A new four-dimensional hyper-chaotic system for image encryption

Huda R. Shakir, Sadiq A. Mehdi, Anwar A. Hattab
Department of Computer Science, Mustansiriyah University, Baghdad, Iraq

Article Info

Article history:

Received Mar 30, 2022
Revised Sep 16, 2022
Accepted Oct 13, 2022

Keywords:

Hyper-chaotic system
Image encryption
Permutation
Substitution

ABSTRACT

Currently, images are very important with the rapid growth of communication networks. Therefore, image encryption is a process to provide security for private information and prevent unwanted access to sensitive data by unauthorized individuals. Chaos systems provide an important role for key generation, with high randomization properties and accurate performance. In this study, a new four-dimensional hyper-chaotic system has been suggested that is used in the keys generation, which are utilized in the image encryption process to achieve permutation and substitution operations. Firstly, color bands are permuted using the index of the chaotic sequences to remove the high correlation among neighboring pixels. Secondly, dynamic S-boxes achieve the principle of substitution, which are utilized to diffuse the pixel values of the color image. The efficiency of the proposed method is tested by the key space, histogram, and so on. Security analysis shows that the proposed method for encrypting images is secure and resistant to different attacks. It contains a big key space of (2627) and a high sensitivity to a slight change in the secret key, a fairly uniform histogram, and entropy values nearby to the best value of 8. Moreover, it consumes a very short time for encryption and decryption.

Corresponding Author:

Huda R. Shakir
Department of Computer Science, University of Mustansiriyah
Alkadhimiya Street, Baghdad, Iraq
Email: hudarashid@uomustansiriyah.edu.iq

1. INTRODUCTION

With the rapid advancement of digital technologies and communication networks, multimedia communication such as images, videos, and audio is becoming increasingly significant and indispensable in today's society [1]–[3]. However, network openness and sharing introduce several security risks to digital communication. Digital images, as a significant medium of multimedia communication, play a major part in medical, biological, military, and social life. Digital images contain a lot of important information, so the security of image data attracts more and more attention [4], [5].

Due to the strong correlation between image pixels, high redundancy, and large data capacity [6], [7], therefore, conventional encryption algorithms like advanced encryption standard (AES), data encryption standard (DES), and Rivest-Shamir-Adleman (RSA) algorithms are not appropriate for image encryption [8], [9]. As a result, many researchers have proposed new image coding algorithms that satisfy the criteria of confusion and diffusion [10]. Compared with conventional encryption systems, chaotic systems have stronger advantages such as sensitivity to initial value, ergodicity, randomness, and other characteristics [11]–[13], which provide a quick and secure way to protect data transmitted over communication channels like the internet. In order to ensure the effectiveness of the algorithms, Fredrich was the first to combine the chaotic system with cryptography theory in 1998 [10], [14].

Chaotic systems have been used in many studies to encrypt images. Below is a review of some relevant work. Ali and Ali [15] proposed a new approach for encrypting color images using chaotic maps. Three stages are involved in the construction of the cipher image. The first phase involves permuting the digital image using a chaotic map. The second step employs a chaotic substitution box to perform pixel substitutions, and the third phase employs the Boolean operation exclusive OR (XOR) to blend chaotic logistic-based random sequences. The red, green and blue (RGB) components of an image that have been scrambled using permutation, substitution, and XOR operations have good security and resistance to various types of encryption attacks.

Teng *et al.* [16] proposed a new image encryption method based on a cross 2D hyper-chaotic map. The keys are constructed by the hash function (SHA-512) and the plain image. Firstly, the color image is transformed to a merged bit-level array and scrambled using the column and row cogeneration shift process. The scrambled matrix is then diffused using a choosing sequence that is dependent on the chaotic sequence. Lastly, the diffusion matrix is decomposed in order to obtain the color image encryption. Simulations and security analysis show that the method works well at encrypting the color image and provides adequate protection against various types of attacks.

Tariq *et al.* [17], suggested an image encryption scheme based on Lorenz chaotic systems and analyzed the flaws of the public key encryption method based on logarithm and provided an additional layer of security to prevent it from cryptographic assaults. According to both security analyses and experimental findings, the suggested encryption method has proven to be resistant to both linear and differential attacks. Patro *et al.* [18], suggested an image encryption approach based on multiple levels of permutations and diffusions. Bit-level permutations are done with a 4-D hyper-chaotic map. The diffusion step is done with the piecewise linear chaotic map (PWLCM) system, and block-level permutations are done with Alpar's map. According to simulation and security analysis, the comparison tests show that the proposed method is secure and immune to most common threats.

Talhaoui *et al.* [19], suggested a new image encryption method dependent on the chaotic map of the Bülban. The new scheme's security is provided via a substitution and permutation network, which uses a rotating shift of columns and rows to remove the high correlation among neighboring pixels. Then, the pixels values are mask the of with XOR and the modulo function to prevent any data from leaking into the system. Tests and simulations have shown that the system is safe and very fast for real-time images.

Most encryption algorithms have three key problems: insufficient security, limited processing capacity, and poor encryption efficiency. So, traditional algorithms are not suitable for image encryption, in addition to the problems of low-chaotic systems such as small key space and low complexity. To overcome these problems, in this paper, a new color image encryption method built on a novel 4-D chaotic system has been suggested to offer higher security and randomness. Initially, the position of pixels are scrambled through the sequences chaotic; then, high confusion is provided by the dynamic S-Boxes constructed using DNA computing with the new chaotic system. An experimental and security analysis was performed on the proposed image encryption algorithm to verify the security and efficiency of the encryption.

2. THEORETICAL BACKGROUND

2.1. The new chaotic system

The novel 4D-four-dimensional hyperchaotic system with ten positive parameters is proposed. It has more complex properties than lower-dimensional chaotic systems. An entirely new 4D autonomous system can be can be obtained by the following differential equation in system (1).

$$\begin{aligned} \frac{dx}{dt} &= -a\,x - b\,w + c\,y\,z + z\,e^{y} \\ \frac{dy}{dt} &= d\,y + e\,x - f\,x\,z - x\,e^{z} \\ \frac{dz}{dt} &= -g\,z + h\,x\,y \\ \frac{dw}{dt} &= -bw + i\,x\,z + jy\,z \end{aligned} \tag{1}$$

The states of system are x, y, z, and w, t ∈ real number and the positive parameters are $b, d, a, c, g, e, h, f, j, i$ wich displays a chaotic attractors in the new 4-D chaotic system by the initial conditions of: $x_0 = 0.2, y_0 = 0.4,\ z_0 = 1.5,$ and $w_0 = 0.8$ and parameter values of: $a = 3.1,\ b = 2.1,\ c = 15.8,\ d = 1.1,\ e = 16.5, f = 1.5,\ g = 2.4, i = 5.1, h = 26.6,$ and $j = 12.9$. Figure 1(a) shows the chaotic attractors in (y-x-w), Figure 1(b) in (x-y-z), Figure 1(c) in (x-z-w), and Figure 1(d) in (y-x-z). The four Lyapnov Exponents of system (1), by parameters are obtained: LE_1=4.05761, LE_2=0.347562, LE3=-3.94257 and LE4=-6.61896. Due to LE1 and LE2 are positive Lyaponov exponents. Therefore, the new system is a hyper chaotic.

