International Journal of Safety and Security Engineering
Vol. 14, No. 5, October, 2024, pp. 1465-1476
Journal homepage: http://iieta.org/journals/ijsse

Enhanced Image Encryption Using a Novel Chaotic System and Scramble Dithering Technique

Omar A. Jasim[1*], Sattar Rana Amer[1], Suhad Fakhri Hussein[1], Sadiq A. Mehdi[2]

[1] The Ministry of Education, Rusafa First, Baghdad 10045, Iraq
[2] Computer Science Department, College of Education, Mustansiriyah University, Baghdad 10045, Iraq

Corresponding Author Email: omar.abdul.ghafoor83@gmail.com

https://doi.org/10.18280/ijsse.140514

Received: 2 August 2024
Revised: 4 October 2024
Accepted: 17 October 2024
Available online: 31 October 2024

Keywords:
hyperchaotic system, dithering, image encryption, confusion, diffusion, 4-dimensional chaos

ABSTRACT

In the past years, the importance of using chaotic systems in encrypting media has increased, for example, media (text, images, videos, etc.). To protect these media from the increasing hacking risk and modification, it is necessary to use highly secure and efficient encryption methods to repel these attacks over the network, due to the strength that chaotic systems provide in any encryption system in which they are used. In this article, a cipher algorithm for 2D images is proposed. The proposed cryptosystem includes a new 4D chaotic system and a new permutation method for the image using the proposed dithering box [256]. The proposed chaotic system is used for the dynamic permutation of pixels depending on the value from the chaotic system then it applies XOR operations on the input image data (a dynamic stream cipher). The proposed permutation method is used to encrypt each set of 256 pixels in the image data (a dynamic block cipher). The most important feature of the proposed encryption scheme is the speed of performance in encrypting data. The encryption speed of a color image (256×256) reaches 0.9946 milliseconds, and 1 that of a color image (512×512) is 1.561223 milliseconds. In addition, the proposed encryption scheme shows excellent diffusion and confusion properties. The proposed algorithm keyspace length (10^{195}) increases its defense against brute force attacks, the encrypted image analyses show entropy values up to (7.9998) indicating that the values in the image appear almost equal. The proposed scheme shows more resistance to differential attacks through the values of NPCR tests of 99.607% and UACI tests of 33.447, which are ideal values compared to modern encryption schemes. Performance analyses show that the proposed cryptosystem has excellent cipher speed and security performance.

1. INTRODUCTION

In the last two decades, the communications system has grown significantly with the development and diversity of attacks on transmitted data [1]. This has created the need to significantly change communication methods to suit the rapid development of network and computer technology. Several scholars have proposed various approaches, including highly secure methods [2].

Therefore, there is a need to develop encryption systems capable of confronting attacks on data and maintaining information. Among the algorithms capable of confronting such attacks are chaotic systems. Chaotic systems have features that make them suitable for data encryption operations, for example, images, such as randomness, parameter sensitivity, initial value sensitivity, unpredictability, and ergodicity. They also provide speed in generating keys in encryption algorithms [3]. It is preferable to use high-dimensional chaotic systems in encryption systems because they have excellent randomness properties that help encrypt data in a secure, efficient, and attack-resistant manner [4].

In 2020, Lu et al. [5] used an efficient new algorithm for image encryption based on a new discrete combined chaotic maps, LSS, and a single S-box was proposed, which, when compared to Sine and Logistic chaotic systems, offers a wider chaotic range and superior chaotic performance. The new chaotic system is used to build a powerful S-Box. The new encryption method consists of substitution and permutation operations, In the encryption procedure, a key-associating approach for image content is used. This method can have the effect of a "one-time pad" and allows the algorithm to withstand a specific plaintext-attack (CPA) [5].

In 2020, Zhang [6] proposed algorithm for image encryption based on a lifting scheme and chaotic system. The proposed chaotic system uses pseudo-random sequences generated with a secret key to make changes to the generated keychain to increase security, and then alter the image's approximation components. In the lifting scheme, the original image is divided into odd and even indexed sequences. Using these two sequences as a basis, the high-frequency components and the low-frequency components of the image are achieved by procedures on prediction and updating. The key stream for the suggested system only requires a pseudo-random sequence that is half the size of the image.

