IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 15, 2021, accepted January 4, 2022, date of publication January 13, 2022, date of current version February 9, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3143096

A Color Image Encryption Scheme Combining Hyperchaos and Genetic Codes

HIRA NAZIR[1], IMRAN SARWAR BAJWA[1], SAIMA ABDULLAH[1], RAFAQUT KAZMI[2], AND MUHAMMAD SAMIULLAH[3]

[1]Department of Computer Science, Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[2]Department of Software Engineering, Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[3]Department of Information Technology, Government Sadiq Graduate College of Commerce, Bahawalpur 63100, Pakistan

Corresponding author: Hira Nazir (hira.nazir@iub.edu.pk)

ABSTRACT In consideration of the reduced chaotic range and susceptibility of a single chaotic map, we exploit the 4D-hyperchaotic system for creating three S-boxes i.e., red, green and blue S-boxes and a logistic map to transform a plain image into DNA strands. Afterwards, a logistic map based fake image is also generated which is also mapped to deoxyribonucleic acid (DNA) strands. Then DNA operations based on logistic map sequence are performed among the DNA strands and the resultant strands are decoded. The decoded strands are substituted by three substitution-boxes (s-boxes) to create an encrypted image. In this research, a cryptanalysis driven design approach is used to prove the security of a proposed encryption scheme. The proposed scheme operates on numerous image dimensions N $\times$ M and different image file sizes and formats. Experimental results and analysis are completed for visual analysis, key space, key sensitivity, energy analysis, homogeneity analysis, contrast analysis, entropy analysis, histogram analysis, correlation analysis, chosen-plaintext attacks, number of pixels change rate (NPCR), universal average changing intensity (UACI), mean absolute error, robustness against noises and occlusion attacks and encryption efficiency analysis. The visual as well as numerical simulations demonstrate that the proposed algorithm is safe and reliable.

INDEX TERMS Pixels block scrambling, fake image, hyperchaos, substitution, randomness.

I. INTRODUCTION

Security has become the dire need of today's age to secure the user's private and confidential data from malicious attacks. At present, many algorithms are applicable for ensuring high level of information security of gray and color medical images, such as watermarking [1], [2] steganography [3], [4] and encryption schemes [5]. Growing research in securing image data using various encryption techniques based on chaotic systems, S-box and Deoxyribo Nucleic Acid (DNA) perform reasonably well but they still possess weaknesses and deficiencies in terms of encryption efficiency, security level and computational complexity. To maintain an equilibrium between good security level and reasonably well computational complexity is gaining attention and becoming a challenging issue among computer science researchers. Thus, novel and efficient digital image encryption schemes can outstandingly boost the encryption efficiency with a high level of security.

The significance of information security using chaotic systems jointed with DNA is increasing with digitization and has gained increasing attention among scholars owing to excellent results in security. The importance of S-box is being neglected by various researchers in designing the algorithms [6]. S-box is an efficient look up table which obfuscates the relationship between a ciphered image and a secret key. A good s-box must have higher efficiency, minimum delay, zero or negative correlation among the s-box values, bijectivity, non-linearity, completeness, and avalanche criterion. The s-box and the features of chaotic system such as ergodicity, high sensitivity to initial conditions, long term unpredictability and random behavior can enhance the level of confusion and diffusion in securing plain images. Also, converting the problem or image data into DNA and using DNA's massive parallelism property and extra ordinary storage i.e., in a small volume we can have 10^{12} to 10^{13} molecules [7], can decrease the computational

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Chi Chen.

complexity and can increase the efficiency of algorithms to great extent in the future generations of computers. Therefore, we can use chaotic maps [8], [9], [18]–[20], [10]–[17], s-boxes [21]–[34] and DNA and its encoding rules [35], [36] to steer its usage in the present generation of computers as well as in the future generations of computer to encrypt the sensitive image data before its transference over an IoT network. Moreover, strong encryption/ decryption keys based on genetic databases can be produced by exploiting the DNA recombination, fragmentation, hybridization, sequencing and conservative site specific recombination (CSSR) operations [37].

Multiple and independent pixels' randomization within the image stays as the fundamental process for securing images. To this end, breadth first search and dynamic diffusion based on hyperchaotic system are combined for increasing the level of confusion and diffusion while encrypting the image [38]. Other than processing images row-wise and column-wise, diagonal-wise is also a possibility. The resistance against chosen-plaintext attack, differential attack (DA), and known-plaintext attack (KPA) is evaluated by cryptanalyzing a recent lightweight image encryption-approach for embedded systems grounded on continuous 3rd order sine hyperbolic chaotic map. The proposed approach was ascertained to be weak against the differential attack (DA) with only two chosen plain images and against chosen-plaintext attack (CPA) with only one chosen plain image having dimensions (3×400) [39]. Likewise, the authors in [40], disclosed the security weaknesses in the recently proposed encryption approach [41] which ignores the confusion phenomenon and is based on diffusions only phenomenon by using conditional shift algorithm, and 2-D chaotic map. The key was retrieved by applying the chosen-plaintext attack and one known plaintext-ciphertext pair with little computation.

To date, hyperchaotic maps having more than five dimensions have rarely been used in the encryption algorithms. The higher dimensional hyperchaotic maps have more randomness and show complex dynamic behavior. The authors in [42], adopted 7D hyperchaotic map to generate a good hyperchaotic sequence. Although it improves the randomness of the cryptosystem by exploiting a new scrambling procedure i.e., it permutes rows and columns simultaneously but it increases encryption time i.e., it takes 0.51 seconds to encrypt a grayscale image which is less than the recent state of the art algorithms. A highly secured encryption was proposed by [43], which combines two chaotic systems and four cryptographic phases namely diffusion based on XOR, substitution based on S-boxes, diffusion based on a chaotic map and block permutation for reinforcing the statistical results. The author in [11] used two generators namely bit permutation generator and bit diffusion generator which rely on SHA-256 has yielded promising encryption results. Although, we may use the single higher dimensional hyperchaotic map or than two chaotic maps or single higher dimensional hyperchaotic map with a chaotic map, the challenging issue is to improve efficiency in terms of security analysis and computational speed by maintaining a good balance between security level and computational speed. Some of the deficiencies in the proposed schemes mentioned above can be stated as:

1) Many of the color image encryption schemes exploit limited chaotic range.

2) Encryption efficiency is found missing in most of the recent works.

We describe the methodology to overcome the above mentioned deficiencies. The major contributions of this research are as follows: 1- Compared with the existing schemes, the proposed encryption scheme employs 4D-hyperchaotic based R, G and B S-boxes. 2- We employ SHA-512 of plain image and fake image with the 128-bit passcode for the generation of initial conditions of 4D-hyperchaotic system. 3- Logistic map is used for selecting the DNA encodings, and DNA operations. The positions of nitrogen bases of DNA are also mixed by applying different kinds of shifts.

The rest of the paper is organized as follows. Section 2 gives the related work. Section 3 presents our proposed algorithm. Section 4 contains the security analysis results when our algorithm is applied on some reference images. Section 5 grasps our final conclusions and directions for future research.

II. RELATED WORK

In this section, we will review the schemes related to our proposed scheme. We classify encryption schemes into three main classes: (1) schemes that apply high-dimensional chaotic systems to the color image data, (2) schemes that combine high dimensional chaotic systems with digital DNA operations and SHA, and (3) schemes that use s-box with schemes (1) or (2). Therefore, the scope of our related work is confined to the above mentioned classifications. Chaotic systems have applications in ecology, biology, cryptography, robotics, communication systems etc. A higher dimensional non-linear system must be checked whether it generates chaos or not before applying it to encrypt the data [44].

Chaotic systems are non-linear ordinary differential equations having time derivatives and are indicated by a sensitive dependence on initial states, pseudo randomness, aperiodicity, and ergodicity [45]. On the contrary, hyperchaotic systems possesses more complex dynamical behavior, larger key space, and enhanced randomness. Mathematically, chaotic and hyperchaotic systems can be categorized by calculating lyapunov exponents (LEs). A non-linear differential equation is considered chaotic if it has only one positive LE whereas, non-linear differential equation having two or more positive LEs is considered a hyperchaotic system. LE can be computed by using (1) and is defined as: the average logarithmic rate of separation or convergence between the two points on the orbits at time series t. Briefly, *LE* is the exponential separation rate for two nearby trajectories of a dynamical system [46] and can be computed by (1).

$$LE = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \ln \left| \frac{\Delta D_i}{\Delta D_o} \right|, \tag{1}$$

where ΔD_o is the initial difference between the two initial conditions X_o and Y_o. If the non-linear system has two or more Positive Lyapunov Exponents (PLEs), it is called as hyper chaos and they show much-complicated behavior as compared to chaotic systems.

Higher dimensional chaotic systems have shown the remarkable performance and cryptanalysis results [5], [47], [48]. For example, a color image encryption scheme proposed by [49], first analyses the chaotic properties (phase diagram, bifurcation diagram and Lyapunov Exponent) of 4D hyperchaotic memristive map, then it is applied for the encryption of color images. A novel substitution box based on Gaussian distributions method which achieves maximum non linearity score equivalent to Advanced Encryption Standard (AES) is proposed in [50] and passes S-box security evaluation criteria and differential analysis. But the size of the color images is not specified while applying the S-box for encryption and also gray images of only 125 gray values are encrypted in this scheme. Similarly, another chaotic S-box for wireless sensor networks [24] which is based on compound chaotic sequence and sinusoidal chaotic sequence followed by Linear Congruence Generator (LCG) and finally Baker map is used for image scrambling. It gives the higher security with lower resource consumption. Another approach for the generation of S-boxes depends on chaotic entropy source is described in [30] in which more than 20,000 S-boxes are generated. In this approach four different discrete time chaotic systems are used as an entropy source for filling the empty S-boxes with unique values. The values of initial conditions and control parameters for discrete time chaotic systems are based on optimization algorithms. Another hyperchaotic based encryption scheme that resists statistical and differential attacks with larger key space, faster encryption speed and entropy value closer to 8 is presented in [51] in which R, G and B components' pixels are permuted and diffused by following the hyperchaotic sequences. The above mentioned algorithm can be cracked if applied to gray images. In order to minimize the security risks that exist in low dimensional chaotic systems, a four dimensional hyperchaotic system followed by SHA 384 hash of plain image, external keys and dynamic DNA encodings is introduced by [52]. Although this scheme resists all kinds of brute force attacks, has high key sensitivity but more formatting operations in encoding and decoding takes more time.