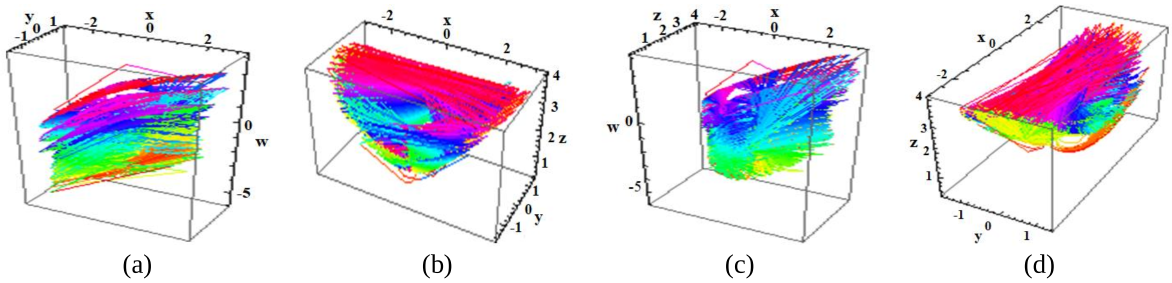

Figure 1. Attractors of chaotic (a) (y-x-w), (b) (x-y-z), (c) (x-z- w), and (d) (y-x-z)

2.2. Waveform analysis of the novel chaotic system

The waveform of a chaotic system should be aperiodic in order to show that the suggested system is a chaotic system, as shown in Figure 2. Figure 2(a) depicts the *x(t)* waveform, Figure 2(b) the *y(t)* waveform, Figure 2(c) the *z(t)* waveform, and Figure 2(d) the *w(t)* waveform in the time domain. By plotting the results of a MATHEMATICA simulation, it is clear that the waveform of the system has complex behavior, chaotic motion, and non-periodic characteristics. So, the time-domain waveform has non-cyclical characteristics so that it can tell the difference between several periodic motions with complicated behaviors and chaotic motion.

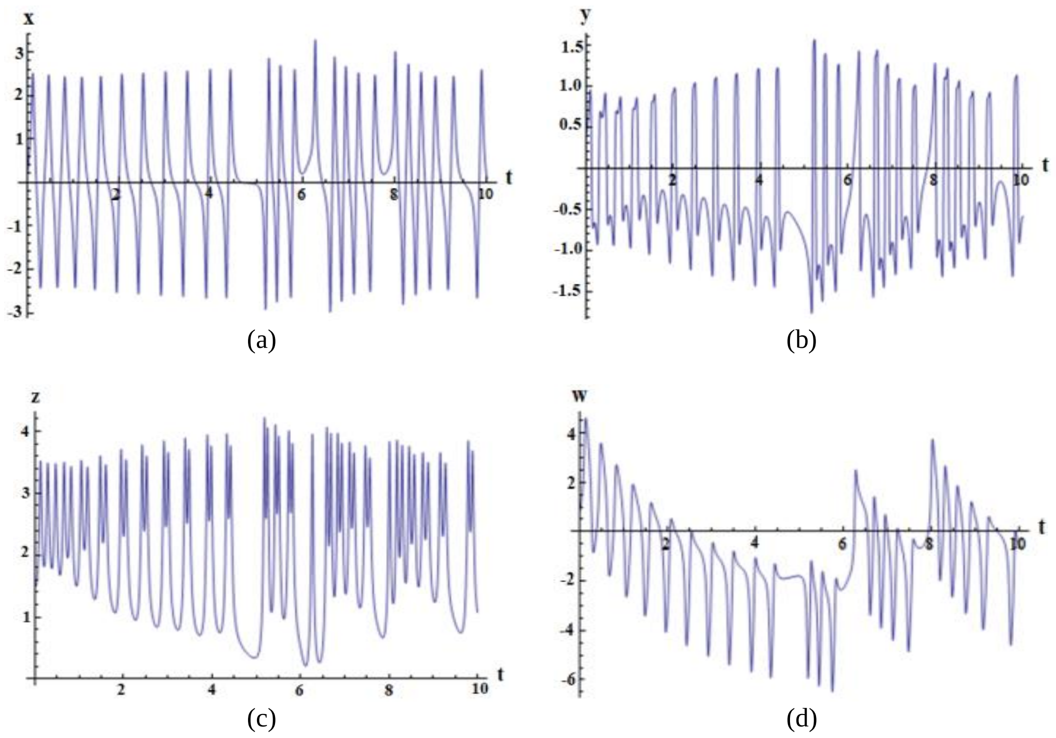

Figure 2. Waveforms of the novel chaotic system (a) *x(t)*, (b) *y(t)*, (c) *z(t)*, and (d) *w(t)*

2.3. The sensitivity to initial conditions of the novel chaotic system

Long-term unpredictability is perhaps the most key characteristic of a chaotic system. This is due to the sensitivity of solution dependency on initial variables. Two distinct initial states, despite how close they begin, will eventually separate widely. As a result, for every initial condition with a finite number of digits of accuracy, a future time will come when no reliable predictions can be formed about the state of a system. Figures 3(a) to 3(d) demonstrate the sensitivity of the chaotic trajectories' evolution to the initial conditions in *x(t), y(t), z(t)*, and *w(t)*. For the solid line, the initial values are x_0=0.2, y_0=0.4, z_0=1.5, and w_0=0.8; for the dashed line, they are *x(0)=0.2*, *y(0)=0.400000000000001*, *z(0)=1.5*, and *w(0)=0.8*.