In this paper, a new 2D algorithm for image encryption is proposed. The proposed scheme includes a new 4D chaotic system and a new image permutation method using the proposed dithering box [256] (dynamic block cipher). The results of security and statistical analysis prove the efficiency of the proposed scheme in media encryption and it can also be used in data encryption in 5G communications.

This article consists of seven sections, the second section includes related works, the third section includes an overview of the proposed cryptosystem, the fourth section illustrates the analysis and discussion of results, the fifth section includes a conclusion and recommendations, and the final section includes references.

2. RELATED WORKS

After evaluating and analyzing various algorithms for image encryption based on chaotic systems and new permutation methods, we can summarize the findings as follows:

In 2019, Farah et al. [7] proposed a new algorithm of image encryption using fractional Fourier transform (FRFT), the DNA sequence operation, and the chaos of the Lorenz system. For the first time, they used optical transform with a DNA sequence approach for image encryption. Chaos is used for permuting the positions of pixels. Using a DNA sequence, the XOR operation scrambles the pixel values of the plain image. Three times the FRFT was used. As a result, the fractional order functions as a reliable key for blind decryption. The FRFT scheme requires more time in its algorithm.

In 2019, Wang et al. [8] proposed a new cryptosystem for an image encryption algorithm based on the LL complex chaotic system and ZigZag transform. The proposed Lü system and the ZigZag transform method are used together to scramble the plain image block sub-channel. Moreover, by sorting the pixels with identification values (scrambling algorithm) using the Lü system and the new hybrid chaotic system (logistic chaotic map), the pixel values are destroyed, and adjacent-side XOR operations are applied for confusion. The amount of complexity in the algorithm is good but at the expense of execution time.

In 2021 Budiman et al. [9] suggested a scheme for encryption that combines two chaotic approaches with two hash functions. To establish local encryption for each image zone, the first chaos method uses rotation and zoning algorithms based on plain text and key hash functions. Additionally, a logistic map is used to carry out the image's total encryption. In the first stage of this encryption model, the chaos method in each image zone is used for the confusion approach, and in the second step, the chaotic system is used for the diffusion approach. The researchers did not indicate the encryption time for the proposed method.

In 2021, Khalil et al. [10] proposed an algorithm of image encryption for color / grayscale based on a hybrid 2-dimensional chaotic map combined (a cosine cross and sine chaotic map). In the confusion phase, a hybrid 2-dimensional chaotic map is used to scramble the pixels of the image. Then, a chaotic self-diffusion matrix is created using a 1-dimensional combined Logistic chaotic with XOR processes applied to the image. The article does not mention the implementation time of the encryption system.

In 2021, Wang and Zhang [11] proposed a new chaotic system with μ and λ parameters for controlling its range and performance. The proposed chaotic system is one-dimensional, so multi-dimensional chaotic maps are more complex than one-dimensional maps.

In 2022, Shakir et al. [12] proposed A new algorithm for image encryption based on a DNA encoding and a four dimensional-hyperchaotic system. Firstly, they applied permutation operations on the pixel positions using key sequences generated from a four-dimensional-hyperchaotic system. Secondly, diffusion operations using DNA operations (like DNA XOR, DNA addition, and shift to the right, and left) change the pixel values. The proposed encryption system has effective encryption features. But the time for encryption and decryption of the proposed system is not mentioned.

3. METHODOLOGY

In this article, we will use a mathematical model that represents a novel four-dimensional chaotic system. The novel autonomous system in four dimensions can be acquired as follows:

$$\begin{aligned} \frac{dx}{dt} &= -ax + byz - czw - dyCos(w) \\ \frac{dy}{dt} &= -y - xz + exw - xzw \\ \frac{dz}{dt} &= -fz - xy - xw + xyw(1) \\ \frac{dw}{dt} &= gxy + xz + hyz - iyCos(w) \end{aligned} \tag{1}$$

The states of the system are x, y, z, and w, with $t \in \mathfrak{R}$. The positive parameters of the system, a, b, c, d, e, f, g, h, and i, are defined as follows. A chaotic attractor is observed in the new four-dimensional chaotic system (1) when the system parameter values are selected as a = 24, b = 20, c = 5, d = 10, e = 3.1, f = 2.5, g = 26, h = 0.8, and i = 9. The initial conditions are set to x(0) = 2.5, y(0) = 1, z(0) = 1.5, and w(0) = 2.