Various improvements in permutation-substitution architecture and permutation only architecture have been done after the classical chaos based architecture of permutation-substitution proposed by [53]. Encryption architecture is a way or a method of doing substitutions, permutations and transformations in the encryption algorithms by exploiting an encryption key. The key can be generated from chaotic or hyperchaotic system. While encrypting an image whether color or gray, a good encryption algorithm strictly follows the principles of confusion and diffusion. Encryption architectures include substitution-permutation network (SPN),

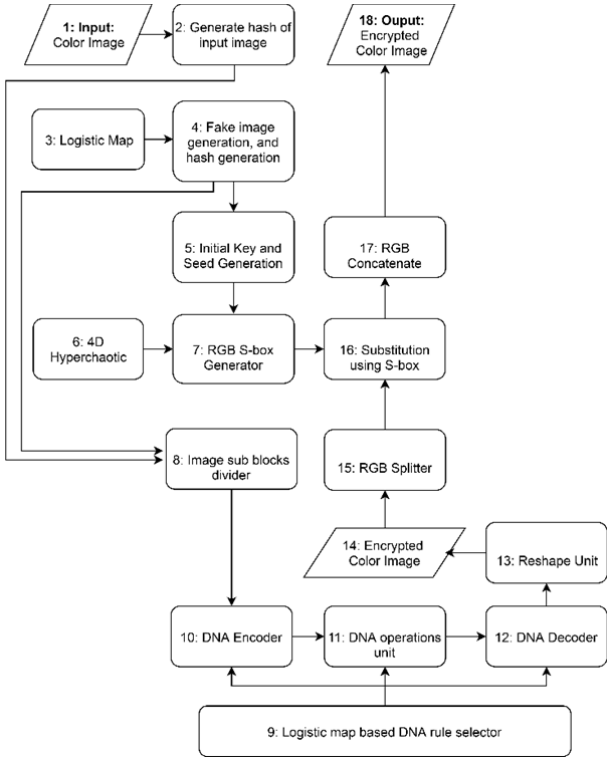


FIGURE 1. A flowchart of SbDCE-Encryption algorithm.

Feistel Network (FN), Generalized Feistel Structure (GFS) and its different variants. For example, a key substitution architecture proposed by [54] yields satisfactory encryption performance, better security and computational efficiency. It includes one round of key scheming and novel substitution method. The key scheming phase is based on logistic chaotic map whose initial conditions are calculated from weighted summation method. And the substitution is composed of random grouping, chaotic S-box construction and random substitution. Similarly, Chaotic S-box based substitution proposed by [55], provides better information entropy, Number of Pixels Changing Rate (NPCR), Universal Average Changing Intensity (UACI) and correlation coefficients results and increases the immunity against plaintext attacks (PTAs).

III. PROPOSED APPROACH

We propose an encryption algorithm that aims at improving the information security and keeping computational complexity at reasonable level. The proposed encryption algorithm, hereafter called SbDCE (S-box based DNA and Chaotic Encryption) is composed of four parts: A. Generation of initial key, chaotic image and S-box, B. Dividing the original and chaotic image into n sub blocks, C. Dynamic DNA encodings, DNA based operations and decoding, D. S-box implementation. The flow diagram of the proposed encryption algorithm is shown in the Fig. 1. The SbDCD is the decryption algorithm, which is applied on an encrypted

image to produce the decrypted image. The 4D hyperchaotic system [56] exploited in the encryption process is given in (2):

$$\begin{aligned} \dot{x} &= a\,(y - x)\,, \\ \dot{y} &= cx - dy - xz, \\ \dot{z} &= -bz + xy + w, \\ \dot{w} &= -rw + kz. \end{aligned} \tag{2}$$

When the control parameters $a = 21.7$, $b = 7.3$, $c = 6.6$, $d = -2$, $r = 0.1$ and $k = -9.5$, having the initial conditions x, y, z and w then the system (2) exhibits hyperchaotic behavior with two positive Lyapunov exponents. The initial conditions are computed in section III-A. The Logistic map is also associated in the encryption process and is given in (3):

$$x_{n+1} = rx_n(1 - x_n). \tag{3}$$

When $2.75 < c_1 \leq 3.4$, $2.75 < c_2 \leq 3.45$, $0.15 < c_3 \leq 0.21$, $0.13 < c_4 \leq 0.15$ and $x_{(i)}y_{(i)} \in [0, 1]$ then the Logistic map is in a chaotic state.

The coefficients and the initial values of (2) and (3) are exploited to produce pseudo random sequences for image encryption and decryption with the precision of 10^{-15} each. The specific encryption steps are given in the sub-sections 3A, 3B, and 3C.

In the Fig. 1, a hash is generated by using SHA-512. A chaotic image $\left(\boldsymbol{I^c}\right)$ is produced by using Logistic map to produce key $\boldsymbol{k_o}$. The size of $\left(\boldsymbol{I^c}\right)$ is kept equal to the size of plain image ($\boldsymbol{I^p}$). By hashing $\left(\boldsymbol{I^c}\right)$ with SHA-512, we get $\boldsymbol{k_1}$. The keys $\boldsymbol{k_o}$ and $\boldsymbol{k_1}$ are concatenated and are passed through SHA-512 to generate another key called as $(\boldsymbol{kI})$. Now initial conditions and seeds for Eq. (1) are produced by using $\boldsymbol{kI}$. After this, RGB S-boxes based on the Eq. (1) are created. Now decompose $\boldsymbol{I^p}$ and $\boldsymbol{I^c}$ into $\boldsymbol{n}$ sub-blocks. These sub blocks are now passed through dynamic DNA encoder, DNA operations, DNA decoder, and reshape unit to produce 1st stage encrypted image. Note that, the DNA encoding and operations are based on the values produced Logistic map. And finally, the substitution of 1st stage encrypted image is done by using RGB S-boxes to produce the encrypted color image.

A. GENERATION OF INITIAL KEY, FAKE IMAGE AND S-BOX

Step 1: Take the plain image (I^p) and hash it with SHA-512 to generate the original key k_o.

$$k_o = f_{SHA-512}\left(I^p\right) \tag{4}$$

Step 2: Set $r_1 = 3.99$. Initial parameter x_1 of the (3) can be computed as:

$$x_1 = \frac{sum\,(R, G)}{p_{max} \times size(I^p)} \tag{5}$$

where $sum\,(R, G)$ is the sum of pixel values of red and green channel, p_{max} is the maximum pixel value and $size(I^p)$ is the size of plain image.

Step 2.1: Generate the fake image (I^c) by using (3). The size of (I^c) is kept equal to the size of I^p. By hashing (I^c) with SHA-512 we get k_1.

$$k_1 = f_{SHA-512}\left(I^c\right) \tag{6}$$

Step 3: Now combine the k_o and k_1 and hash it with SHA-512 again to get initial key (kI).

$$kI = f_{SHA-512}k_1||k_0 \tag{7}$$

Step 4: Now obtain the initial parameters for (2) [57] by using (7).

Step 4.1: Firstly, divide the 512 bit of kI into 64 blocks $(b_0, b_1 \ldots b_{63})$ each of size 8 bits. Now divide 64 blocks into 16 groups $(g_0, g_1 \ldots g_{15})$, where each group will consist of 4 blocks. For example, $g_0 = \{b_0, b_1, b_2, b_3\}$, $g_1 = \{b_4, b_5, b_6, b_7\} \ldots g_{15} = \{b_{60}, b_{61}, b_{62}, b_{63}\}$.

Step 4.2: Calculate 16 random seeds from 16 groups. For example seed 1 (s_1) can be calculated as:

$$s_1 = \sum_{i=0}^{15} \frac{g_i}{PC} \tag{8}$$

where PC denotes the 128-bit pass code got by the true random noise or it can be entered by the user. Similarly, the remaining seeds $s_2, s_3 \ldots s_{16}$ can be calculated by using the above formula and same PC.

Step 4.3: Now we can use the seeds of step 4.1 to generate the initial parameters x_1, y_1, z_1, w_1 for the 4D hyperchaotic system (2). The initial parameters are computes as $x_1 = s_1 + s_2 + s_3 + s_4$, $y_1 = s_5 + s_6 + s_7 + s_8$, $z_1 = s_9 + s_{10} + s_{11} + s_{12}$, and $w_1 = s_{13} + s_{14} + s_{15} + s_{16}$.

Step 5: In this step three S-boxes (Red, Green and Blue) are generated. Following are the steps of S-box generation:

Step 5.1: Set $S-box\,(k) = k, k \in [0-255]$. Here, $S-box$, denotes the substitution box.

Step 5.2: (2) is iterated 300 times by using the initial conditions computed in step 4.3 with $a = 21.7$, $b = 7.3$, $c = 6.6$, $d = -2$, $r = 0.1$ and $k = -9.5$ to obtain four double precision sequences i.e., $X = \left[x_j\right]_{j=1}^{300}$, $Y = \left[y_j\right]_{j=1}^{300}$, $Z = \left[z_j\right]_{j=1}^{300}$ and $W = \left[w_j\right]_{j=1}^{300}$.

Step 5.3: Generate a temporary supporting sequence $TSS = \left[tss_j\right]_{j=1}^{300}$ by using (7):

$$tss_j = fix((x_j + y_j + z_j + w_j) \times g), \tag{9}$$

where, g represents gain factor. The gain factor $g \in \left\{10^{14}, 10^{15}, 10^{16}\right\}$ is used to get best randomness performance.

Step 5.4: The pseudo-code for s-box generation is presented below:

```
1.  i = 256, j = 0
2.  while i ≥ 2
3.  j = j + 1
4.    if mod (tss, i) ≠ 0
5.    swap (S − box (i), S − box (mod (tss_j, i)))
6.    i = i − 1
7.  end
8.  end
```

TABLE 1. Eight kinds of DNA mapping rules [35].

	1	**2**	**3**	**4**	5	**6**	**7**	**8**
A	00	00	11	11	01	10	01	10
G	11	11	00	00	10	01	10	01
C	10	01	10	01	00	00	11	11
T	01	10	01	10	11	11	00	00

TABLE 2. DNA based operations [35].

Addition						Multiplication				
+	**A**	**G**	**C**	**T**		×	**A**	**G**	**C**	**T**
A	A	G	C	T		**A**	T	C	G	A
G	G	C	T	A		**G**	C	T	A	G
C	C	T	A	G		**C**	G	A	T	C
T	T	A	G	C		**T**	A	G	C	T
Subtraction						Left circular shift				
−	**A**	**G**	**C**	**T**		≪	**A**	**G**	**C**	**T**
A	A	T	C	G		**A**	A	G	C	T
G	G	A	T	C		**G**	G	A	T	C
C	C	G	A	T		**C**	C	T	G	A
T	T	C	G	A		**T**	T	C	A	G
XOR						Right circular shift				
⊕	**A**	**G**	**C**	**T**		≫	**A**	**G**	**C**	**T**
A	A	G	C	T		**A**	A	G	C	T
G	G	A	T	C		**G**	G	A	T	C
C	C	T	A	G		**C**	T	C	A	G
T	A	C	G	A		**T**	C	T	G	A

Above mentioned pseudo-code is repeated by using different initial conditions of (2) to generate 3 s-boxes (red, green, and blue) each of size 16 × 16. Refer to (8), input the different 128-bit PC for modifying the initial conditions.

B. DYNAMIC DNA ENCODING, DNA OPERATIONS AND DECODING

Step 1: We decompose I^p and I^c into n sub-blocks such that total number of pixels of the all the sub-blocks of I^p or I^c will be equal to the total number of pixels of I^p or I^c. The size of each sub-block is divisible by n.