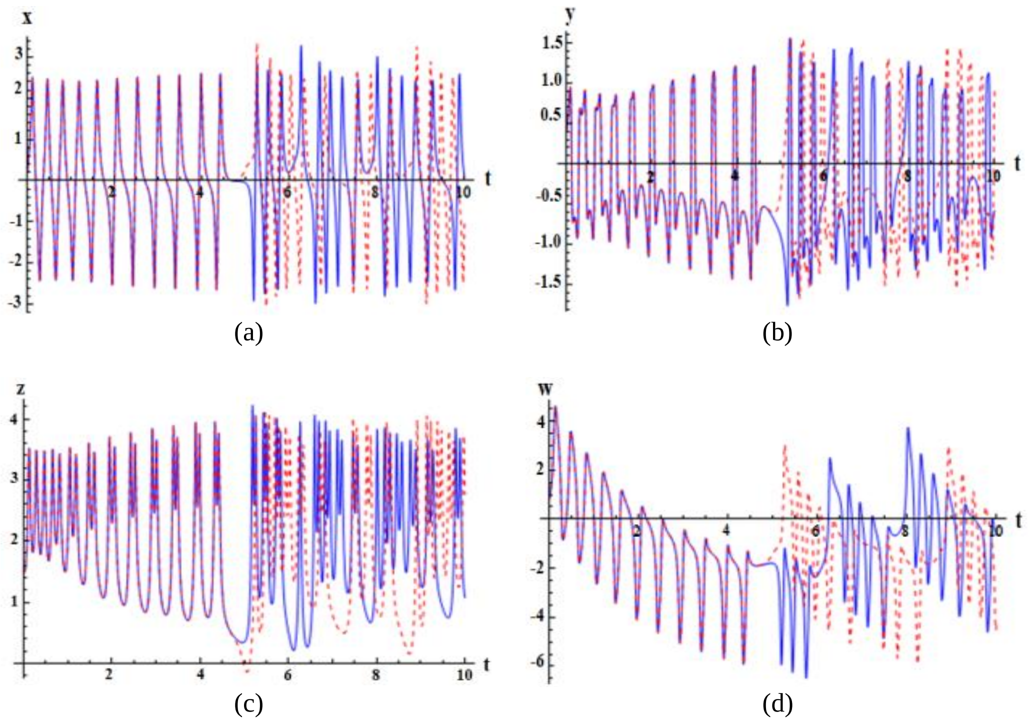

Figure 3. Sensitivity tests of the new system (a) *x(t)*, (b) *y(t)*, (c) *z(t)*, and (d) *w(t)*

3. RESEARCH METHOD

In order to increase the level of security, we proposed a new algorithm for color image encryption based on a new four-dimensional chaotic system and dynamic S-boxes. The suggested method consists of two main phases: the encryption phase and the decryption phase. These steps will be explained in detail in the following sections.

3.1. Encryption phase

The encryption phase uses the chaotic sequences to transform red, green and blue (RGB) images into encoded images with unexpected features to withstand statistical attacks. It includes four stages: generating four chaotic sequences from the new 4D-hyperchaotic system, permutation, S-box generation, and substitution, in order to produce the final encrypted image. Figure 4 explains the total steps of the encryption phase.

3.1.1. Generate chaotic sequences from the new chaotic system

The generation of chaotic sequences begins by entering the parameters with the initial values that belong to the 4D chaotic system. In this step, we generate four chaotic sequences: *xn, yn, zn,* and *wn*; the length of each sequence is the same as the size of the image dimensions (*Row*×*Column*). Then, sort sequences in ascending order for *xn*, *yn*, and *zn*; and take the locations for each element of these sequences and store them in the index vectors *Xi*, *Yi*, and *Zi*, which represent the keys.

3.1.2. Permutation

There are strong correlations between neighboring pixels in most of the images, so we utilize chaotic sequences to permute the RGB image pixels in order to eliminate the correlations. The first step in the proposed algorithm is image permutation. The plain color image RGB is split into three channels and converted into three vectors: V_R, V_G, and V_B and then is permuted depending on three sequences: x_i, y_i, and z_i that were generated in the previous step, so that the correlation between them is lessened. In order to scramble the position of every value in the vectors, the sequence $\{x_i\}$ scrambles the red vector (V_R), the sequence $\{yi\}$ scrambles the green vector (V_G), and the sequence $\{zi\}$ scrambles the blue vector (V_B). Hence, these vectors represent the permutation image.

3.1.3. S-boxes generation

At this stage, we proposed new S-boxes to achieve the substitution process in image encryption, built on a hybrid method combining a chaotic system and DNA encoding. Firstly, depending on the chaotic sequences (*xi*, *yi*, *zi*, and *wi*), which have been transformed to vectors *k1*, *k2*, *k3*, and *k4* from chaotic sequences, also, using algebraic DNA processes (exclusive-or, subtraction, and addition), which are used to make a set of S-boxes (1616), each chaotic sequence is turned into a binary sequence and then changed to a DNA sequence so that each pair of bits is replaced by a DNA code. For example, (00) is replaced by (A), (01) by (C), (10) by (G), and (11) by (T), giving the four sequences (*k1*, *k2*, *k3*, and *k4*). The addition operation is applied to the first and second sequences (*k1*, *k2*), while the subtraction operation is applied to the third and fourth sequences (*k3*, *k4*). Finally, the XOR process is performed on the results of the previous subtraction and addition stages. The results of these operations are used for generating a set of S-boxes by getting each digit as one cell in the S-box, and this cell should not be repeated before. The process continues until unique 256 values are placed in the S-box. Table 1 denotes the results of applying the suggested method for creating S-box.