Lyapunov exponents and Lyapunov dimensions

The four Lyapunov exponents of the nonlinear dynamical system (1) with the above parameters were values $L_1 = 1.95247, L_2 = 1.05247, L_3 = 0, and\ L_4 = -22.6213$.

It can be said that this system has chaotic properties when at least one Lyapunov exponent is positive. In the proposed system, L_1 and L_2 are positive and L_3 and L_4 are negative. The Lyapunov dimension is (3.13284) for this novel chaotic system.

Phase portraits

This nonlinear system displays a lot of chaotic dynamics characteristics and complex and strange attractors, as shown in Figure 1 using the MATHEMATICA program. From Figure 1, it can be seen that the topology resembles the shape of a flying butterfly flapping its wings, which refers to the term "butterfly effect".

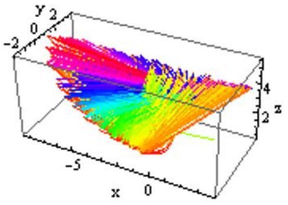


Figure 1. Three-dimensional vision and chaotic attractors

Sensitivity to initial conditions

Figure 2 demonstrates that the chaos trajectories are highly sensitive to even slight changes in initial conditions. The initial values are set as x(0) = 2.5, y(0) = 1, z(0) = 1.5, and w(0) = 2 for the solid line; and x(0) = 2.5, y(0) = 1, z(0) = 1.500000000000001, and w(0) = 2 for the dashed line, as shown in Figure 2.

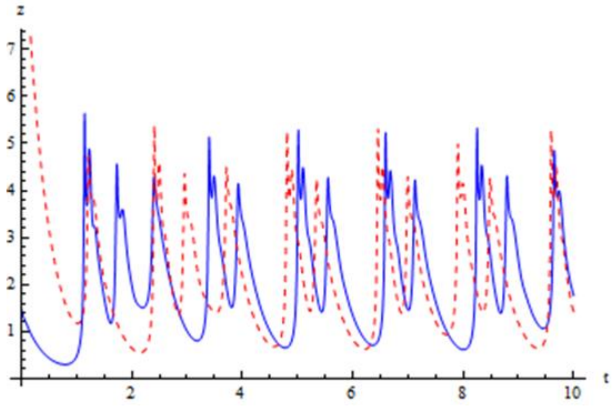


Figure 2. Sensitivity evaluations of a novel system z(t)

Bifurcation diagram

The bifurcation diagram with $f \in [2.1,2.3]$, shows that as f is increased, there is a period-doubling in the region $2.1 \leq f \leq 2.3$, shown in Figure 3.

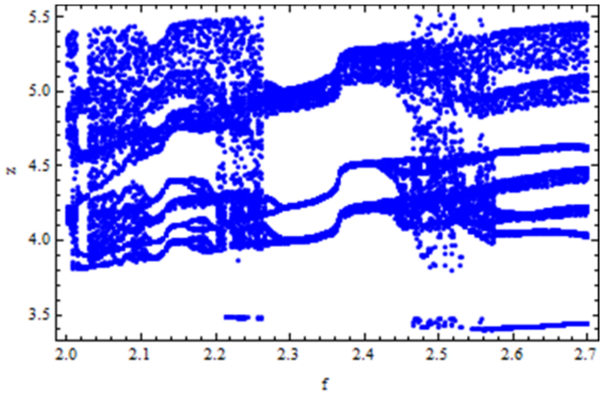


Figure 3. Bifurcation diagram

3.1 Proposed dynamic dithering matrix method

In this article, a dithering value of m=1/256 is used [13], as shown in Figure 4. It is known that the Dithering Matrix is used to remove noise from images in image processing. This algorithm will be used as an S-box in the permutation operations for image encryption because it contains values [0 to 255] that correspond to the values of any 2D image.

