 512)	99.57	33.41
	Lena (512 × 512)	99.61	33.42
	Airplane (512 × 512)	99.60	33.44
Ref [22]	Boat	99.60	33.43
Ref [23]	Peppers	99.57	33.32
Ref [57]	Peppers	99.61	33.40
Ref [58]	Cameraman	99.61	33.47
Ref [59]	Lena	99.64	33.57

TABLE 11. Peak signal to noise ratio.

Algorithms	Images	Peak Signal to Noise Ratio
Ours	Lena	9.4314
	Cameraman	9.4384
	Peppers	8.7723
	Airplane	8.9998
	Boat	9.4139
	Tree	9.4311
	Truck	9.2189
	Girl	8.9532
	Baboon	8.5412

TABLE 12. Contrast of cipher image.

Algorithms	Images	Contrast
Ours	Lena	10.8062
	Baboon	10.7947
	Cameraman	10.7951
	Boat	10.7982
	Truck	10.8415
	Peppers	10.8421
	Tree	10.8237
	Airplane	10.8312
	Tree (512 × 512)	10.8421
	Lena (512 × 512)	10.8552
	Airplane (512 × 512)	10.8219
Ref [60]	Lena	10.6201
Ref [61]	Lena	10.4880
Ref [62]	Cameraman	10.4554

TABLE 13. Encryption time consumption [unit:sec].

Algorithms	Images (256 × 256)	Encryption Time (unit:sec)
Ours	Lena	0.1838
	Baboon	0.1167
	Cameraman	0.1243
	Boat	0.1135
	Truck	0.1139
	Peppers	0.1131
	Tree	0.1149
	Airplane	0.1151

TABLE 14. Decryption time consumption [unit:sec].

Algorithms	Images (256 × 256)	Decryption Time (unit:sec)
Ours	Lena	0.4595
	Baboon	0.3684
	Cameraman	0.4107
	Boat	0.2837
	Truck	0.2961
	Peppers	0.2827
	Tree	0.2875
	Airplane	0.2992

The proposed image encryption technique has utilized XOR operation through the diffusion operation. It is very significant to check its resistance towards chosen plain text attack [65]–[67]. This attack analysis is applied as follows:

$$C_1(x, y) \oplus C_2(x, y) = P_1(x, y) \oplus P_2(x, y) \tag{34}$$

P_1 and P_2 are plain Lena and Cameraman images, where C_1 and C_2 are their corresponding encrypted forms respectively. If both C and P are similar, then the algorithm is vulnerable to chosen plain text attack. Figure 11 ensures that the XOR of plain text and ciphertext are not similar which means the proposed algorithm is not vulnerable to chosen plain text attack.