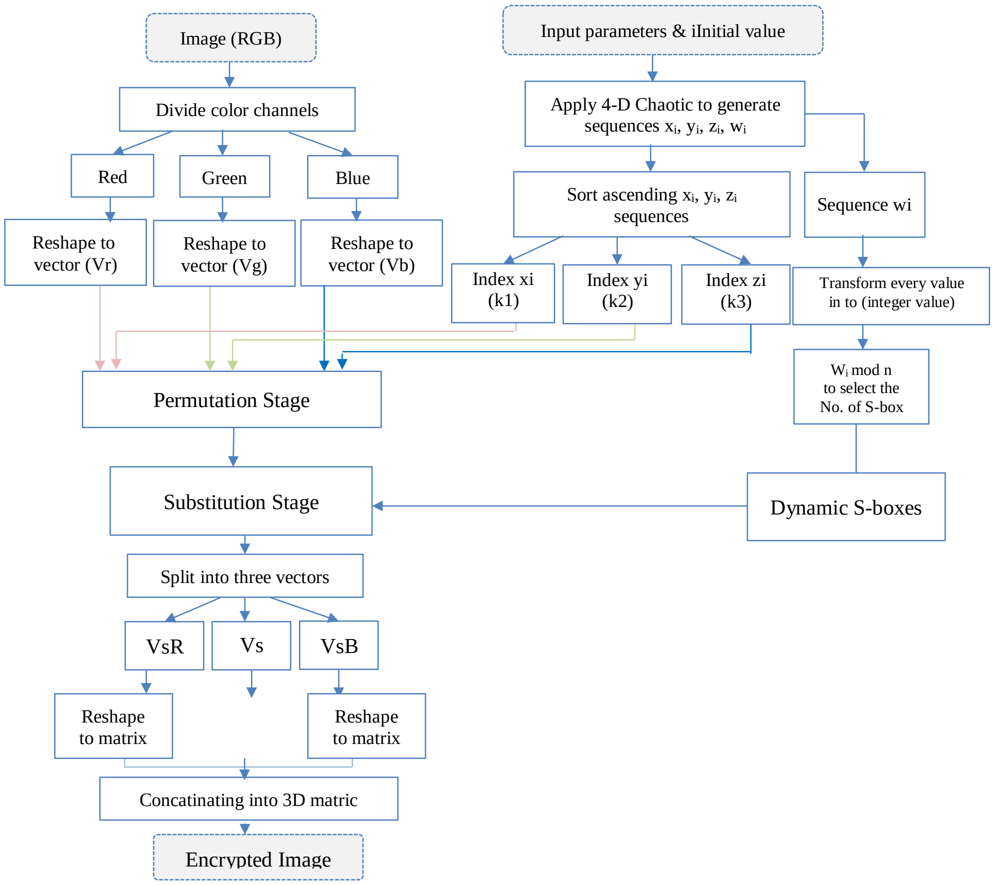

Figure 4. Block diagram of encryption algorithm

3.1.4. Substitution

The S-box achieves the principle of substitution. In this stage, the permuted pixels in the previous step will be substituted based on new S-boxes have been generated, where three permuted vectors merge into one vector and are split into blocks, each block having a size of 8×8 bytes. Each block is converted into a hexadecimal string and substituted by one S-box. Based on a (*wi*) chaotic sequence for the selection No. of S-box, (*wi* mod *n*), where *wi* represents the No. of S-box. After the S-box selection, each two-digit hexadecimal is substituted in the S-box. To change the value of each pixel, the first hexadecimal digit is given to the rows and the second hexadecimal digit is given to the columns. The value resulting from the intersection of the row and column in the s-box is put in the new matrix. The previous steps are repeated until all the blocks are completed and the get the image confusion. Algorithm 1 shows the total steps of the encryption method.

Table 1. S-box values results

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	50	41	33	36	91	177	51	21	118	25	159	237	62	1	169	103
1	87	200	39	4	40	246	229	213	139	110	134	69	190	119	235	142
2	17	70	170	97	194	147	3	211	135	2	143	243	116	5	35	67
3	78	167	121	145	204	37	42	132	140	188	141	90	57	14	201	18
4	89	133	20	196	205	185	191	117	210	228	131	56	197	179	249	32
5	52	181	250	66	47	0	82	166	83	199	58	175	100	216	182	113
6	176	45	202	55	48	157	178	184	12	88	239	232	248	44	6	99
7	102	122	238	163	22	128	168	92	43	187	98	11	72	74	24	153
8	165	207	215	34	148	26	150	149	227	127	192	64	107	209	95	120
9	171	96	231	241	198	212	164	254	46	236	253	81	230	125	138	245
A	161	7	195	31	38	75	233	77	30	71	65	189	158	162	106	19
B	146	203	61	60	244	86	208	123	222	224	180	151	206	156	79	152
C	193	252	94	54	85	242	8	240	255	68	124	214	247	15	111	109
D	126	154	53	218	49	217	155	59	225	105	226	173	221	16	104	130
E	234	219	115	136	76	13	28	174	9	108	10	29	172	23	186	137
F	63	114	220	144	80	27	73	101	93	129	160	251	112	84	223	183

Algorithm 1. Encryption
Input: plain color image (PI), initial values: Y0, X0, Z0, W0, parameters: b, d, a, c, g, e, h, f, j, i
Output: Encrypted image (EI)
Begin
Step 1: Read color image (PI)
Step 2: M$\leftarrow$hight of PI
N$\leftarrow$width of PI
S$\leftarrow$ M×N
Step 3: Iterate proposed chaotic system with parameter and initial conditions to
Create four series x_n, y_n, z_n, and w_n, size of sequences $\geq$S
Step 4: split color image (im) into R, G, B bands
Step 5: Each color channel (Red, Green, and Blue) are reshape to (V_R, V_G, and V_B) vector
Step 6: Sort (xi, yi, zi) sequences in ascending order, get the index, and find the
position of each sequence in the index (k1, k2, k3)
Step 7: Using k1 to permutate the vector (V_R) // (Red scrambling)
using k2 to permutate the vector (V_G) // (Green scrambling)
using k3 to permutate the vector (V_B) // (Blue scrambling)
Step 8: Merging the three (V_R, V_G and V_B) vectors to one (V_{RGB}) vector
Step 9: divide the (V_{RGB}) vector to blocks of size 8×8
Step 10: using the 4th sequence wi as index (wi mod n) of dynamic S-Boxes, to select the
No. of S-box // n the number of S-boxes produced
Step 11: After selecting the No. of S-box, for each block we substitute each value that
denotes the address of the row and column in the S-box and get the new value
by the intersection of the row and column.
Step 12: Continue until all the blocks are complete to get the confused image (Vsbox).
Step 13: split Vsbox into three vectors VsR, VsG, and VsB
Step 14: reshape VsR, VsG and VsB into 2D (m_R, m_G, and m_B) matrix
Step 15: combine (m_R, m_G, and m_B) to create 3-D of matrix represented the enciphered image
End

3.2. Decryption stage

Decryption is the inverse operation of encryption. The inputs to this stage are the encrypted image, initial condition values, and parameters, while the output is the extracted image. At first, the chaotic sequences are generated by iterating the proposed new chaotic system to create sequences: *xi*, *yi*, *zi*, and *wi*. These sequences are based on the same values of parameters and primary values as the encryption phase. Second, in the inverse substitution, the encrypted image is split into three bands and reshaped into vectors. The vectors are divided into blocks of size 8×8. Then each block is substituted by one from an inverse S-box. The chaotic {*wi*} sequence is used to select the inverse S-box (wi mod n), which is the number (NO. of inverse S-boxes). Thirdly, the inverse permutations are performed by the index of the chaotic sequences (*k1*, *k2*, and *k3*) and the image vectors (*ViR*, *ViB*, and *ViB*) to get the re-permuted image. Lastly, the image's vectors are combined and turned into a matrix that shows the extracted image. Figure 5 shows the total steps in the decryption stage, and algorithm 2 displays the main steps of decryption:

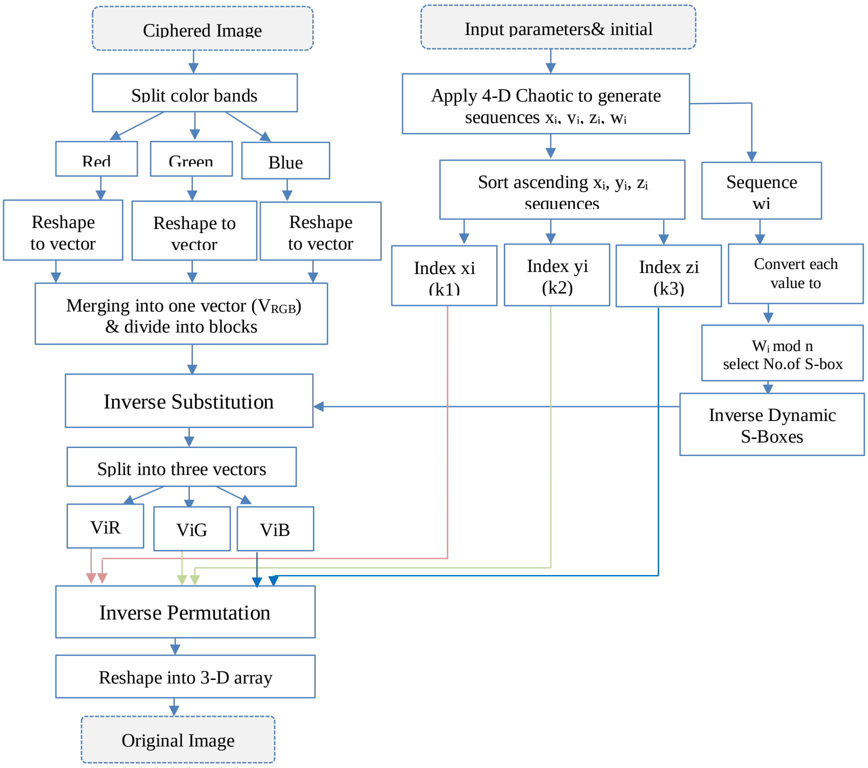

Figure 5. Block diagram of decryption algorithm

Algorithm 2. Decryption
Input: Encrypted image (imx), initial values: Y0, X0, Z0, W0, parameters: b, d, a, c, g, e, h, f, j, i
Output: Original image (Ex) of size (h×w×3)
Begin
Step 1: read encrypted image (imx)
Step 2: m←hight of imx
n←width of imx
S← m×n
Step 3: Iterate proposed chaotic system with parameter and initial conditions to Create four series x_n, y_n, z_n, and w_n, size of sequences ≥S
Step 4: split image (imx) into three R, G, B bands
Step 5: Each color channel (Red, Green, and Blue) are reshape to (IV_R, IV_G, and IV_B)vector
Step 6: Merge IV_R, IV_G, and IV_B to one (IV_{RGB}) vector
Step 7: divide the (IV_{RGB}) vector to blocks of size 8×8
Step 8: using the 4th sequence wi as index (wi mod n) of inverse dynamic S-Boxes, to select the No. of Inverse S-box // n the number of inverse S-boxes produced
Step 9: After selecting the No. of inverse S-box, for each block we substitute each value that denotes the address of the row and column in the S-box and get the new value by the intersection of the row and column.
Step 10: Continue until all the blocks are complete to get the Re-confused image(IVsbox)
Step 11: split (IVsbox) into three vectors ViR, VsG, and VsB
Step 12: Sort (xi, yi, zi) sequences in ascending order, get the index, and find the position of each sequence in the index (k1, k2, k3).
Step 13: Using (k1) to re-permutate the vector(ViR)
using (k2) to re-permutate the vector(ViG)
using (k3) to re-permutate the vector(ViB)
Step 14: reshape ViR, ViG and ViB into 2D matrices MRi, MGi, and MBi
Step 15: combine MRi, MGi, and MBi to create 3D matrix represented the ciphered image(Ex)
End

4. RESULTS AND ANALYSIS

In order to illustrate our algorithm's security and efficiency, some security analysis has been implemented on the suggested enciphering algorithm. In this paper, several standard images (256×256×3 and 512×512×3 pixels) have been tested. Furthermore, the proposed algorithm is simulated in MATLAB R2020a, and our testing results were compared with those of other encryption methods of the same type. These analyses are discussed.

4.1. Key space analysis

A cryptosystem's key space should be at least 2^{100} to withstand brute force attacks [20], [21]. If the precision is in the range of 10^{-14}, then the key space is $10^{196}\approx 2^{627}$. Therefore, the key space is large enough to withstand brute force attack. The key space comparative results between the proposed method and the related works are shown in Table 2.

Table 2. Comparison the key space with the other methods

Algorithm	key space
Proposed method	2^{627}
Ref. [15]	2^{299}
Ref. [16]	2^{425}
Ref. [17]	$\cong 4\times10^{12}$

4.2. Key sensitivity analysis

A secure encryption system should be extremely sensitive to small changes in the key. In other words, the encryption technique should be able to withstand brute-force attacks and must be sensitive to any changes in the keys [22]. As explained in Figure 6, the key sensitivity for images is Figure 6(a) show the original image, Figure 6(b) show an encrypted image, Figure 6(c) show a decrypted image with the correct key, and Figure 6(d) show a decrypted image with the incorrect key.

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

Figure 6. The key sensitivity for images (a) original image, (b) encrypted image, (c) decrypted image with the correct key, and (d) decrypted image with the incorrect key

4.3. Histogram analysis

To stop the attacker from extracting any data contained in the plain image, any statistical relationship or similarity between the plain and decrypted images should be avoided. From a histogram analysis of an image, the statistical properties of that image can be derived. The plain and ciphered images must have completely different statistical properties [23]. Figure 7 displays the histogram of images tested before and after encryption.

4.4. Entropy analysis

Entropy is the significant characteristic that reflects information's randomness and unpredictability [24], [25]. The entropy of the cipher image should ideally be 8 [11]. The entropy of H (s) can be calculated:

$$H(s) = -\sum_{i=0}^{N-1} p(s_i)\log_2 p(s_i) \quad (2)$$

where N denotes the number needed to represent the symbol *si*. *s* denotes the source, and *P (si)* is the symbol's probability. Table 3 shows the entropy values of encryption images.