It is known that the dithering matrix is used to remove noise from images in image processing. This algorithm will be used as an S-box in the permutation operations for image encryption because it contains values [0 to 255] that correspond to the values of any 2D image $(x_n + 1)$.

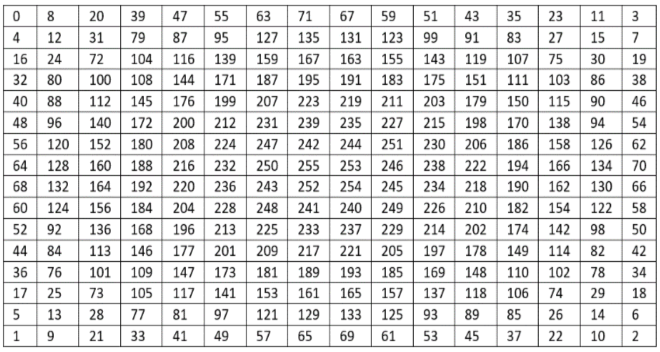
In this paper, the dynamic dithering matrix is proposed as a rearrangement of the dithering matrix based on the value from the proposed chaotic system $(x_n + 1)$, which will be used in the encryption operations of two-dimensional images.

Steps to create a Dynamic Dithering Box:

Step 1: Convert the ordered dithering matrix from 2D to 1D:

$$dith\ [k] \leftarrow dithering\ [i][j] \tag{2}$$

where, $0 \leq i, j \leq 8, 0 \leq k \leq 256$.



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

Figure 4. Dither matrix

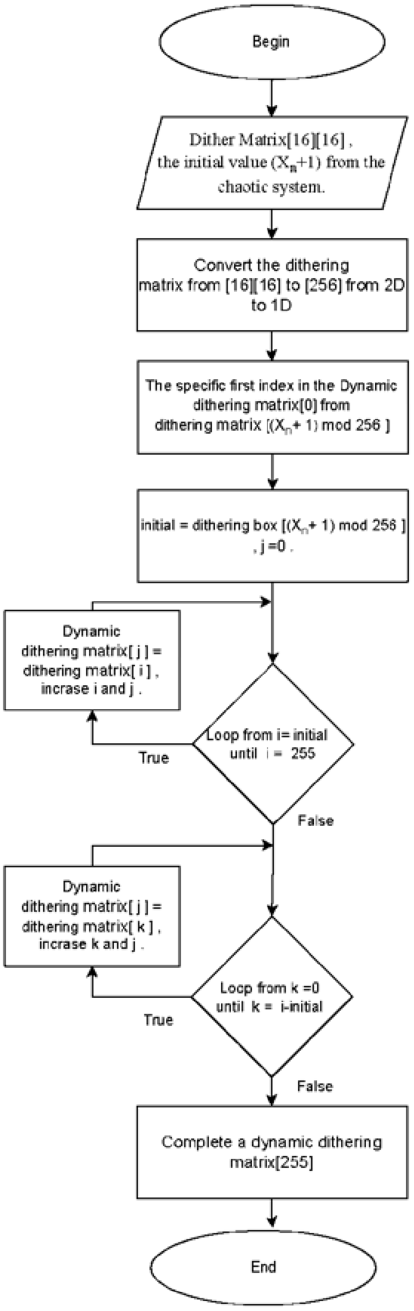

Begin
Dither Matrix[16][16] , the initial value (X_n+1) from the chaotic system.
Convert the dithering matrix from [16][16] to [256] from 2D to 1D
The specific first index in the Dynamic dithering matrix[0] from dithering matrix [(X_n+ 1) mod 256]
initial = dithering box [(X_n+ 1) mod 256] , j =0 .
Dynamic dithering matrix[j] = dithering matrix[i] , incrase i and j .
Loop from i= initial until i = 255
True
False
Dynamic dithering matrix[j] = dithering matrix[k] , incrase k and j .
Loop from k =0 until k = i-initial
True
False
Complete a dynamic dithering matrix[255]
End

Figure 5. Flow chart for creating a dynamic dithering matrix [255]