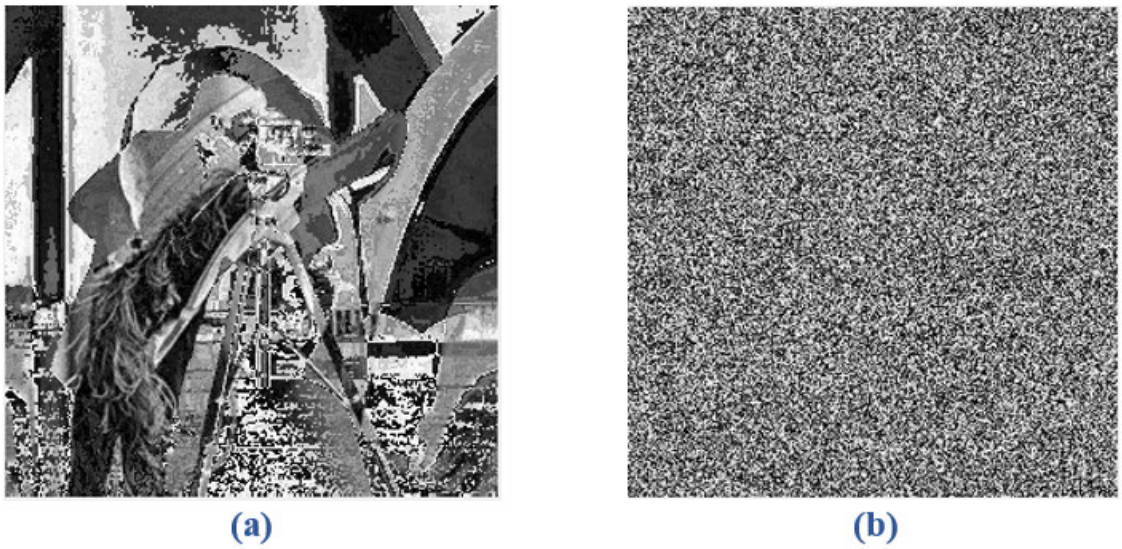


FIGURE 11. Chosen plain text analysis: (a) XOR, Two original images. (b) XOR, The cipher image of its original image.

FIGURE 12. Decrypted images under salt and pepper noise with: (a)0.005; (b) 0.05; (c) 0.1.

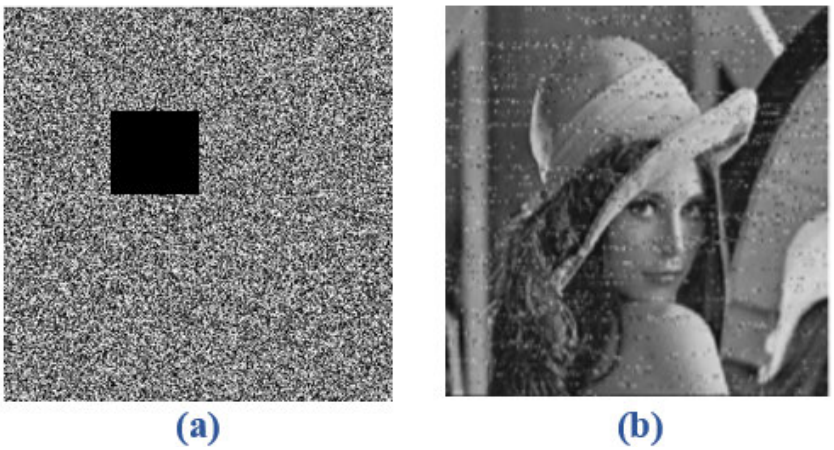


FIGURE 13. Occlusion attack: (a) Encrypted image with lost block; (b) Decrypted image.

H. ROBUSTNESS TEST

Several types of attacks were used to test the proposed algorithm, including salt and pepper noise, occlusion attack. PSNR is measured in this article to analyze the quality of the decrypted image in comparison with the plain-image. The greater value of PSNR results in minimal distortion in the

TABLE 15. PSNR of decrypted image after different attacks.

Test	PSNR (dB)
occlusion	20.0421
Salt and Pepper noise (0.005)	30.7847
Salt and Pepper noise (0.05)	21.1864
Salt and Pepper noise (0.1)	18.3045

plain image [55], [68]. The results are listed in Table 15 which displays the PSNR values of the decrypted images as the encrypted images are attacked by various image operations. The robustness of the proposed algorithm is explained by the results that the decrypted images are still identifiable in spite of the cipher-image being distorted. Test result displays that we can still decrypt the encrypted image when the encrypted image is added salt and pepper noise, as shown in Figure 12 as well as occlusion attack displayed in Figure 13.