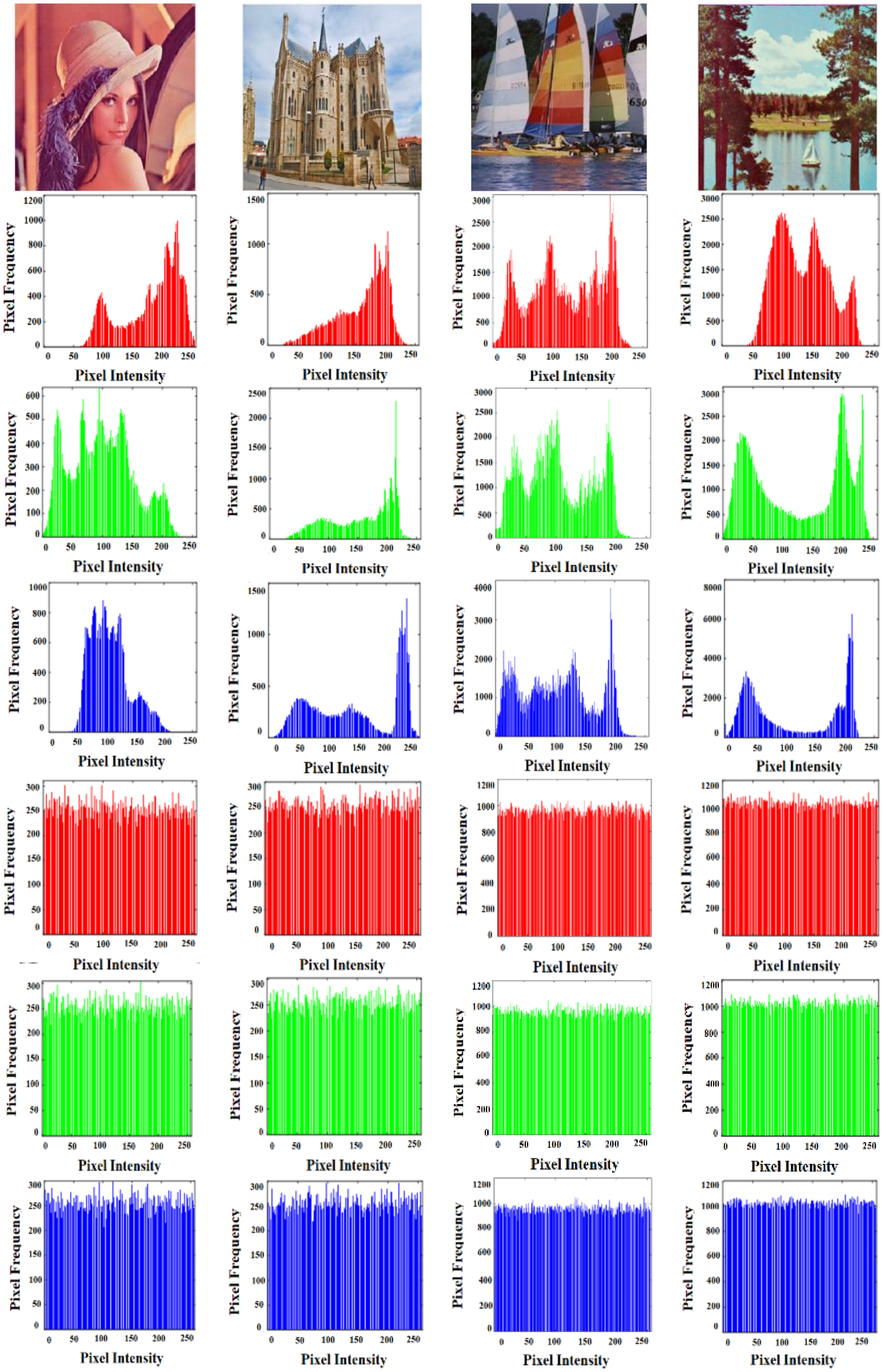

Figure 7. The histogram of image tested before and after encryption

Table 3. The entropy test of images

Name image	Entropy of original images	Entropy of encryption
Lina	7.7516	7.99857
palace	7.09016	7.99896
Yaght	7.6035	7.99802
sailboat on lake	7.7632	7.99897

4.5. Correlation coefficients analysis

Correlation coefficient analysis is used to determine the degree of similarity between plaintext and ciphertext images [1], [26]. Table 4 shows the correlation coefficients analysis. The test was applied to red, green, and blue colors, respectively, between an original image and an encrypted image. As shown in Figure 8, the correlation is applied in horizontal, which tests the pixel with a neighbor in a row, vertical, which tests the pixel with a neighbor in a column, and diagonal, which tests the pixel with a neighbor pixel in the diagonal.

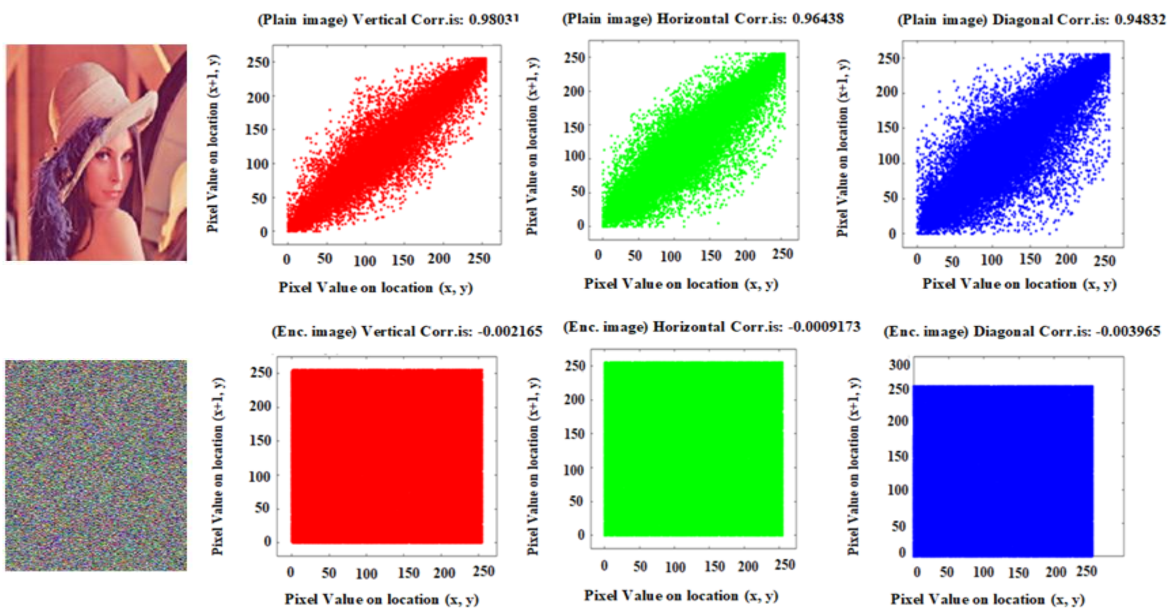

Figure 8. Correlation coefficients of plain and encrypted image

Table 4. Correlation coefficients analysis

Direction	image	Lena	palace	Yaght	Sailboat	Direction
Horizontal	original	0.96438	0.89285	0.97313	0.96793	original
	encryption	-0.00020	0.00036	0.00068	-0.00121	encryption
vertical	original	0.98031	0.92422	0.96652	0.96448	original
	encryption	-0.00045	0.00130	0.00356	-0.00126	encryption
diagonal	original	0.94832	0.84829	0.93995	0.95048	original
	encryption	-0.00474	-0.00612	-0.00141	-0.00090	encryption

4.6. Number of pixels change rate (NPCR) and unified average changing intensity (UACI) analysis

Two differential assault metrics are used to evaluate how vulnerable the original data is to slight alterations: NPCR as well as UACI. Suppose the enciphered image is (C and C') before and after changing one pixel in the original image [27]–[29]. The results of applying the proposed method on tested images are represented in Table 5. The (3) and (4) [30], express the NPCR and UACI formulas, respectively:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \quad (3)$$

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|c(i,j) - c'(i,j)|}{255}\right] \times 100\% \quad (4)$$

4.7. Peak signal-to-noise ratio (PSNR) and mean square error (MSE) analysis

The PSNR and MSE are two of the more commonly used tests for image encryption techniques; the PSNR may be used to evaluate an encrypting method. It is a metric that indicates the difference in pixel values between the plain and cipher images. A lower PSNR value indicates a higher encoding quality. The PSNR and MSE formulas are [31]:

$$PSNR = 20\log_{10}\left(\frac{255}{\sqrt{MSE}}\right)dB \tag{5}$$

$$MSE = \frac{1}{M\times N}\Sigma_{i.j}\ \ (p_0(i.j) - p_1(i.j))^2 \tag{6}$$

where M and N represent the height and width of images. For the original and ciphered images, the intensity values of the pixels are P0 (i, j) and P1 (i, j). According to the results, a high MSE value and a low PSNR value between the original and encrypted images indicate desirable encryption quality, as shown in Table 6.