Step 2: Determine the starting point for creating a dynamic dithering matrix depending on the value x $(x_n + 1)$ from the chaotic system:

$$\begin{gathered} intial\ Ddith = (x_n + 1) mod\ 256 \\ 0 \le n \le size\ of\ image \end{gathered} \tag{3}$$

Step 3: Create a dynamic dithering box:

$$Loop\ j = dith\ [initial\ Ddith] to\ dith\ [255] \tag{4}$$

$$Ddith\ Box[i] = dith[j], where\ 0 \le i \le 255 \tag{5}$$

Step 4: Another iteration from the dithering [0] to dithering $[initial\ Ddith]$:

$$Loop\ j = dith[0] to\ dith\ [initial\ Ddith - 1] \tag{6}$$

$$Ddith\ Box[i] = dith\ [j] \tag{7}$$

Steps to create a dynamic inverse dithering matrix:

Step 1: This step creates the dynamic inverse dithering box [256] based on the dynamic dithering box:

$$Loop\ i = Ddith\ Box[0] to\ Ddith\ Box\ [255] \tag{8}$$

$$inv - Ddith\ Box\big[Ddith\ Box[i]\big] = i \tag{9}$$

where, $0 \le n \le 255$.

Flow chart for creating dynamic dithering matrix [255] and inverse dithering matrix [255].

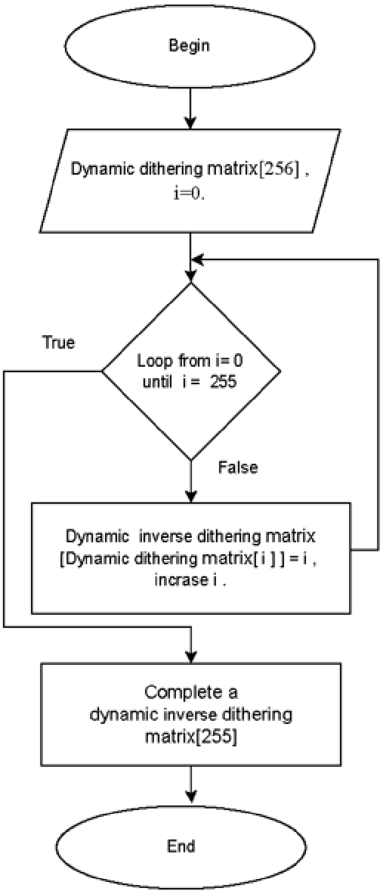

Begin
Dynamic dithering matrix[256] , i=0.
True
Loop from i= 0 until i = 255
False
Dynamic inverse dithering matrix [Dynamic dithering matrix[i]] = i , incrase i .
Complete a dynamic inverse dithering matrix[255]
End

Figure 6. Flow chart for creating a dynamic inverse dithering matrix [255]

At this stage from the proposed scheme, the dynamic dithering matrix [256] is created based on the dithering matrix [16] in Figure 4, and the inverse dithering matrix [256] is also based on the dynamic dithering matrix [256] is created, as shown in Figures 5 and 6.

3.2 Dynamic block cipher

The dynamic aspect intended in the proposed cryptosystem is to select the starting point of the fragmentation in the size of the data to be encrypted.

Encryption steps: This point is determined by the value from the proposed chaotic system.

Fetch the initial value for select start chopping for input data from the proposed chaotic system in Eq. (1).

$$\begin{gathered} initial\ point \\ = X_n + 1\ mod\ (Size\ of\ data\ for\ image\ H * W) \end{gathered} \tag{10}$$

Re-Arrange the pixel values based on the point (initial point) specified by the point above.

$$Firstseqment = imge\ [0 : initial\ point] \tag{11}$$

$$\begin{gathered} Secondseqment \\ = imge[initial\ point : (H * w) - 1] \end{gathered} \tag{12}$$

$$Tem - imge = Secondseqment + Firstseqment \tag{13}$$

Here perform permutations on image data by the Dynamic dithering proposed and XOR operations using k1 from the chaotic system $(X_n + 1)$ where $0 \le n \le size\ of\ input\ data.$

$$\begin{gathered} Loop\ Tem - imge[0]\ to\ Tem - imge\ [h * w], \\ jumper\ 256\ pixel \end{gathered} \tag{14}$$

$$Loop\ i = 0\ to\ 255 \tag{15}$$

$$\begin{gathered} Tem - imge[i] \\ = Ddith\ Box[Tem - imge\ [i]]\ XOR\ X_{n+1}\ [i] \end{gathered} \tag{16}$$