IX. DISCUSSION AND CONCLUSION

There are several available encryption algorithms developed by researchers, however technological advances do not cease to require better, more secure, and more resistant schemes to ward off attacks. A more secure method based on a new designed S-Box based DNA-like techniques and CFI; and combined with hyper-chaotic maps, is suggested in this article. The new proposed S-boxes are more secure and highly complex and are shown to be excellent choices for robust image encryption. Some considerations were taken to ensure resistance against attacks. The first one was to uses CFI to develop four random sequences. To create the initial input, a secret image key and a finite length key were used. The second consideration was basing the four different FZ S-boxes on CFI. The third consideration was using a DNA-inspired technique and applying it through a control code to select four of the eight DNA rules, one per each FZ S-box. The fourth one was extracting four sub-images from the plain image, through downsampling. The fifth one was changing the values of the sub-images with values of the four DNAFZ S-boxes. The sixth consideration was diffusing each of the DNAFZ sub-images using a random DNA sequence extracted from Chen's hyper-chaotic map. The final consideration was combining the four DNAFZ/chaotic sub-images, to form the cipher image. The proposed algorithm has been proven effective and secure through tests and security analyses. To test the security of the new algorithm different 256×256 images were used, with results indicating its high sensitivity to its key. The values of the pixel correlation coefficient were quite small either horizontally, vertically, or diagonally. In addition, the information entropy of the encrypted image was very near to the ideal value of 8. As for the UACI and the NPCR values were also very close to the optimum ones. These results show that the performance of the proposed algorithm is advantageous over most other of other image encryption techniques. Hence, we can conclude that the proposed DNAFZ S-Box and encryption algorithm have the potential to be used in image encryption, through future secure multimedia communication systems.

REFERENCES

[1] E. Hasanzadeh and M. Yaghoobi, "A novel color image encryption algorithm based on substitution box and hyper-chaotic system with fractal keys," *Multimedia Tools Appl.*, vol. 79, pp. 7279–7297, Dec. 2019.

[2] Q. Lu, C. Zhu, and X. Deng, "An efficient image encryption scheme based on the LSS chaotic map and single S-Box," *IEEE Access*, vol. 8, pp. 25664–25678, 2020.

[3] Ü. Çavu o lu, S. Kaçar, I. Pehlivan, and A. Zengin, "Secure image encryption algorithm design using a novel chaos based S-Box," *Chaos, Solitons Fractals*, vol. 95, pp. 92–101, Feb. 2017.

[4] A. Jain, P. Agarwal, R. Jain, and V. Singh, "Chaotic image encryption technique using S-box based on DNA approach," *Int. J. Comput. Appl.*, vol. 92, no. 13, pp. 30–34, Apr. 2014.

[5] C. Zhu, Z. Gan, Y. Lu, and X. Chai, "An image encryption algorithm based on 3-D DNA level permutation and substitution scheme," *Multimedia Tools Appl.*, vol. 79, pp. 7227–7258, Dec. 2019.

[6] J. Chen, Z.-L. Zhu, L.-B. Zhang, Y. Zhang, and B.-Q. Yang, "Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption," *Signal Process.*, vol. 142, pp. 340–353, Jan. 2018.

[7] D. Dubois and H. Prade, "A review of fuzzy set aggregation connectives," *Inf. Sci.*, vol. 36, nos. 1–2, pp. 85–121, Jul. 1985.

[8] M. Grabisch and J.-M. Nicolas, "Classification by fuzzy integral: Performance and tests," *Fuzzy Sets Syst.*, vol. 65, nos. 2–3, pp. 255–271, Aug. 1994.

[9] H. Liu, X. Wang, and A. Kadir, "Chaos-based color image encryption using one-time keys and choquet fuzzy integral," *Int. J. Nonlinear Sci. Numer. Simul.*, vol. 15, no. 1, pp. 1–10, Jan. 2014.

[10] S. M. Seyedzadeh and S. Mirzakuchaki, "Image encryption scheme based on choquet fuzzy integral with pseudo-random keystream generator," in *Proc. Int. Symp. Artif. Intell. Signal Process. (AISP)*, Jun. 2011, pp. 101–106.

[11] Q. Zhang, L. Guo, and X. Wei, "A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system," *Optik-Int. J. Light Electron Opt.*, vol. 124, no. 18, pp. 3596–3600, Sep. 2013.

[12] N. Zhou, S. Pan, S. Cheng, and Z. Zhou, "Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing," *Opt. Laser Technol.*, vol. 82, pp. 121–133, Aug. 2016.

[13] B. Norouzi and S. Mirzakuchaki, "A fast color image encryption algorithm based on hyper-chaotic systems," *Nonlinear Dyn.*, vol. 78, no. 2, pp. 995–1015, Oct. 2014.

[14] A. Belazi, M. Khan, A. A. Abd El-Latif, and S. Belghith, "Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 337–361, 2017.