Table 5. UCAI and NPCR result

Image	NPCR	UCAI
Lena	99.6367	33.0305
Palace	99.6329	29.4177
Yaght	99.6183	31.0267
Sailboat	99.6152	34.0265

Table 6. The MSE and PSNR test

Image	MSE	PSNR
Lena	8914.693	3.8508
Palace	8327.430	3.4910
Yaght	9262.862	3.2850
Sailboat	10171.02	2.9726

4.8. Comparison results

The proposed method is compared to other algorithms in order to see if it satisfies general requirements and how well it performs. Lena is used as the test image. Entropy, key space, NPCR, UACI, and correlation coefficient are utilized for the comparison. By looking at the algorithms, the suggested method is more secure against differential, statistical, and brute-force attacks than the methods that are currently being used. The results of the comparison are listed in Table 7.

Table 7. Comparison results of the proposed method with other methods

Image	method	Key space	Entropy	NPCR	UACI	Correlation.H	Correlation.V	Correlation.D
Encrypted Lena	proposed	2^{627}	7.9985	99.6367	33.0305	-0.00020	-0.00045	-0.00474
	Ref. [15]	2^{299}	7.9984	99.6094	33.4635	-0.00216	0.00103	0.0004
	Ref. [16]	2^{425}	7.9912	99.6235	33.4620	0.000617	-0.00053	-0.00041
	Ref. [17]	4×10^{12}	7.9974	99.62	31.03	-0.0026	0.0031	-0.0043
	Ref. [20]	2^{260}	7.929	99.65	32.4966	-0.0921	-0.0372	-0.1013
	Ref. [28]	-	7.9976	99.64	28.66	-	-	-

4.9. Time consuming

The execution time is also an important factor in evaluating the performance of an encryption algorithm. The encryption algorithm is more efficient when the execution time is less. Table 8 shows the average consumption time for two-stage encryption and decryption.

Table 8. Time-consuming for encryption/decryption technique

Image size	Encryption Time (sec)	Decryption time (sec)
256×256	0.58233	0.79361
512×512	1.29370	1.36520

5. CONCLUSION

In this study, a robust color image encryption algorithm built on a novel chaotic four-dimensional system has been suggested to achieve a high level of security, fast speed, and high performance. A variety of security analyses, including entropy analysis and histogram analysis, have been performed on the proposed scheme. The results of the simulations and performance measurements have shown that the encryption effect on security and reliability of our suggested method is good and it is well suited for image encryption. The proposed method has a large key space of (2^{627}) and a high sensitivity to a slight change in the secret key of decrypted images, a fairly uniform histogram, and entropy values close to the ideal value of 8, mean values for NPCR and UACI values of (99.63) and (33.03), respectively, and a very short time for encryption and decryption. In future work, the proposed method could be applied to other types of data, such as video and audio data, also by merging the proposed system with other techniques like watermarking or steganography techniques.

REFERENCES

[1] Y. Luo, J. Yu, W. Lai, and L. Liu, "A novel chaotic image encryption algorithm based on improved baker map and logistic map," *Multimedia Tools and Applications*, vol. 78, no. 15, pp. 22023–22043, Aug. 2019, doi: 10.1007/s11042-019-7453-3.

[2] S. Sharma, T. Kumar, R. Dhaundiyal, A. K. Mishra, N. Duklan, and A. Maithani, "Improved method for image security based on chaotic-shuffle and chaotic-diffusion algorithms," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, pp. 273–280, Feb. 2019, doi: 10.11591/ijece.v9i1.pp273-280.

[3] M. S. Croock, S. D. Khuder, and Z. A. Hassan, "Self-checking method for fault tolerance solution in wireless sensor network," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 4, pp. 4416–4425, Aug. 2020, doi: 10.11591/ijece.v10i4.pp4416-4425.

[4] M. Liu and G. Ye, "A new DNA coding and hyperchaotic system based asymmetric image encryption algorithm," *Mathematical Biosciences and Engineering*, vol. 18, no. 4, pp. 3887–3906, 2021, doi: 10.3934/mbe.2021194.

[5] A. Gupta, D. Singh, and M. Kaur, "An efficient image encryption using non-dominated sorting genetic algorithm-III based 4-D chaotic maps," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1309–1324, Mar. 2020, doi: 10.1007/s12652-019-01493-x.

[6] C. Han, "An image encryption algorithm based on modified logistic chaotic map," *Optik*, vol. 181, pp. 779–785, Mar. 2019, doi: 10.1016/j.ijleo.2018.12.178.

[7] S. N. Prajwalasimha and L. Basavaraj, "Performance analysis of transformation and Bogdonov chaotic substitution based image cryptosystem," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 188–195, Feb. 2020, doi: 10.11591/ijece.v10i1.pp188-195.

[8] A. T. Hashim and B. D. Jalil, "Color image encryption based on chaotic shit keying with lossless compression," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 5736–5748, Dec. 2020, doi: 10.11591/ijece.v10i6.pp5736-5748.

[9] F. Q. A. Al-Yousuf and R. Din, "Review on secured data capabilities of cryptography, steganography, and watermarking domain," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 17, no. 2, pp. 1053–1058, Feb. 2020, doi: 10.11591/ijeecs.v17.i2.pp1053-1058.

[10] Y. Niu, Z. Zhou, and X. Zhang, "An image encryption approach based on chaotic maps and genetic operations," *Multimedia Tools and Applications*, vol. 79, no. 35–36, pp. 25613–25633, Jul. 2020, doi: 10.1007/s11042-020-09237-2.

[11] S. A. Mehdi and A. A. Kadhim, "Image encryption algorithm based on a new five dimensional hyperchaotic system and sudoku matrix," in *2019 International Engineering Conference (IEC)*, Jun. 2019, pp. 188–193, doi: 10.1109/IEC47844.2019.8950560.

[12] Y. Xian and X. Wang, "Fractal sorting matrix and its application on chaotic image encryption," *Information Sciences*, vol. 547, pp. 1154–1169, Feb. 2021, doi: 10.1016/j.ins.2020.09.055.