Decryption steps:

Step 1: The decryption process first fetches the initial point from the proposed chaotic system in Eq. (1).

$$\begin{gathered} initial\ point = X_n + 1\ mod \\ (Size\ of\ data\ for\ image\ H * W) \end{gathered} \tag{17}$$

Step 2: Here perform re-permutations on image data by the Dynamic inverse dithering proposed and the values of the chaotic system $(X_n + 1)$ where $0 \le n \le size\ of\ input\ data.$

$$\begin{gathered} Loop\ Tem - imge[0]\ to\ Tem - imge\ [h * w], \\ jumper\ 256\ pixel \end{gathered} \tag{18}$$

$$\begin{gathered} Tem - imge[i] = \\ Inv - Ddith\ Box[Tem - imge\ [i]]\ XOR\ X_{n+1}\ [i] \end{gathered} \tag{19}$$

$$Loop\ i = 0\ to\ 255 \tag{20}$$

Step 3: Return the pixel values to original location based on the (initial point) specified from proposed chaotic system.

$$\begin{gathered} Secondpart \\ = abs(Lengh(imge) - initial\ point) \end{gathered} \tag{21}$$

$$Firstseqment = imge\ [Secondpart : (H * w) - 1] \tag{22}$$

$$Secondseqment = imge[0 : Secondpart\ ] \tag{23}$$

$$Image = Secondseqment + Firstseqment \tag{24}$$

3.3 Dynamic stream cipher stage

This stage includes determining the starting point of the stream cipher based on a value from the proposed cryptosystem ($Y_n + 1$). This dynamic procedure represents the strength of the proposed cryptosystem.

Fetch the value from the proposed chaotic system generated from ($Y_n + 1$) ,then use the modulo to select a specific pixel in the input image.

$$specific\ pixel = (Y_n + 1)\ mod\ (H * w), \\ H\&\ w\ for\ size\ image \tag{25}$$

Apply confusion operations to the input image using ($Y_n + 1$), represent k_2

$$Loop\ i = specific\ pixel\ to\ H * W \tag{26}$$

$$imge[i] = imge[i] XOR\ k_2\ [i] \tag{27}$$

$$Loop\ i = 0\ to\ specific\ pixel \tag{28}$$

$$imge[i] = imge[i] XOR\ k_2\ [i] \tag{29}$$

3.4 Encryption algorithm for proposed cryptosystem

Figure 7, shows the stages of applying the proposed image encryption algorithm on colored images. Initially, the plain image is split into three R, G, and B channels, after which the channels are converted into R, G, and B vectors.

The proposed novel chaotic system fetches two keys (K1 and K2), the length of each depends on the size of the image ($N \times M$) and it also fetches two values ($Xn + 1$ and $Yn + 1$).

The process of generating the dynamic dithering box begins depending on the value of ($Xn + 1$), see Eqs. (3)-(7).

The rearranging of the first operations of the pixel locations in each R, G, and B vectors depends on the value of ($Xn + 1$) of the proposed chaotic system, see Eqs. (10)-(13).

The dynamic block cipher operations are parallel executed for each R, G, and B vectors using the proposed dynamic dithering box and Xored key 1. See Eqs. (14)-(16).

The rearranging of the second operations of the pixel locations in each R, G, and B vectors depends on the value of ($Yn + 1$) of the proposed chaotic system, see Eqs. (21)-(24).