[15] A. Ullah, S. S. Jamal, and T. Shah, "A novel construction of substitution box using a combination of chaotic maps with improved chaotic range," *Nonlinear Dyn.*, vol. 88, no. 4, pp. 2757–2769, Jun. 2017.

[16] T. Ye and L. Zhimao, "Chaotic S-box: Six-dimensional fractional Lorenz–Duffing chaotic system and O-shaped path scrambling," *Nonlinear Dyn.*, vol. 94, no. 3, pp. 2115–2126, 2018.

[17] A. Razaq, H. Alolaiyan, M. Ahmad, M. A. Yousaf, U. Shuaib, W. Aslam, and M. Alawida, "A novel method for generation of strong substitution-boxes based on coset graphs and symmetric groups," *IEEE Access*, vol. 8, pp. 75473–75490, 2020.

[18] M. A. Yousaf, H. Alolaiyan, M. Ahmad, M. Dilbar, and A. Razaq, "Comparison of pre and post-action of a finite abelian group over certain nonlinear schemes," *IEEE Access*, vol. 8, pp. 39781–39792, 2020.

[19] A. Anees and Y.-P.-P. Chen, "Designing secure substitution boxes based on permutation of symmetric group," *Neural Comput. Appl.*, vol. 32, no. 11, pp. 7045–7056, Jun. 2020.

[20] X. Wang and Q. Wang, "A novel image encryption algorithm based on dynamic S-boxes constructed by chaos," *Nonlinear Dyn.*, vol. 75, no. 3, pp. 567–576, Feb. 2014.

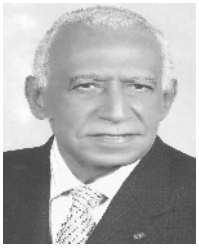
[21] X. Zhang, Z. Zhao, and J. Wang, "Chaotic image encryption based on circular substitution box and key stream buffer," *Signal Process., Image Commun.*, vol. 29, no. 8, pp. 902–913, Sep. 2014.

[22] H. Liu, B. Zhao, and L. Huang, "Quantum image encryption scheme using arnold transform and S-box scrambling," *Entropy*, vol. 21, no. 4, p. 343, Mar. 2019.