[13] A. N. K. Telem, H. B. Fotsin, and J. Kengne, "Image encryption algorithm based on dynamic DNA coding operations and 3D chaotic systems," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 19011–19041, May 2021, doi: 10.1007/s11042-021-10549-0.

[14] X.-Y. Wang, P. Li, Y.-Q. Zhang, L.-Y. Liu, H. Zhang, and X. Wang, "A novel color image encryption scheme using DNA permutation based on the Lorenz system," *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 6243–6265, Mar. 2018, doi: 10.1007/s11042-017-4534-z.

[15] T. S. Ali and R. Ali, "A new chaos based color image encryption algorithm using permutation substitution and Boolean operation," *Multimedia Tools and Applications*, vol. 79, no. 27–28, pp. 19853–19873, Mar. 2020, doi: 10.1007/s11042-020-08850-5.

[16] L. Teng, X. Wang, F. Yang, and Y. Xian, "Color image encryption based on cross 2D hyperchaotic map using combined cycle shift scrambling and selecting diffusion," *Nonlinear Dynamics*, vol. 105, no. 2, pp. 1859–1876, Jul. 2021, doi: 10.1007/s11071-021-06663-1.

[17] S. Tariq, M. Khan, A. Alghafis, and M. Amin, "A novel hybrid encryption scheme based on chaotic Lorenz system and logarithmic key generation," *Multimedia Tools and Applications*, vol. 79, no. 31–32, pp. 23507–23529, Aug. 2020, doi: 10.1007/s11042-020-09134-8.

[18] K. A. K. Patro, B. Acharya, and V. Nath, "Secure multilevel permutation-diffusion based image encryption using chaotic and hyper-chaotic maps," *Microsystem Technologies*, vol. 25, no. 12, pp. 4593–4607, Dec. 2019, doi: 10.1007/s00542-019-04395-2.

[19] M. Z. Talhaoui, X. Wang, and M. A. Midoun, "Fast image encryption algorithm with high security level using the Bülban chaotic map," *Journal of Real-Time Image Processing*, vol. 18, no. 1, pp. 85–98, Feb. 2021, doi: 10.1007/s11554-020-00948-1.

[20] H. A. Abdullah and H. N. Abdullah, "FPGA implementation of color image encryption using a new chaotic map," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 13, no. 1, pp. 129–137, Jan. 2019, doi: 10.11591/ijeecs.v13.i1.pp129-137.

[21] S. Zhu and C. Zhu, "Secure image encryption algorithm based on hyperchaos and dynamic DNA coding," *Entropy*, vol. 22, no. 7, Jul. 2020, doi: 10.3390/e22070772.

[22] M. Ghazvini, M. Mirzadi, and N. Parvar, "A modified method for image encryption based on chaotic map and genetic algorithm," *Multimedia Tools and Applications*, vol. 79, no. 37–38, pp. 26927–26950, Oct. 2020, doi: 10.1007/s11042-020-09058-3.

[23] A. Girdhar and V. Kumar, "A RGB image encryption technique using Lorenz and Rossler chaotic system on DNA sequences," *Multimedia Tools and Applications*, vol. 77, no. 20, pp. 27017–27039, Oct. 2018, doi: 10.1007/s11042-018-5902-z.

[24] A. Alghafis, F. Firdousi, M. Khan, S. I. Batool, and M. Amin, "An efficient image encryption scheme based on chaotic and Deoxyribonucleic acid sequencing," *Mathematics and Computers in Simulation*, vol. 177, pp. 441–466, Nov. 2020, doi: 10.1016/j.matcom.2020.05.016.

[25] S. Zhou, X. Wang, Y. Zhang, B. Ge, M. Wang, and S. Gao, "A novel image encryption cryptosystem based on true random numbers and chaotic systems," *Multimedia Systems*, vol. 28, no. 1, pp. 95–112, Feb. 2022, doi: 10.1007/s00530-021-00803-8.

[26] F. Elamrawy, M. Sharkas, and A. M. Nasser, "An image encryption based on DNA coding and 2D Logistic chaotic map," *International Journal of Signal Processing*, vol. 3, 2018.

[27] S. A. Banu and R. Amirtharajan, "Tri-level scrambling and enhanced diffusion for DICOM image cipher- DNA and chaotic fused approach," *Multimedia Tools and Applications*, vol. 79, no. 39–40, pp. 28807–28824, Oct. 2020, doi: 10.1007/s11042-020-09501-5.

[28] A. Susanto *et al.*, "Triple layer image security using bit-shift, chaos, and stream encryption," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, no. 3, pp. 980–987, Jun. 2020, doi: 10.11591/eei.v9i3.2001.

[29] J. Zheng, Z. Luo, and Q. Zeng, "An efficient image encryption algorithm based on multi chaotic system and random DAN coding," *Multimedia Tools and Applications*, vol. 79, no. 39–40, pp. 29901–29921, Oct. 2020, doi: 10.1007/s11042-020-09454-9.

[30] J. Zhang and D. Huo, "Image encryption algorithm based on quantum chaotic map and DNA coding," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15605–15621, Jun. 2019, doi: 10.1007/s11042-018-6973-6.

[31] M. G. A. Malik, Z. Bashir, N. Iqbal, and M. A. Imtiaz, "Color image encryption algorithm based on hyper-chaos and DNA computing," *IEEE Access*, vol. 8, pp. 88093–88107, 2020, doi: 10.1109/ACCESS.2020.2990170.

BIOGRAPHIES OF AUTHORS

Huda R. Shakir received a B.Sc. in Computer Sciences from Mustansiriyah University, Baghdad, Iraq. Currently, she is studying M. Sc. in Computer Science, College of Education, Department of Computer Science, Mustansiriya University, as well as working in the Iraqi Ministry of Education. She can be contacted at hudarashid@uomustansiriyah.edu.iq.

Sadiq A. Mehdi received the B.Sc. in Mathematical Sciences from Mustansiriyah University, Baghdad, Iraq in 1996, M.Sc. in Applied Mathematics Sciences-Modeling and Simulation from Al al-Bayt University, Jordan in 2002, and Ph.D. in Applied Mathematics/Data Cryptography from University of Mustansiriyah, Baghdad, Iraq in 2011. Current position and Functions: computer science from Mustansiriyah University. His research interest is in the fields of Dynamical system, Chaotic system, Chaotic Encryption and Modeling and Simulation. He can be contacted at sadiqmehdi71@uomustansiriyah.edu.iq.

Anwar A. Hattab received the B.Sc in Computer Science from Baghdad University and her MSc degree in Network Management in 2003 from the Iraq Commission for Computer and Informatics, Institute for Post Graduate Studies in Informatics. Currently she is a lecturer in computer science. Anwar has more than 18 years of experience and has supervised Msc and BSc final year project. Her research interests include cryptography, image processing, data security, network security, and databases. She can be contacted at email: anwarabbas76@uomustansiriyah.edu.iq.