Step 2: All the sub-blocks I^p and I^c are dynamically converted into DNA sequences according to the rules of Table 1. For dynamic DNA encodings the random sequence generated by Logistic map (R_{SL}) is transformed into integers from 1 to 8. Select the DNA mapping rules based on the integers taken from the transformed sequence and do the encodings of all the sub-blocks.

For example, the matlab code ($mod(round(R_{SL}(i)*10\hat{}4),8)+1$) will transform the R_{SL} value (0.1604) placed at index i into an integer 5. Thus, integer 5 will be used as DNA mapping rule no.5 for converting the binary data to DNA sequence (genetic code).

Step 3: DNA operations (Table 2) are performed on the sub blocks of I^p and I^c. Selection of DNA operations is also based on logistic map.

Step 4: DNA decoding is done by using the same rules as were used in encoding.

Step 5: The decoded sub blocks are reshaped to stage 1 encrypted image (I^{e1}).

C. S-BOX IMPLEMENTATION

Following are the steps to do substitutions.

Step 1: The stage 1 encrypted image I^{e1} is decomposed into red, green and blue channels.

Step 2: Following steps are carried out for S-box substitution:

Step 2.1: Select a pixel value from red, green and blue channels of I^{e1} respectively.

Step 2.2: Convert it into binary.

Step 2.3: Take first 7 bits of the binary string computed in step 2.2 to represent a row number.

Step 2.4: Left shift the pixel value of 2.1 by 2 bit and take first 7 bit for representing column number.

Step 2.5: Substitute the selected pixel value of 2.1 in the S-box by using the row number and column number generated in steps 2.3 and 2.4.

TABLE 3. An example of proposed 16 × 16 S-box generated for red channel.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	218	47	174	177	203	190	243	223	33	76	250	238	220	170	186	46
1	217	143	187	133	122	188	156	210	184	206	4	31	241	52	45	150
2	227	131	6	78	74	137	138	29	236	153	18	164	212	24	113	97
3	38	231	14	10	5	65	214	215	159	146	213	77	80	136	33	182
4	100	234	115	246	114	144	244	152	59	145	157	222	124	181	20	98
5	2	254	193	191	15	239	176	237	192	92	106	49	162	13	119	180
6	199	40	224	43	128	195	235	117	67	249	76	255	64	91	93	55
7	107	211	17	62	158	102	9	198	171	44	110	194	70	28	35	36
8	108	95	225	226	0	189	183	196	221	8	230	105	134	148	167	142
9	27	39	7	84	147	89	125	166	245	141	30	56	204	79	118	216
10	34	116	87	132	22	219	18	161	120	168	121	154	94	200	242	109
11	23	240	81	172	229	12	247	240	151	86	103	96	69	205	169	173
12	111	82	155	127	140	130	208	60	54	21	41	37	52	160	101	73
13	202	129	207	228	178	201	25	85	209	61	252	165	3	11	63	135
14	1	139	16	179	50	163	42	248	75	83	68	71	48	58	175	66
15	232	57	88	26	253	99	185	233	123	112	104	126	90	149	251	199

In this way, the red, green and blue channels of I^{e1} are substituted with Red, Green and Blue S-box values respectively. And we get the encrypted Red (eR), encrypted Green (eG) and encrypted Blue (eB) channels of an image.

Step 3: Now, (eR) , (eG) and (eB) are combined to get the final encrypted image (I^e).

To be more intuitive s-box implementation is shown in Fig. 2. Whereas, an example of proposed S-box generated for red channel based on (1) is given in Table 3. In the Fig. 2, 1st stage encrypted image is split into RGB components and are substituted with three S-boxes produced by solving (2). After substitution, three encrypted RGB components are produced. The average non-linearity and strict avalanche criterion of the proposed s-box comes out to be 111.83 and 0.4978 respectively which is better than [26], [58]–[62], and is comparable to AES [63] (112, 0.5058).

D. DECRYPTION PROCEDURE

In order to retrieve back the plain image the steps of SbDCE i.e., section IIIC, IIIB, and IIIA. are executed from bottom to top in the reverse order.

IV. RESULTS AND SECURITY ANALYSIS

The key sensitivity, statistical analysis, entropy analysis, differential analysis and fundamental performance analysis are presented in this section. All the experiments are implemented in MatlabR2015a installed on windows 8. PC specifications are: 8 GB RAM, Intel® Core i7-1165G7 (up to 4.7 GHz, 12 MB L3 cache, 4 cores) + Intel® Iris® Xe Graphics. Standard test images of Lena, Peppers, Baboon, Panda, and Female are used in the simulations. While comparing the results, only the improved results are marked with bold font.

TABLE 4. PSNR comparisons.

Image (512×512)	PSNR (plain-encrypted)	PSNR (plain-decrypted)
Lena	**8.0248** (8.1293 [65])	∞ (∞ [65])
Baboon	**8.7576** (8.7729 [65])	∞ (∞ [65])
Peppers	**7.2003** (7.6393 [65])	∞ (∞ [65])
Knee	7.5411	∞

A. VISUAL ANALYSIS

We applied encryption on four color images as shown in Fig. 3. The visual representations of encrypted images are not recognizable from the human visual system. PSNRs without addition of noise for the decrypted images given in Table 4 indicate that there are no differences among decrypted and original images.

PSNR between the original plain image ($M \times N$) and the encrypted image ($M \times N$) [64] can be computed by (10).

$$PSNR = 10log_{10}\left(\frac{MAX^2}{MSE(I^p, I^d)}\right), \tag{10}$$

where MAX is the maximum possible intensity value of a pixel in the plain color image I^p, m and n are the width and height of a color image, I^d is the decrypted image that incorporates noise and MSE is the mean squared error and can be computed as $MSE(I^p, I^d) = \frac{\sum_{1\le i\le m}\sum_{1\le j\le n}[I^p(i,j)-I^d(i,j)]^2}{m\times n}$.

B. NIST SP800-22 TEST SUIT ANALYSIS

We have also applied NIST SP800-22 test suit [66], [12] on chaotic sequences produced by 4D-hyperchaotic system to assess the randomness. p values $> (\alpha = 0.001)$ in the Table 5 indicates that the test is passed and the proposed hyperchaotic system can be used as pseudo random number generator.

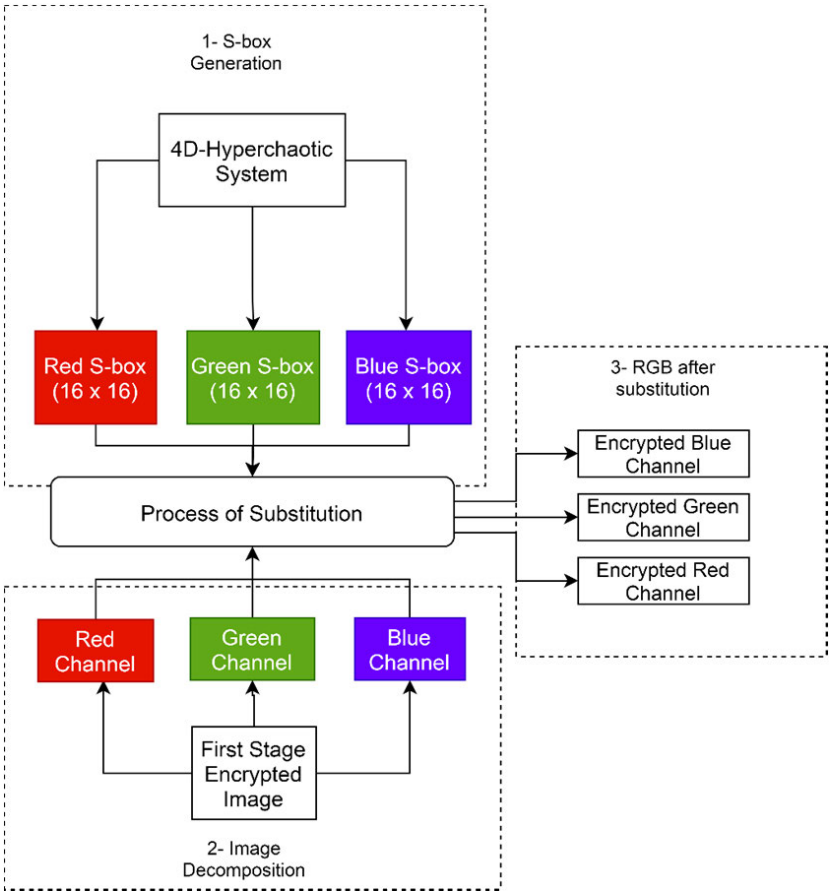


FIGURE 2. S-box implementation.

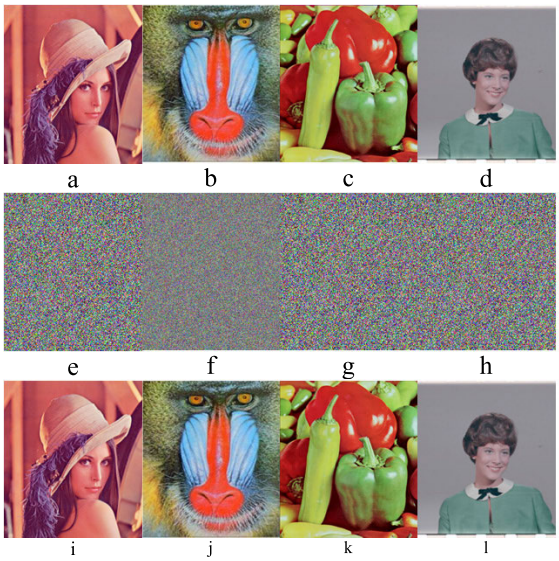


FIGURE 3. SbDCE-SbDCD performance: (a)-(d) the original images, Lena (256 × 256), Baboon (512 × 512), Peppers (256 × 256) and female-4.1.03 (256 × 256). (e)-(h) are encrypted-images, while (i)-(l) are decrypted images.

C. KEYSPACE

SbDCE is characterized by a large keyspace. A keyspace minimum of 2^{100} bit is enough to repel brute force attacks [67]. The initial values x_1, y_1, z_1, w_1 of the 4D-hyperchaotic system (1), the coefficient r_1, initial condition x_1 of (2) are used as the secret keys for SbDCE-SbDCD with floating-point precision of 10^{-15} each. We have also used the 128-bit passcode. Hence, the total key space emerges out as $(10 \wedge 15)^6 = 10^{90} \cong 2^{300}.2^{128} = 2^{428}$ which is strong enough to resist all kinds of brute force attacks [68].

TABLE 5. The NIST-800-22 randomness test results of chaotic sequence.