[23] B. Idrees, S. Zafar, T. Rashid, and W. Gao, "Image encryption algorithm using S-box and dynamic Hénon bit level permutation," *Multimedia Tools Appl.*, vol. 79, no. 9, pp. 6135–6162, 2020.
[24] X.-Y. Wang, L. Yang, R. Liu, and A. Kadir, "A chaotic image encryption algorithm based on perceptron model," *Nonlinear Dyn.*, vol. 62, no. 3, pp. 615–621, Nov. 2010.
[25] H. Liu, X. Wang, and A. Kadir, "Image encryption using DNA complementary rule and chaotic maps," *Appl. Soft Comput.*, vol. 12, no. 5, pp. 1457–1466, May 2012.
[26] H. Liu and X. Wang, "Color image encryption using spatial bit-level permutation and high-dimension chaotic system," *Opt. Commun.*, vol. 284, nos. 16–17, pp. 3895–3903, Aug. 2011.
[27] H. Liu and X. Wang, "Color image encryption based on one-time keys and robust chaotic maps," *Comput. Math. with Appl.*, vol. 59, no. 10, pp. 3320–3327, May 2010.
[28] S. E. El-Khamy and A. G. Mohamed, "Image keyed PN sequence generator and authentication technique based on choquet fuzzy integral," in *Proc. 35th Nat. Radio Sci. Conf. (NRSC)*, Mar. 2018, pp. 293–299.
[29] S. E. El-Khamy, N. O. Korany, and A. G. Mohamed, "A new fuzzy-DNA image encryption and steganography technique," *IEEE Access*, vol. 8, pp. 148935–148951, 2020.
[30] R. Enayatifar, F. G. Guimarās, and P. Siarry, "Index-based permutation-diffusion in multiple-image encryption using DNA sequence," *Opt. Lasers Eng.*, vol. 115, pp. 131–140, Apr. 2019.
[31] X. Chai, Z. Gan, K. Yuan, Y. Chen, and X. Liu, "A novel image encryption scheme based on DNA sequence operations and chaotic systems," *Neural Comput. Appl.*, vol. 31, no. 1, pp. 219–237, Jan. 2019.
[32] S. Suri and R. Vijay, "A synchronous intertwining logistic map-DNA approach for color image encryption," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 6, pp. 2277–2290, Jun. 2019.
[33] H. Liu, X. Wang, and A. Kadir, "Color image encryption using choquet fuzzy integral and hyper chaotic system," *Optik-Int. J. Light Electron Opt.*, vol. 124, no. 18, pp. 3527–3533, Sep. 2013.
[34] S. M. Seyedzadeh, B. Norouzi, and S. Mirzakuchaki, "RGB color image encryption based on choquet fuzzy integral," *J. Syst. Softw.*, vol. 97, pp. 128–139, Nov. 2014.
[35] J. Zhang, D. Hou, and H. Ren, "Image encryption algorithm based on dynamic DNA coding and Chen's hyperchaotic system," *Math. Problems Eng.*, vol. 2016, Sep. 2016, Art. no. 6408741.
[36] R.-C. Tan, T. Lei, Q.-M. Zhao, L.-H. Gong, and Z.-H. Zhou, "Quantum color image encryption algorithm based on a hyper-chaotic system and quantum Fourier transform," *Int. J. Theor. Phys.*, vol. 55, no. 12, pp. 5368–5384, Dec. 2016.
[37] A. Razaq, H. A. Al-Olayan, A. Ullah, A. Riaz, and A. Waheed, "A novel technique for the construction of safe substitution boxes based on cyclic and symmetric groups," *Secur. Commun. Netw.*, vol. 2018, pp. 1–9, Oct. 2018.
[38] S. S. Jamal, A. Anees, M. Ahmad, M. F. Khan, and I. Hussain, "Construction of cryptographic S-boxes based on mobius transformation and chaotic tent-sine system," *IEEE Access*, vol. 7, pp. 173273–173285, 2019.
[39] A. Javeed and T. Shah, "Design of an S-box using rabinovich-fabrikant system of differential equations perceiving third order nonlinearity," *Multimedia Tools Appl.*, vol. 79, nos. 9–10, pp. 6649–6660, Mar. 2020.
[40] M. Ahmad and E. Al-Solami, "Evolving dynamic S-Boxes using fractional-order hopfield neural network based scheme," *Entropy*, vol. 22, no. 7, p. 717, Jun. 2020.
[41] A. Razaq, A. Yousaf, U. Shuaib, N. Siddiqui, A. Ullah, and A. Waheed, "A novel construction of substitution box involving coset diagram and a bijective map," *Secur. Commun. Netw.*, vol. 2017, Nov. 2017, Art. no. 5101934.
[42] Y. Naseer, D. Shah, and T. Shah, "A novel approach to improve multimedia security utilizing 3D mixed chaotic map," *Microprocessors Microsyst.*, vol. 65, pp. 1–6, Mar. 2019.
[43] A. Shafique, "A new algorithm for the construction of substitution box by using chaotic map," *Eur. Phys. J. Plus*, vol. 135, no. 2, pp. 1–13, Feb. 2020.
[44] K. A. Kumar Patro and B. Acharya, "An efficient colour image encryption scheme based on 1-D chaotic maps," *J. Inf. Secur. Appl.*, vol. 46, pp. 23–41, Jun. 2019.
[45] Y.-Q. Zhang and X.-Y. Wang, "A new image encryption algorithm based on non-adjacent coupled map lattices," *Appl. Soft Comput.*, vol. 26, pp. 10–20, Jan. 2015.
[46] Lu, Zhu, and Wang, "A novel S-box design algorithm based on a new compound chaotic system," *Entropy*, vol. 21, no. 10, p. 1004, Oct. 2019.
[47] M. S. Mahmood Malik, M. A. Ali, M. A. Khan, M. Ehatisham-Ul-Haq, S. N. M. Shah, M. Rehman, and W. Ahmad, "Generation of highly nonlinear and dynamic AES substitution-boxes (S-Boxes) using chaos-based rotational matrices," *IEEE Access*, vol. 8, pp. 35682–35695, 2020.
[48] R. Zahmoul, R. Ejbali, and M. Zaied, "Image encryption based on new beta chaotic maps," *Opt. Lasers Eng.*, vol. 96, pp. 39–49, Sep. 2017.
[49] Y.-Q. Zhang and X.-Y. Wang, "A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice," *Inf. Sci.*, vol. 273, pp. 329–351, Jul. 2014.
[50] K. A. K. Patro, B. Acharya, and V. Nath, "Secure multilevel permutation-diffusion based image encryption using chaotic and hyper-chaotic maps," *Microsyst. Technol.*, vol. 25, no. 12, pp. 4593–4607, Dec. 2019.
[51] X. Wang and P. Liu, "A new image encryption scheme based on a novel one-dimensional chaotic system," *IEEE Access*, vol. 8, pp. 174463–174479, 2020.
[52] Y. Luo, X. Ouyang, J. Liu, and L. Cao, "An image encryption method based on elliptic curve elgamal encryption and chaotic systems," *IEEE Access*, vol. 7, pp. 38507–38522, 2019.
[53] S. Zhu, G. Wang, and C. Zhu, "A secure and fast image encryption scheme based on double chaotic S-boxes," *Entropy*, vol. 21, no. 8, p. 790, Aug. 2019.
[54] K. A. K. Patro, A. Soni, P. K. Netam, and B. Acharya, "Multiple grayscale image encryption using cross-coupled chaotic maps," *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102470.
[55] K. A. K. Patro, B. Acharya, and V. Nath, "Secure, lossless, and noise-resistive image encryption using chaos, hyper-chaos, and DNA sequence operation," *IETE Tech. Rev.*, vol. 37, no. 3, pp. 223–245, 2019.
[56] Y. Zhang, "The unified image encryption algorithm based on chaos and cubic S-Box," *Inf. Sci.*, vol. 450, pp. 361–377, Jun. 2018.
[57] M. B. Farah, R. Guesmi, A. Kachouri, and M. Samet, "A new design of cryptosystem based on S-box and chaotic permutation," *Multimedia Tools Appl.*, vol. 79, pp. 19129–19150, Mar. 2020.
[58] M. B. Farah, A. Farah, and T. Farah, "An image encryption scheme based on a new hybrid chaotic map and optimized substitution box," *Nonlinear Dyn.*, vol. 99, pp. 3041–3064, Dec. 2019.
[59] Y. Liu, J. Wang, J. Fan, and L. Gong, "Image encryption algorithm based on chaotic system and dynamic S-boxes composed of DNA sequences," *Multimedia Tools Appl.*, vol. 75, no. 8, pp. 4363–4382, Apr. 2016.
[60] S. Askar, A. Karawia, A. Al-Khedhairi, and F. Al-Ammar, "An algorithm of image encryption using logistic and two-dimensional chaotic economic maps," *Entropy*, vol. 21, no. 1, p. 44, Jan. 2019.
[61] Y.-Q. Zhang, J.-L. Hao, and X.-Y. Wang, "An efficient image encryption scheme based on S-Boxes and fractional-order differential logistic map," *IEEE Access*, vol. 8, pp. 54175–54188, 2020.
[62] S. M. Ismail, L. A. Said, A. G. Radwan, A. H. Madian, and M. F. Abu-ElYazeed, "A novel image encryption system merging fractional-order edge detection and generalized chaotic maps," *Signal Process.*, vol. 167, Feb. 2020, Art. no. 107280.
[63] X. Wang, L. Teng, and X. Qin, "A novel colour image encryption algorithm based on chaos," *Signal Process.*, vol. 92, no. 4, pp. 1101–1108, Apr. 2012.
[64] D. S. Malik and T. Shah, "Color multiple image encryption scheme based on 3D-chaotic maps," *Math. Comput. Simul.*, vol. 178, pp. 646–666, Dec. 2020.
[65] A. S. Banu and R. Amirtharajan, "A robust medical image encryption in dual domain: Chaos-DNA-IWT combined approach," *Med. Biol. Eng. Comput.*, vol. 58, pp. 1445–1458, Apr. 2020.
[66] S. A. Banu and R. Amirtharajan, "Tri-level scrambling and enhanced diffusion for DICOM image cipher-DNA and chaotic fused approach," *Multimedia Tools Appl.*, vol. 79, no. 39, pp. 28807–28824, 2020.
[67] S. Rajagopalan, S. Rethinam, S. Arumugham, H. N. Upadhyay, J. B. B. Rayappan, and R. Amirtharajan, "Networked hardware assisted key image and chaotic attractors for secure RGB image communication," *Multimedia Tools Appl.*, vol. 77, no. 18, pp. 23449–23482, Sep. 2018.
[68] K. A. K. Patro and B. Acharya, "Novel data encryption scheme using DNA computing," in *Advances of DNA Computing in Cryptography*. 2018, pp. 69–110.