NIST parameters	Min p −value	Results
Frequency	0.4128	Passed
Block Frequency	0.3821	Passed
The Run test	0.5631	Passed
Longest run of Ones	0.5720	Passed
Binary Matrix Rank	0.4651	Passed
DFT Spectral	0.6225	Passed
Non-Overlapping Templates	0.4224	Passed
Overlapping- Templates	0.5923	Passed
Linear Complexity	0.6184	Passed
Serial Test	0.4422	Passed
Approximate Entropy	0.3222	Passed
Cummulative Sums	0.4850	Passed
Random Excursions	0.7652	Passed
Random Excursions Variant	0.3947	Passed

Algorithm 1 : SbDCE Encryption

Input: A plain color image (m × n), initial conditions for logistic map and 4D hyperchaotic system).
Output: An encrypted image (m × n).
1: Input the plain image $\left(I^p\right)$ and hash it with SHA-512 to generate the original key k_o. (details in section III (1).
$k_o \Longleftarrow f_{SHA-512}\left(I^p\right)$.
2: Compute initial conditions for the Logistic map (2). (details in section III (1).
$x_1 \Longleftarrow sum\,(R, G)\,/(p_{max} * size(I^p))$.
3: Generate the fake image $\left(I^c\right)$ by using (2) and hash it with SHA-512 to produce k_1. (details in section III (1).
$I^c \Longleftarrow fakeImage\,(logistic\_map\_Seq)$.
$k_1 \Longleftarrow f_{SHA-512}\left(I^c\right)$.
4: Concatenate the k_o and k_1 and hash it with SHA-512 again to get initial key (kI). (details in section III (1).

$$kI \Longleftarrow f_{SHA-512} k_1 || k_0$$

5: Now obtain the initial parameters for (2) by using kI. (details in section III A steps 4.1 to 4.3).
6: Generate hyperchaotic sequences by iterating (2) and generate 3 S-boxes (Red S-box, Green S-box and Blue S-box) each of size $\mathbf{16} \times \mathbf{16}$. (details in section III (1).
7: Decompose I^p and I^c into n sub-blocks and perform DNA encodings, DNA based operations and DNA decoding to produce decoded sub blocks. (details in Section III B).
8: Decoded sub blocks are reshaped to stage 1 encrypted image (I^{e1}). (details in section III B).
9:I^{e1} is decomposed into red, green and blue channels. (details in section III C).
10: The red, green and blue channels of I^{e1} are substituted with 3 S-boxes. (details in section III C).

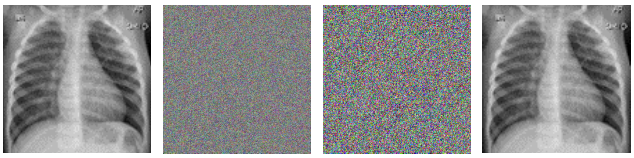
Algorithm 2 : SbDCD Decryption

Input: An encrypted image (m × n), key.
Output: Decrypted image (m × n).
Steps: In order to retrieve back the plain image the steps of SbDCE are executed from bottom to top in the reverse order.

D. KEY SENSITIVITY RESULTS

Key sensitivity means, a slight change in the secret key cannot decrypt the encrypted image. We verified the key sensitivity of SbDCE-SbDCD by applying it to (covid-19-pneumonia-paediatric.jpg) by encrypting it with correct secret key and decrypting it with a minor change in a secret key. The visual evidences shown in the Fig. 4 obviously indicate the absence of relation between the original image and decrypted image.

We symbolize the original image by I, the secret keys by $SK^1 = sk_0^1, sk_1^1, \ldots, sk_{MN-1}^1, SK^2 = sk_0^2, sk_1^2, \ldots, sk_{MN-1}^2$ and the encrypted image by $E^1 = e_0^1, e_1^1, \ldots, e_{MN-1}^1, E^2 = e_0^2, e_1^2, \ldots, e_{MN-1}^2$. Key sensitivity by hamming distance



(a) (b) (c) (d)

FIGURE 4. Key sensitivity test for the image (covid-19-pneumonia-paediatric.jpg): (a) the plain image, (b) the encrypted image, (c) the decrypted image with different initial conditions, (d) the decrypted image with same initial conditions.

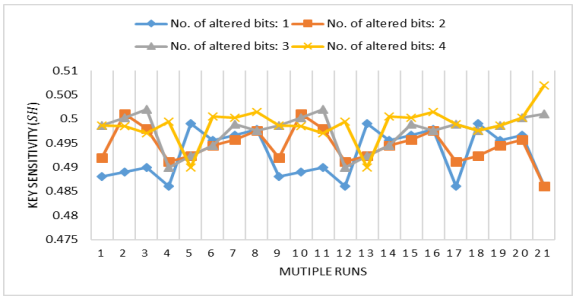


FIGURE 5. Key sensitivity analysis.

(SH) given in (11) is computed as [69]

$$SH = \frac{1}{MN} \sum_{j=0}^{MN-1} \left(E_j^1 \oplus E_j^2\right), \tag{11}$$

where E^1 and E^2 are given by:

$$\begin{cases} E^1 = encrypt\left(I, SK^1\right) \\ E^2 = encrypt\left(I, SK^2\right). \end{cases}$$

A value of 0.5 for SH indicates a worthy cipher [69]. SK^1 and SK^2 have n-bit difference. Fig. 5 visualizes the SH test for the SbDCE. In this experiment, lena color images are encrypted for hundred iterations by varying the n-bits of SK^2. We note that on average 99.1% of SH values be found in $[0.48 - 0.501]$ range which is closer to 0.5 thus demonstrating the proposed scheme has higher key sensitivity. Moreover, the key sensitivity is also tested by computing the average NPCR for the images shown in Fig. 4 (a-d). The average NPCR comes out to be 99.49%.

E. DIFFERENTIAL ATTACK ANALYSIS

In this test, the effect of 1-bit change in the original plain image to the resultant encrypted image is evaluated. The tests such as, Number of Pixel Changing Rate (NPCR) and Unified Average Changing Intensity (UACI) are commonly used to assess the differential attack [70]. NPCR and UACI [71] are computed by (12) and (13).

$$NPCR\left(I^e, I^{e*}\right) = \frac{\sum_{1 \le i \le m} d\,(i,j)}{\substack{1 \le j \le n \\ T}} \times 100, \tag{12}$$

where

$$d\,(i,j) = \begin{cases} 1 & if\ I^e\,(i,j) = I^{e*}\,(i,j) \\ 0 & if\ I^e\,(i,j) \neq I^{e*}\,(i,j), \end{cases}$$

TABLE 6. NPCR and UACI results.

Images (512×512)	NPCR				UACI			
	Min	**Max**	**Avg. (single run)**	**Avg. (several runs)**	**Min**	**Max**	**Avg. (single run)**	**Avg. (several runs)**
Lena	99.6214	99.6596	**99.6294** (99.6 [76])	99.62 (99.6314 [8], 99.6177 [74], 52.8244 [73], 99.59[72], 99.69 [77])	33.3214	33.4252	**33.3514** (33.3 [76])	33.5576 (33.5513 [8], 33.6694 [74], 20.6665 [73], 33.50[72], 33.51 [77])
Panda	99.6231	99.6345	99.6220	-	33.3802	33.4965	33.4518	-
Baboon	99.5964	99.6478	99.6278	-	33.3934	33.4142	33.3956	-
Peppers	99.5676	99.6543	99.6287	-	33.4222	33.4587	33.4316	-

TABLE 7. Correlation coefficients results.

Images (512×512)	Plain			Encrypted		
	H	**V**	**D**	**H**	**V**	D
Lena	**0.9445** (0.974662 [78], 0.9778 [8])	**0.9328** (0.985274 [78], 0.9886 [8])	**0.9082** (0.964243 [78], 0.9878 [8])	**-0.0447** (0.0042 [77], 0.000946 [78], 0.0031 [8])	0.00037 (-0.0021 [77], 0.000844 [78], 0.0005 [8])	**-0.0047** (-0.0043 [77], 0.002741 [78], -0.0041 [8])
Panda	0.9458	0.9458	0.9458	-0.0357	-0.0357	-0.0223
Baboon	0.9458	0.9458	0.9458	-0.0357	-0.0223	-0.0223
Pepper	**0.9658** (0.9654 [79])	0.9458 (0.9552 [79])	**0.9458** (0.9243 [79])	**-0.0223** (-0.0012 [79])	**-0.0223** (-0.0213 [79])	**-0.0223** (0.0027 [79])

and

$$UACI\left(I^{e}, I^{e*}\right) = \frac{1}{T}\left[\frac{\sum_{\substack{1\le i\le m\\ 1\le j\le n}} |I^{e}(i,j) - I^{e*}(i,j)|}{LSP}\right] \times 100, \tag{13}$$

where T symbolizes total number of pixels in an image, LSP denotes largest supported pixel value (i.e., 255) and I^{e}, I^{e*} are the two encrypted versions resulted from two plain images i.e., plain image without a single-bit change and plain image with 1-bit change. NPCR and UACI outcomes are shown in Table 6. The average values several runs (NPCR) of the proposed algorithm, are better than [72], [73], [74] and the average values several runs (UACI) of the proposed algorithm, are better than [8], [72], [73] which obviously indicate the resistance against the differential attacks.

F. CORRELATION ANALYSIS

There is a positive correlation either in vertical, horizontal or diagonal direction among the adjacent pixels of an original plain image. Correlation among adjacent pixels is diminished or closer to 0, in an efficient encryption scheme [8], [72]. We chose 5000 pairs of two adjacent pixels of plain and an encrypted image and computed the correlation coefficients in horizontal, vertical and diagonal directions. Correlations plots are depicted in Fig. 6, while correlation coefficient results are written in Table 7. The comparison results clearly reveal that the encrypted images have no correlation among adjacent pixels, so SbDCE has a strong strength against statistical attacks.

TABLE 8. Chi-square test applied to the images (Fig.3(e-h)).