AMIRA G. MOHAMED (Student Member, IEEE) received the B.S. degree in electronics and communication engineering from the Alexandria Higher Institute of Engineering and Technology (AIET), Alexandria, Egypt, in 2013, and the M.S. degree in electrical engineering from the Faculty of Engineering, Alexandria University, Alexandria, in 2017, where she is currently pursuing the Ph.D. degree in electrical engineering. She is currently a Teaching Assistant with the Department of Electronics and Communication Department, AIET. Her research interests include image processing, steganography, cryptography, and information security.

NOHA O. KORANY received the B.Sc. degree in engineering from Alexandria University, Egypt, and the Ph.D. degree from Alexandria University. She was a member of the Scientific Staff of the Institute of Communication Acoustics, Ruhr-University at Bochum, Germany from 2002 to 2004. She is currently a Professor with the University of Alexandria, Egypt. She received a fellowship at Ruhr-University, Bochum, Germany. Her main research field is acoustics and communications.



SAID E. EL-KHAMY (Life Fellow, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees from Alexandria University, Alexandria, Egypt, in 1965 and 1967, respectively, and the Ph.D. degree from the University of Massachusetts, Amherst, USA, in 1971. He has been a Teaching Staff with the Department of Electrical Engineering, Faculty of Engineering, Alexandria University, since 1972, and was appointed as a Full-Time Professor in 1982 and the Chairman of the Department of Electrical Engineering from 2000 to 2003, where he is currently an Emeritus Professor. His current research areas of interest include wireless multimedia communications, wave propagation, smart antenna arrays, modern signal processing techniques, image processing, and security and watermarking techniques. He has authored or coauthored about 400 scientific articles in national and international conferences and journals. He took part in the organization of many local and international conferences, including the yearly series of NRSC (URSI) conference series from 1990 to 2019, ISCC'95, ISCC'97, ISSPIT'2000, MELECON'2002, and IEEE GCIoT'2019. He took part in many IEEE Region eight activities and URSI general assemblies.

Prof. El-Khamy has earned many national and international research awards among which are the R. W. P. King Best Paper Award of the Antennas and Propagation Society of the IEEE in 1980, the Egypt's State Engineering Encouraging Research Award for two times in 1980 and 1989, respectively; Abdel-Hamid Schoman–Kingdom of Jordan award for Engineering Research in 1982, the State Scientific Excellence Award in Engineering Sciences in 2002, Alexandria University Appreciation Award of Engineering Sciences in 2003; State Appreciation Award of Engineering Sciences for 2004 and as the IEEE Region 8 Volunteer Award in 2010. In 2016, he was honored by Egypt's National Radio Science Committee of URSI and was selected as the Radio Science recognized figure of the year. In 2016, he was announced to be The Distinct Scientist of Alexandria University, in Engineering Sciences.

Prof. El-Khamy has been a Fellow Member of the IEEE since 1999 and has been an IEEE Life Fellow member since 2010. He is also a Fellow of the Electromagnetic Academy. He is the founder and the Past President of the IEEE Alexandria/Egypt Subsection and the past President of Egypt's National Radio Science Committee of URSI.