Image	χ^2-test	Remarks
Fig. 3(e) (**512 × 512**)	257.912	Pass
Fig. 3(f) (**512 × 512**)	232.970	Pass
Fig. 3(g) (**512 × 512**)	254.816	Pass
Fig. 3(h) (**512 × 512**)	264.145	Pass

G. HISTOGRAM ANALYSIS

Histogram analysis deals with the dispersion of pixel intensity values in an entire image (plain or encrypted). Histograms of the encrypted images are balanced in comparison with the original plain images. Histograms for R, G and B channels for the images shown in Fig.3 (a-d) are shown in Fig. 7. The histograms uniformity verified through chi-square test is reported in Table 8. Pearson's chi-square (χ^2) goodness of fit statistic [75] for categorical data of encrypted image's histogram can be computed by (14).

$$\chi^2 = \sum_{i=0}^{mp}\left(\frac{(O_i - E_i)^2}{E_i}\right). \tag{14}$$

In (14), mp signifies the maximum pixel intensity i.e., 255. O_i represents the observed frequency count for the pixel at index i in the histogram. Similarly, E_i represents the expected frequency count which is same at every index i. And it can be calculated as $M \times N \times 3/Tbins$. Here, $M \times N \times 3$ represents total number of pixels of RGB image and $Tbins$ represents

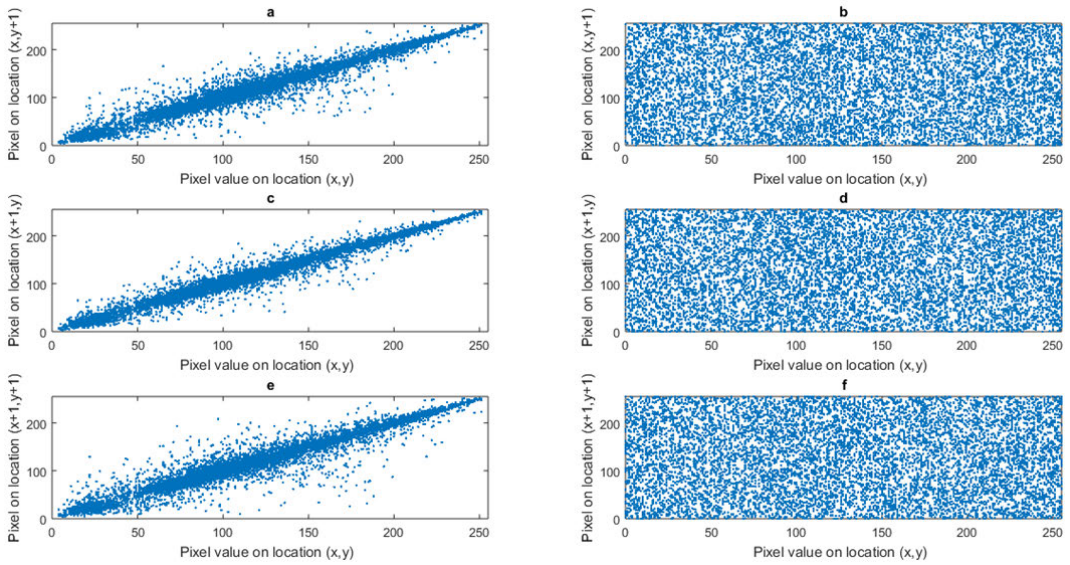


FIGURE 6. Correlation plots for image Lena. (a,c,e) represent correlation plots of the plain image Lena, whereas, (b,d,f) represent correlation plots of encrypted image of Lena.

TABLE 9. Histogram variance.

Image	Encrypted Image Components			
	R	G	B	Avg. (R,G,B)
Lena (256×256)	1047.3	1001.2	1003.5	1017.33 (977.02 [80], 1079.20 [18])
Panda (256×256)	494.98	392.92	668.69	518.86
Baboon (512×512)	1,097.3	985.92	1130.8	1,071.34
Peppers (512×512)	**974.16** (1077 [8])	**1051.3** (1059.6 [8])	**1040.20**, (1061.4 [8])	**1021.88** (1066 [8])

the total number of bins i.e., 256 for 8-bit image. The chi-square test score for the histogram of encrypted image is acceptable if it is smaller than $\chi^2_{th}(255, 0.05 = 293.2478)$. The computed values of chi-square in Table 8 are less than 293.2478. Even so not enough, but still SbDCE provides resistance against statistical attacks.

Additionally, the histogram's uniform distribution can be quantified by computing the variance, i.e., low variance denotes higher histogram uniformity and high variance designates lower histogram's uniformity. The variances for Peppers and Lena (512 × 512) images presented in Table 9, are lower than [8], and [10], thus showing higher uniformity in the histogram of the encrypted images.

H. GLCM ANALYSIS

It is an image texture analysis technique i.e., how often dissimilar combinations of gray levels in a specified spatial relationship occur in an image section or in an entire image. The measurements for instance contrast, energy and homogeneity can be derived from the normalized GLCM [81], [82].

In contrast analysis, the intensity contrast between a pixel and its neighbor pixels over the entire image is computed by using the following equation:

$$Cn = \sum_{i,j}^{gl-1} |i-j|^2 G_{i,j}, \tag{15}$$

where i, j are the spatial coordinates, $G_{i,j}$ symbolizes the GLCM of an encrypted image and gl represents the gray level. The higher the Cn is, the higher the security of SbDCE.

The energy in GLCM represents the uniformity and gives the sum of squared elements in the GLCM and can be calculated as:

$$En = \sum_{i,j}^{gl-1} G_{i,j}^2, \tag{16}$$

where i, j are the spatial coordinates, $G_{i,j}$ is the GLCM of an encrypted image and gl is the gray tone. The lower the En value is, the higher the encryption quality of SbDCE.

The homogeneity measures the closeness of the distribution of elements in the GLCM and can be calculated as:

$$Hm = \sum_{i,j}^{gl-1} \frac{G_{i,j}}{1+|i-j|}, \tag{17}$$

where i, j are the spatial coordinates, $G_{i,j}$ is the GLCM of an encrypted image and gl is the gray tone. The lower the Hm is, the higher the encryption quality of SbDCE. Cn, En and Hm results are given in Table 10. Improved results are written in bold font.

I. INFORMATION ENTROPY ANALYSIS

Entropy of an encrypted image reflects the pixel values randomness in an encrypted image. A value nearer to 8 (for

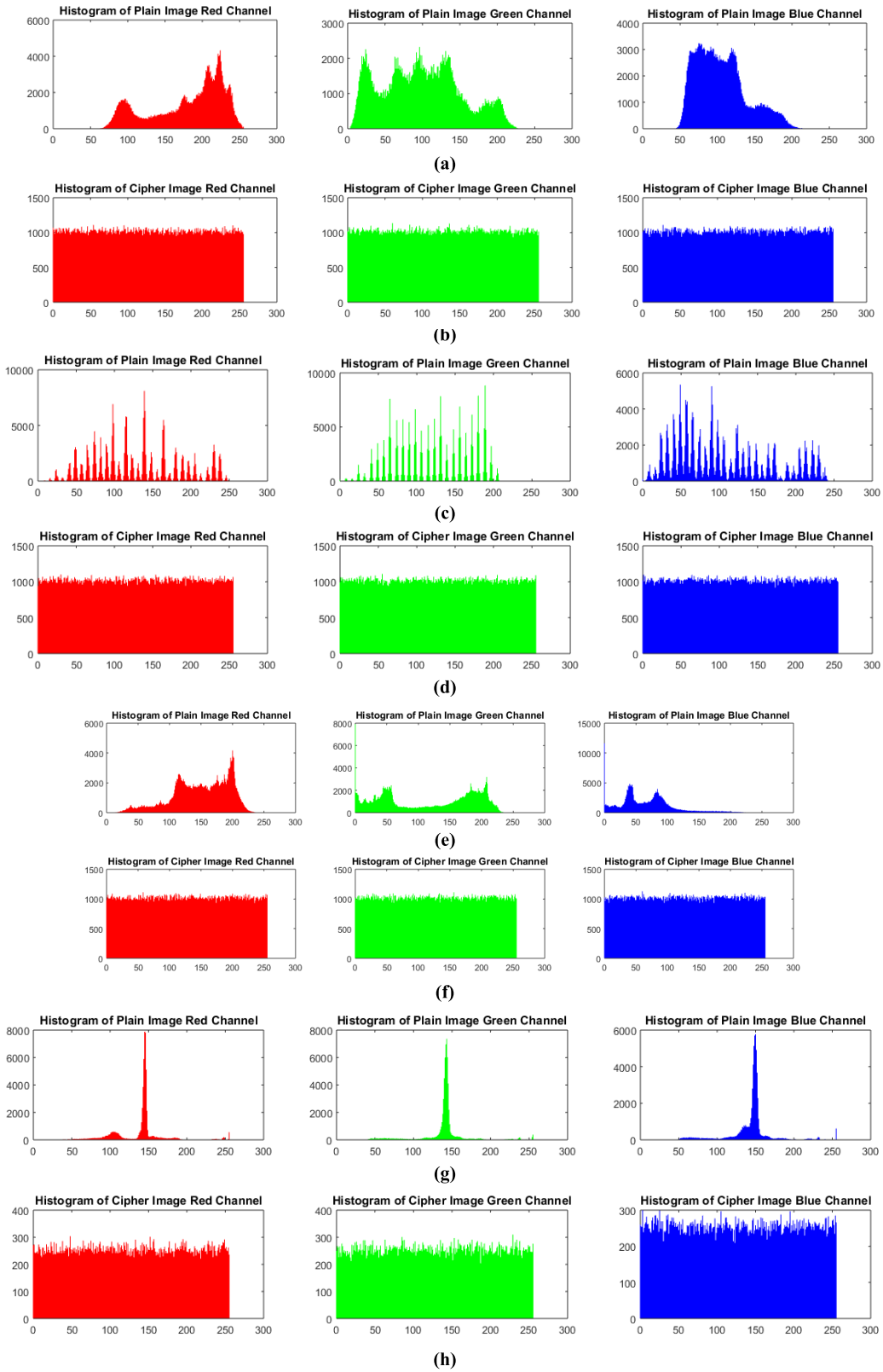


FIGURE 7. RGB Histograms: (a) RGB histograms for plain image Lena (Fig. 3 (a)), (b) RGB histograms for encrypted image Lena (Fig. 3 (e)), (c) RGB histograms for plain image Baboon (Fig. 3 (b)), (d) RGB histograms for encrypted image Baboon (Fig. 3 (f)), (e) RGB histograms for plain image Peppers (Fig. 3 (c)), (f) RGB histograms for encrypted image Peppers (Fig. 3 (c)), (g) RGB histograms for plain image Female (Fig. 3 (d)), (h) RGB histograms for encrypted image Female (Fig. 3 (h)).

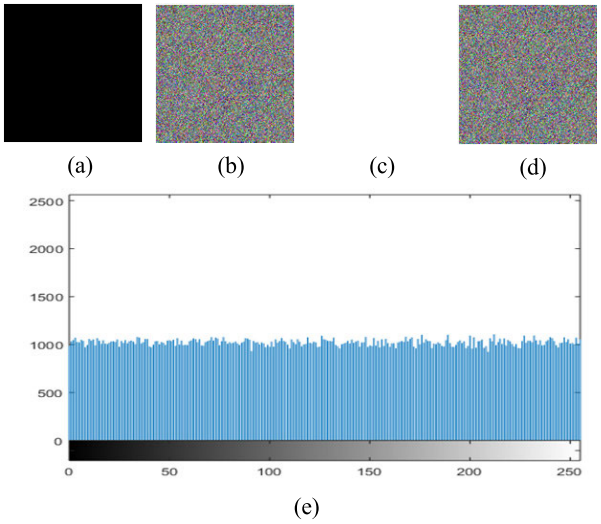


FIGURE 8. Simulation of CPA: (a) RGB-black image, (b) encrypted RGB-black image, (c) RGB-white image, (d) encrypted RGB-white image, (e) histogram of image (b).

TABLE 10. Contrast, energy and homogeneity comparisons.

Plain Color Image (512×512)	GLCM Based Analysis	Encrypted Color Image (512×512)	GLCM Based Analysis	Average
Lena	Cn=0.2739 En=0.1411 Hm=0.8848	Lena	Cn=10.515347 En=0.015629 Hm=0.389141	Cn=10.4963 (10.48 [17], 10.1098 [81], 10.1145 [83]. En=**0.015628** (0.0155 [17], 0.165 [81]). Hm=**0.3895** (0.3893 [17], 0.4110 [81])
Peppers	Cn=0.1333 En=0.1726 Hm=0.9500	Peppers	Cn=10.482891 En=0.015629 Hm=0.389860	
Baboon	Cn=0.8004 En=0.0651 Hm=0.7575	Baboon	Cn=10.490675 En=0.015628 Hm=0.389631	

an 8-bit image), signifies the maximum randomness. The entropy$E$ of an encrypted image I^e is calculated as [84]:

$$E\left(I^e\right) = -\sum_{i=0}^{2^L-1} p\left(\left(I_i^e\right)\right) log_2\left(p\left(\left(I_i^e\right)\right)\right), \qquad (18)$$

where I_i^e is the ith intensity value of an encrypted or plain gray level image, p is the probability function of each gray level count and L denotes the number of gray levels. Entropy results of the images Lena, Peppers, Baboon, and Splash each of size 512 × 512 and comparisons are given in Table 11. Improved results are highlighted with bold font. Hence, the SbDCE can attain maximal randomness thus leading to information protection.

J. CHOSEN PLAINTEXT ATTACK ANALYSIS

Known-plaintext and chosen-plaintext attacks are two important attacks which are commonly used by cryptanalysts to extract some useful information [8]. If a cipher can repel the chosen-plaintext attack (CPA), it can repel other types of attacks [12]. The proposed SbDCE resists both types of attacks. For simulation of chosen-plaintext attack (see Fig. 8.), we have used RGB-black and RGB-white images whose RGB components are all black or white. The NPCR values of the Fig. 8(b) and (d) is 99.51 and 99.49% respectively. So, by observing the Fig. 8(b), (d) and (e), it is impossible for the cryptanalyst to extract any information about the plain RGB-image.

TABLE 11. Entropy comparisons.

Images (512×512)	Entropy Plain (	Entropy Encrypted
Panda	7.8028	7.9988
Lena	7.7365	**7.9989** (7.9964 [73], 7.9973 [18], 7.9975 [85])
Peppers	7.7150	**7.9996** (7.9992 [73], 7.9974 [18], 7.9972 [85], 7.9995 [77])
Baboon	7.6429	**7.9991** (7.9978 [14], 7.9971 [85])
Splash	7.2428	7.9997 (7.9998 [84])
Tiffany	6.7926	7.9997 (7.9998 [84])

TABLE 12. MAE results and the comparison with other works. Improved results are highlighted with bold font.

Image (512×512)	MAE		
	R	G	B
Lena	76.8790 (84.2492 [52], 78.201950 [21])	**79.2186** (78.5999 [52], 78.7050 [21])	**82.6461** (70.7826 [52], 77.4049 [21])
All white	127.29997 (127.7523 [52])	**127.64708** (127.2898 [52])	127.62926 (127.7881 [52])
All black	128.1602	127.7749	127.13945
Female4.1.03	**68.87440** (69.1804 [52])	**68.21820** (68.1057 [52])	**68.34314** (68.1736 [52])

K. MEAN ABSOLUTE ERROR ANALYSIS

It is a statistical parameter for evaluating how much the encrypted image I^e differs from the plain image I^p. The larger the *MAE* is, the higher the security of proposed cryptosystem. *MAE* is defined as [21]:

$$MAE(I^p, I^e) = \frac{\sum_{\substack{1\le i\le m \\ 1\le j\le n}} |I^e(i,j) - I^p(i,j)|}{m \times n} \qquad (19)$$

The computed *MAE* values of our proposed scheme and its comparison with the existing works are given in Table 12.

L. ROBUSTNESS AGAINST NOISE AND OCCLUSION ATTACK

Noise is defined as a random and uninvited form of energy or signal that contaminates the original signal and it may originate from communication medium or receiver's end. The noise in an encrypted image can produce errors in decrypted

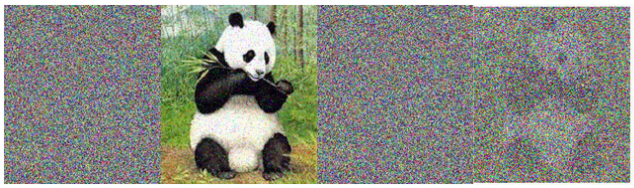


FIGURE 9. Robustness against noises added to encrypted image panda: (a) SPN density 5%, (b) decrypted image of image (a) having PSNR = 64.2137, (c) Speckle noise density 1%, (d) decrypted image of image (c) having PSNR = 18.4527.

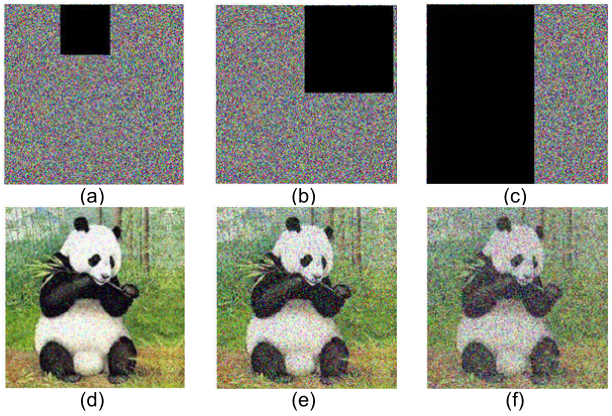


FIGURE 10. Simulation results of occlusion attacks on encrypted panda image: (a)-(c) occlusion attacks, (d)-(f) recovered images.

TABLE 13. PSNR comparisons of the image panda (256 × 256).

Noise & parameters	PSNR
SPN (0.2%)	**47.0834** (34.4133 [52])
SPN (0.5%)	**37.2037** (29.03 [27])
SPN (5%)	27.7173
Speckle (1%)	23.4837
Gaussian (0, 0.000001)	**52.2473** (28 [52])
Gaussian (0, 0.0001)	27.2021
Gaussian (0, 0.0003)	20.97

images i.e., decrypted images are difficult to be recognized from human visual system. The common noise types are salt & pepper noise (SPN), gauss noise and speckle noise etc. [86], [87].

In an occlusion attack, an encrypted image is shielded partially or completely with some fixed valued pixels. The robustness of the proposed approach against three noise types is shown in Fig. 9. Similarly, performance against occlusion attacks in three different scenarios, indicate recognizable decryption from occlusion attacks of up to 60% (see Fig. 10). Robustness of the proposed approach in terms of PSNR is also tested after applying different attacks. PSNR results are given in Table 13. The robustness against three noise types and the occlusion attacks are validated by the excellent PSNR values.

TABLE 14. Encryption quality of the color images.

Image (512×512)	Encryption Quality Our value (Ref value)
Panda	**802.43** (785.487 [90])
Lena	899.03 (789.34 [53], 910.04 [89], 663.82 [91])
Peppers	**891.34** (788.14 [65])
Baboon	**958.20** (811.14 [69], 804.113 [68], 773.90 [91])

TABLE 15. The computational complexity analysis.

Algorithm	Image	Complexity
[90]	Color	$O(168 \times m \times n)$
[51]	Color	$O(9 \times m \times n)$
[65]	Color	$O(69 \times m \times n)$
[69]	Gray	$O(124 \times m \times n)$
[68]	Gray	$O(579 \times m \times n)$
[89]	Gray	$O(108 \times m \times n + 72L^4)$
Proposed	Color	$O(16 \times m \times n)$

M. ENCRYPTION QUALITY

Pixels of plain image change after encryption. Hence, the higher the change in the image pixels, the higher the encryption quality (*EQlty*). Therefore, the encryption quality is defined as the total change in image pixels between the plain and ciphered image. It may be defined as the average number of alterations to each gray level [88]. Let $H^p_{i,j}$ represents the number of pixels having j gray levels in the i^{th} channel of the plain image and $H^e_{i,j}$ denotes the number of pixels having j gray levels in the i^{th} channel of the encrypted image. Here, i =(1,2,3) and j = (0, 1, 2, 3, . . . 255). Hence, encryption quality of a color image can be expressed as:

$$EQlty = \frac{\sum_{i=1}^{3} \sum_{j=0}^{255} \left| H^p_{i,j} - H^e_{i,j} \right|}{3 \times 256} \tag{20}$$

Encryption quality results are given in Table 14. The average number of alterations to each gray level are better in most of the cases except [89].

N. COMPUTATIONAL COMPLEXITY

No doubt, security is the prime concern of any encryption algorithm but the next weighty part is its computational complexity (CC). Our CC evaluation considers all oper- ations such as DNA encoding, DNA decoding, permutations based on chaotic sequences, s-box generation, RGB substitutions. The upper bound for large enough input sizes of the proposed algorithm comes out to be O (26 × m × n) and its comparison is given in Table 15. We observe that substitutions based on s-boxes, DNA encodings and formatting operations, DNA based operations among $\boldsymbol{n}$ sub-blocks of $\boldsymbol{I^p}$ and $\boldsymbol{I^c}$ improves the security but increases CC to some extent.

TABLE 16. Encryption efficiency analysis and comparison whenever the data is available.

No. of Color Images	Dimension	*AvgEncT* (ms)	*AvgDecT* (ms)	*AvgEncTh* (MB/s)	*AvgNCPB*
10	256×256	110 (82 [92], 267 [93], 540 [94])	109	**0.37642** (0.1471 [72])	11907.61 (25932.68 [72])
10	512×512	**190** (320 [92], 1186 [93])	191	3.21	3135.50

O. ENCRYPTION EFFICIENCY ANALYSIS

Encryption efficiency of SbDCE-SbDCD is done on the RGB images of peppers and Lena having different dimensions. The average times of encryption and decryption are noted for multiple runs. The encryption efficiency based on average encryption time, I^p file size and processor's speed can be computed as:

$$EncTh = \frac{I^p_{size}}{AvgEncT}, \tag{21}$$

$$NCPB = \frac{CPU_{speed}}{EncTh}, \tag{22}$$

where *EncTh* is the encryption throughput [72], *NCPB* is the number of CPU cycles per byte and *AvgEncT* is the average encryption time. *AvgEncT* of 10 different images (4.1.01 − 4.1.10) each of size 256 × 256 is computed. And the *AvgEncT* of 10 different images (2.1.01 − 2.1.10) each of size 512 × 512 is also computed. All the images are taken from USC-SIPI database (https://sipi.usc.edu/database/). Encryption efficiency analysis is given in Table 16. It is examined that the average encryption time *AvgEncT* of 10 different images (256 × 256) takes 110 milli-seconds (ms) and increases up to 190 ms when image dimensions are doubled but is still comparable to most of the recent state of the art color image encryption algorithms.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a symmetric block cipher in which s-box generation is based on 4D- hyperchaos whereas fake image generation, DNA encodings, DNA operations are based on logistic map. Finally, the DNA decoding is done and the decoded values are replaced with s-box values to produce the encrypted image data. Experimental results show that the proposed scheme has secure key space, improved performance in the sense of randomness, encryption quality, correlation, key sensitivity, energy, mean absolute error and robustness, while comparable to the recent works in the sense of NPCR, UACI and encryption efficiency. The proposed scheme can operate on numerous image dimensions N × M with different image sizes and formats.

In future we intend to study computational intensive operations to make it more lightweight cipher. We also intend to introduce selected encryption i.e., to encrypt faces, finger prints, and watermarks in the future version of this cipher. We also plan to re-design the proposed cryptosystem to test on the 3D medical images with several modalities.

REFERENCES

[1] N. M. Makbol, B. E. Khoo, T. H. Rassem, and K. Loukhaoukha, "A new reliable optimized image watermarking scheme based on the integer wavelet transform and singular value decomposition for copyright protection," *Inf. Sci.*, vol. 417, pp. 381–400, Nov. 2017.

[2] A. Fatahbeygi and F. A. Tab, "A highly robust and secure image watermarking based on classification and visual cryptography," *J. Inf. Secur. Appl.*, vol. 45, pp. 71–78, Apr. 2019.

[3] S. Ahani and S. Ghaemmaghami, "Colour image steganography method based on sparse representation," *IET Image Process.*, vol. 9, no. 6, pp. 496–505, Mar. 2015.

[4] P. Malathi, M. Manoaj, R. Manoj, V. Raghavan, and R. E. Vinodhini, "Highly improved DNA based steganography," *Proc. Comput. Sci.*, vol. 115, pp. 651–659, Jan. 2017.

[5] C. A. Murugan and P. KarthigaiKumar, "Survey on image encryption schemes, bio cryptography and efficient encryption algorithms," *Mobile Netw. Appl.*, 2018, doi: 10.1007/s11036-018-1058-3.

[6] C. Li, Y. Zhang, and E. Yong, "When an attacker meets a cipher-image in 2018?: A year in review," *J. Inf. Secur. Appl.*, vol. 48, pp. 1–9, Oct. 2019.

[7] D. I. Lewin, "DNA computing," *Comput. Sci. Eng.*, vol. 4, no. 3, pp. 5–8, May 2002.

[8] K. A. K. 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.

[9] H. Ghazanfaripour and A. Broumandnia, "Designing a digital image encryption scheme using chaotic maps with prime modular," *Opt. Laser Technol.*, vol. 131, Nov. 2020, Art. no. 106339.

[10] I. Yasser, M. A. Mohamed, and A. S. Samra, "A chaotic-based encryption/decryption framework for secure multimedia communications," *Entropy*, vol. 22, no. 11, pp. 1–23, 2020.

[11] T. Gopalakrishnan and S. Ramakrishnan, "Chaotic image encryption with hash keying as key generator," *IETE J. Res.*, vol. 63, no. 2, pp. 172–187, Mar. 2017.

[12] C. Zhu and K. Sun, "Cryptanalyzing and improving a novel color image encryption algorithm using RT-enhanced chaotic tent maps," *IEEE Access*, vol. 6, pp. 18759–18770, 2018.

[13] J. A. P. Artiles, D. P. B. Chaves, and C. Pimentel, "Image encryption using block cipher and chaotic sequences," *Signal Process., Image Commun.*, vol. 79, pp. 24–31, Nov. 2019.

[14] M. H. Annaby, M. A. Rushdi, and E. A. Nehary, "Color image encryption using random transforms, phase retrieval, chaotic maps, and diffusion," *Opt. Lasers Eng.*, vol. 103, pp. 9–23, Apr. 2018.

[15] M. Ge and R. Ye, "A novel image encryption scheme based on 3D bit matrix and chaotic map with Markov properties," *Egyptian Inform. J.*, vol. 20, no. 1, pp. 45–54, 2019.

[16] M. Khan, F. Masood, A. Alghafis, M. Amin, and S. Naqvi, "A novel image encryption technique using hybrid method of discrete dynamical chaotic maps and Brownian motion," *PLoS ONE*, vol. 14, no. 12, pp. 1–23, 2019.

[17] 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.

[18] A. Rehman, X. Liao, M. A. Hahsmi, and R. Haider, "An efficient mixed inter-intra pixels substitution at 2bits-level for image encryption technique using DNA and chaos," *Optik*, vol. 153, pp. 117–134, Jan. 2018.

[19] L. L. Huang, S. M. Wang, and J. H. Xiang, "A tweak-cube color image encryption scheme jointly manipulated by chaos and hyper-chaos," *Appl. Sci.*, vol. 9, no. 22, pp. 1–21, 2019.

[20] T. Wang and M. hui Wang, "Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding," *Opt. Laser Technol.*, vol. 132, pp. 1–13, May 2020.

[21] H. Liu, B. Zhao, and L. Huang, "Quantum image encryption scheme using Arnold transform and S-box scrambling," *Entropy*, vol. 21, no. 4, pp. 1–14, 2019.

[22] O. Sengel, M. A. Aydin, and A. Sertbas, "An efficient generation and security analysis of substitution box using fingerprint patterns," *IEEE Access*, vol. 8, pp. 160158–160176, 2020.
[23] H. Liu, A. Kadir, and C. Xu, "Cryptanalysis and constructing S-box based on chaotic map and backtracking," *Appl. Math. Comput.*, vol. 376, Jul. 2020, Art. no. 125153.
[24] L. Yi, X. Tong, Z. Wang, M. Zhang, H. Zhu, and J. Liu, "A novel block encryption algorithm based on chaotic S-box for wireless sensor network," *IEEE Access*, vol. 7, pp. 53079–53090, 2019.
[25] 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.
[26] 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.
[27] A. K. Farhan, R. S. Ali, H. R. Yassein, N. M. G. Al-Saidi, and G. H. Abdul-Majeed, "A new approach to generate multi S-boxes based on RNA computing," *Int. J. Innov. Comput. Inf. Control*, vol. 16, pp. 331–348, Oct. 2020.
[28] Ü. Çavuşoğlu, A. Zengin, I. Pehlivan, and S. Kaçar, "A novel approach for strong S-box generation algorithm design based on chaotic scaled Zhongtang system," *Nonlinear Dyn.*, vol. 87, no. 2, pp. 1081–1094, Jan. 2017.
[29] A. K. Farhan, R. S. Ali, H. Natiq, and N. M. G. Al-Saidi, "A new S-box generation algorithm based on multistability behavior of a plasma perturbation model," *IEEE Access*, vol. 7, pp. 124914–124924, 2019.
[30] E. Tanyildizi and F. Ozkaynak, "A new chaotic S-box generation method using parameter optimization of one dimensional chaotic maps," *IEEE Access*, vol. 7, pp. 117829–117838, 2019.
[31] T. Ul Haq and T. Shah, "12×12 S-box design and its application to RGB image encryption," *Optik*, vol. 217, Sep. 2020, Art. no. 164922.
[32] 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, Nov. 2018.
[33] 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, nos. 11–12, pp. 7279–7297, 2020.
[34] W. Gao, B. Idrees, S. Zafar, and T. Rashid, "Construction of nonlinear component of block cipher by action of modular group PSL(2, Z) on projective line PL(GF(28))," *IEEE Access*, vol. 8, pp. 136736–136749, 2020.
[35] E. Z. Zefreh, "An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions," *Multimedia Tools Appl.*, vol. 79, nos. 33–34, pp. 24993–25022, Sep. 2020.
[36] M. T. Suryadi, Y. Satria, and M. Fauzi, "Implementation of digital image encryption algorithm using logistic function and DNA encoding," in *Proc. Int. Conf. Math., Pure, Appl. Comput.*, 2018, pp. 1–9.
[37] A. R. Parks and J. E. Peters, "Conservative site specific recombination," in *Molecular Life Sciences*, R. D. Wells *et al.*, Eds. New York, NY, USA: Springer, 2018, pp. 119–127.
[38] Q. Yin and C. Wang, "Using breadth-first search and dynamic diffusion," *Int. J. Bifurcation Chaos*, vol. 28, no. 4, pp. 1–13, 2018.
[39] I. El Hanouti, H. El Fadili, and K. Zenkouar, "Cryptanalysis of an embedded systems' image encryption," *Multimedia Tools Appl.*, vol. 80, no. 9, pp. 13801–13820, Apr. 2021.
[40] N. Munir, M. Khan, M. M. Hazzazi, A. Aljaedi, A. A. K. H. Ismail, A. R. Alharbi, and I. Hussain, "Cryptanalysis of Internet of Health Things encryption scheme based on chaotic maps," *IEEE Access*, vol. 9, pp. 105678–105685, 2021.
[41] N. Tsafack, S. Sankar, B. Abd-El-Atty, J. Kengne, J. K. C., A. Belazi, I. Mehmood, A. K. Bashir, O.-Y. Song, and A. A. A. El-Latif, "A new chaotic map with dynamic analysis and encryption application in Internet of Health Things," *IEEE Access*, vol. 8, pp. 137731–137744, 2020.
[42] S. Sun, Y. Guo, and R. Wu, "A novel image encryption scheme based on 7D hyperchaotic system and row-column simultaneous swapping," *IEEE Access*, vol. 7, pp. 28539–28547, 2019.
[43] M. Preishuber, T. Hütter, S. Katzenbeisser, and A. Uhl, "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2137–2150. Sep. 2018.
[44] A. Sambas, M. Mamat, S. Vaidyanathan, M. A. Mohamed, and W. S. M. Sanjaya, "A new 4-D chaotic system with hidden attractor and its circuit implementation," *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 1245–1250, 2018.
[45] P. Aparna and P. V. V. Kishore, "Biometric-based efficient medical image watermarking in E-healthcare application," *IET Image Process.*, vol. 13, no. 3, pp. 421–428, Feb. 2019.
[46] M. Rüdisüli, T. J. Schildhauer, S. M. A. Biollaz, and J. R. Van Ommen, "Measurement, monitoring and control of fluidized bed combustion and gasification," in *Fluidized Bed Technologies for Near-Zero Emission Combustion and Gasification*, F. Scala, Ed. Sawston, U.K.: Woodhead Publishing, 2013, pp. 813–864.
[47] X. Zhang and X. Wang, "Digital image encryption algorithm based on elliptic curve public cryptosystem," *IEEE Access*, vol. 6, pp. 70025–70034, 2018.
[48] L. Chun-Lai and Y. Si-Min, "A new hyperchaotic system and its adaptive tracking control," *Acta Phys. Sinica*, vol. 61, no. 4, 2012, Art. no. 040504.
[49] P. Li, J. Xu, J. Mou, and F. Yang, "Fractional-order 4D hyperchaotic memristive system and application in color image encryption," *EURASIP J. Image Video Process.*, vol. 2019, no. 1, pp. 1–11, Dec. 2019.
[50] M. F. Khan, A. Ahmed, and K. Saleem, "A novel cryptographic substitution box design using Gaussian distribution," *IEEE Access*, vol. 7, pp. 15999–16007, 2019.
[51] G. Cheng, C. Wang, and H. Chen, "A novel color image encryption algorithm based on hyperchaotic system and permutation-diffusion architecture," *Int. J. Bifurcation Chaos*, vol. 29, no. 9, pp. 1–17, 2019.
[52] X. Chai, X. Fu, Z. Gan, Y. Lu, and Y. Chen, "A color image cryptosystem based on dynamic DNA encryption and chaos," *Signal Process.*, vol. 155, pp. 44–62, Feb. 2019.
[53] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 8, no. 6, pp. 1259–1284, Jun. 1998.
[54] Y. Song, Z. Zhu, W. Zhang, H. Yu, and Y. Zhao, "Efficient and secure image encryption algorithm using a novel key-substitution architecture," *IEEE Access*, vol. 7, pp. 84386–84400, 2019.
[55] A. Javeed, T. Shah, and A., "Lightweight secure image encryption scheme based on chaotic differential equation," *Chin. J. Phys.*, vol. 66, pp. 645–659, Aug. 2020.
[56] Y. Chen, "The existence of homoclinic orbits in a 4D lorenz-type hyperchaotic system," *Nonlinear Dyn.*, vol. 87, no. 3, pp. 1445–1452, Feb. 2017.
[57] L. X. Jia, H. Dai, and M. Hui, "A new four-dimensional hyperchaotic Chen system and its generalized synchronization," *Chin. Phys. B*, vol. 19, no. 10, pp. 1–12, 2010.
[58] A. A. Alzaidi, M. Ahmad, M. N. Doja, E. Al Solami, and M. M. S. Beg, "A new 1D chaotic map and β-hill climbing for generating substitution-boxes," *IEEE Access*, vol. 6, pp. 55405–55418, 2018.
[59] T. Farah, R. Rhouma, and S. Belghith, "A novel method for designing S-box based on chaotic map and teaching-learning-based optimization," *Nonlinear Dyn.*, vol. 88, no. 2, pp. 1059–1074, 2017.
[60] H. A. Ahmed, M. F. Zolkipli, and M. Ahmad, "A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7201–7210, 2019.
[61] 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, 2017.
[62] K. E. Attaullah, S. S. Jamal, and T. Shah, "A novel algebraic technique for the construction of strong substitution box," *Wireless Pers. Commun.*, vol. 99, no. 1, pp. 213–226, 2018.
[63] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-The Advanced Ecryption Standard*. Berlin, Germany: Springer, 2002, p. 238.
[64] S. N. Lagmiri, J. Elalami, N. Sbiti, and M. Amghar, "Hyperchaos for improving the security of medical data," *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 1049–1055, Jul. 2018.
[65] X. Wu, J. Kurths, and H. Kan, "A robust and lossless DNA encryption scheme for color images," *Multimed Tools Appl.*, vol. 77, no. 10, pp. 12349–12376, May 2018.
[66] L. Bassham *et al.*, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST-SP, Gaithersburg, MD, USA, 2010.
[67] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, Oct. 2019.
[68] S. Sun, "A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling," *IEEE Photon. J.*, vol. 10, no. 2, pp. 1–15, Mar. 2018.
[69] A. Belazi, M. Talha, S. Kharbech, and W. Xiang, "Novel medical image encryption scheme based on chaos and DNA encoding," *IEEE Access*, vol. 7, pp. 36667–36681, 2019.

[70] B. Arpacı, E. Kurt, and K. Çelik, "A new algorithm for the colored image encryption via the modified Chua's circuit," *Eng. Sci. Technol., Int. J.*, vol. 23, no. 3, pp. 595–604, Jun. 2020.
[71] A. Alghafis, F. Firdousi, M. Khan, S. I. Batool, and M. Amin, "An efficient image encryption scheme based on chaotic and deoxyribonucleic acid sequencing," *Math. Comput. Simul.*, vol. 177, pp. 441–466, Nov. 2020.
[72] G. Zhang, W. Ding, and L. Li, "Image encryption algorithm based on tent delay-sine cascade with logistic map," *Symmetry*, vol. 12, no. 3, pp. 1–14, 2020.
[73] K. Zhan, D. Wei, J. Shi, and J. Yu, "Cross-utilizing hyperchaotic and DNA sequences for image encryption," *J. Electron. Imag.*, vol. 26, no. 1, Feb. 2017, Art. no. 013021.
[74] A. Belazi, A. A. A. El-Latif, and S. Belghith, "A novel image encryption scheme based on substitution-permutation network and chaos," *Signal Process.*, vol. 128, pp. 155–170, Nov. 2016.
[75] A. Firdous, A. U. Rehman, and M. M. S. Missen, "A gray image encryption technique using the concept of water waves, chaos and hash function," *IEEE Access*, vol. 9, pp. 11675–11693, 2021.
[76] M. Sokouti and B. Sokouti, "A PRISMA-compliant systematic review and analysis on color image encryption using DNA properties," *Comput. Sci. Rev.*, vol. 29, pp. 14–20, Aug. 2018.
[77] L. Liu and S. Miao, "An image encryption algorithm based on baker map with varying parameter," *Multimedia Tools Appl.*, vol. 76, no. 15, pp. 16511–16527, Aug. 2017.
[78] S. Kandar, D. Chaudhuri, A. Bhattacharjee, and B. C. Dhara, "Image encryption using sequence generated by cyclic group," *J. Inf. Secur. Appl.*, vol. 44, pp. 117–129, Feb. 2019.
[79] Y. Luo, R. Zhou, J. Liu, S. Qiu, and C. Yi, "An efficient and self-adapting colour-image encryption algorithm based on chaos and interactions among multiple layers," *Multimedia Tools Appl.*, vol. 77, no. 20, pp. 26191–26217, Oct. 2018.
[80] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using dna sequence operations," *Opt. Lasers Eng.*, vol. 88, pp. 197–213, Jan. 2017.
[81] A. F. Khan, J. Ahmed, S. J. Khan, J. Ahmad, and M. Khan, "A novel image encryption based on Lorenz equation, Gingerbreadman chaotic map and S_8 permutation," *J. Intell. Fuzzy Syst.*, vol. 33, no. 6, pp. 3753–3765, 2017.
[82] P. Mohanaiah, P. Sathyanarayana, and L. Gurukumar, "Image texture feature extraction using GLCM approach," *Int. J. Sci. Res. Publ.*, vol. 3, no. 5, pp. 1–5, 2013.
[83] J. S. Khan and J. Ahmad, "Chaos based efficient selective image encryption," *Multidimensional Syst. Signal Process.*, vol. 30, no. 2, pp. 943–961, Apr. 2019.
[84] K. A. K. Patro and B. Acharya, "Secure multi–level permutation operation based multiple colour image encryption," *J. Inf. Secur. Appl.*, vol. 40, pp. 111–133, Jun. 2018.
[85] S. Zhou, P. He, and N. Kasabov, "A dynamic DNA color image encryption method based on SHA-512," *Entropy*, vol. 22, no. 10, pp. 1–23, 2020.
[86] R. Dass, "Speckle noise reduction of ultrasound images using BFO cascaded with Wiener filter and discrete wavelet transform in homomorphic region," *Proc. Comput. Sci.*, vol. 132, pp. 1543–1551, Jan. 2018.
[87] W. Xingyuan, Z. Junjian, and C. Guanghui, "An image encryption algorithm based on ZigZag transform and LL compound chaotic system," *Opt. Laser Technol.*, vol. 119, Nov. 2019, Art. no. 105581.
[88] A. Jolfaei and A. Mirghadri, "A new approach to measure quality of image encryption," *Int. J. Comput. Netw. Inf. Secur.*, vol. 2, no. 8, pp. 38–43, 2010.
[89] Z. Hua and Y. Zhou, "Design of image cipher using block-based scrambling and image filtering," *Inf. Sci.*, vol. 396, pp. 97–113, Aug. 2017.
[90] M. Samiullah, W. Aslam, H. Nazir, M. I. Lali, B. Shahzad, M. R. Mufti, and H. Afzal, "An image encryption scheme based on DNA computing and multiple chaotic systems," *IEEE Access*, vol. 8, pp. 25650–25663, 2020.
[91] T. Nadu, "Encryption quality and performance analysis of GKSBC algorithm," *J. Inf. Eng. Appl.*, vol. 2, no. 10, pp. 26–34, 2012.
[92] Z. Li, C. Peng, W. Tan, and L. Li, "A novel chaos-based color image encryption scheme using bit-level permutation," *Symmetry*, vol. 12, no. 9, pp. 1–17, 2020.
[93] W. Hou, S. Li, J. He, and Y. Ma, "A novel image-encryption scheme based on a non-linear cross-coupled hyperchaotic system with the dynamic correlation of plaintext pixels," *Entropy*, vol. 22, no. 7, pp. 1–21, 2020.
[94] L. Ding and Q. Ding, "A novel image encryption scheme based on 2D fractional chaotic map, DWT and 4D hyper-chaos," *Electronics*, vol. 9, no. 8, pp. 1–20, 2020.

HIRA NAZIR received the B.S. degree in Information Technology and the M.S. degree in software engineering from The Islamia University of Bahawalpur, Pakistan, where she is currently pursuing the Ph.D. degree in computer science. She is currently an Associate Lecturer with The Islamia University of Bahawalpur. Her research interests include formal methods, and process models in software engineering, digital image watermarking, social network data analysis, interference management for cellular systems, and applications of DNA computing in information security.

IMRAN SARWAR BAJWA is currently the Chairperson and an Associate Professor in computer science with The Islamia University of Bahawalpur, Pakistan. His research interests include computer science, artificial intelligence, and smart systems. He is an Associate Editor of IEEE Access. He is a Guest Editor of *Computers & Electrical Engineering* journal (Elsevier), *SAGE Open*, *J.UCS* (Springer), and *Journal of Ambient Intelligence and Smart Environments*. He is the General Chair of INTAP Conference.

SAIMA ABDULLAH is currently an Assistant Professor in computer science with The Islamia University of Bahawalpur, Pakistan, where she is also an Assistant Professor with the Department of Computer Science. Her research interests include wireless sensor networks, the Internet of Things (IoT), scheduling patterns of sensor messages, node failures management, and QoS aware message scheduling.

RAFAQUT KAZMI received the Ph.D. degree from University Technology Malaysia. He is currently working as an Associate Professor with the Department of Software Engineering, The Islamia University of Bahawalpur. His research interests include software testing, agile, the IoT, cloud computing, sliding mode control, fractional control, neural networks, cognitive radio networks, and network security.

MUHAMMAD SAMIULLAH received the B.S., M.B.A., and M.S. degrees in computer science from The Islamia University of Bahawalpur, Pakistan, where he is currently pursuing the Ph.D. degree in computer science. He is currently an Instructor and the Head of the Information Technology Department, Government Sadiq Graduate College of Commerce, Bahawalpur, Pakistan. His research interests include wireless network virtualization, radio resource allocation, interference management for cellular systems and applications of non-linear dynamics, and DNA computing in information security.