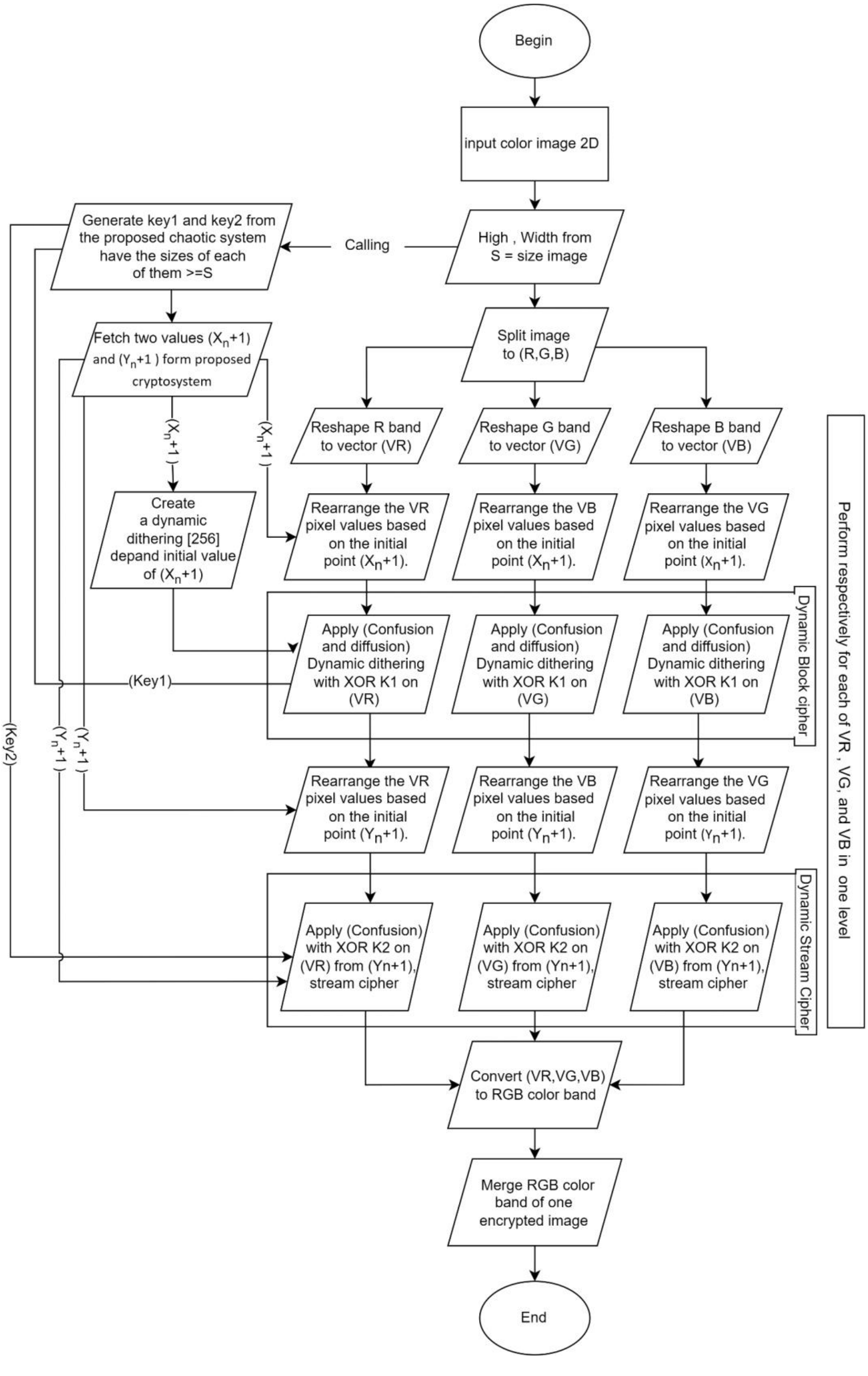
The dynamic stream cipher operations are parallel to each RGB, from the previous step, using R, B, and G vectors Xored with key 2, see the Eqs. (25)-(29).

Finally, each R, G, and B vectors are converted to R, G, and B channels and the three channels into one encrypted image, as shown in Figure 7.

Encryption algorithm

```
Input: original image ( $image_{R,G,B}$ ) has size $h \times w \times 3$ , initial conditions ( $x_0, y_0, z_0,$ and $w_0$ ) and parameters ($a, b, c, d, e, f, g, h$ and $i$)
Output: encrypted image ( $En - image_{R,G,B}$)
Begin
Read image ($image_{R,G,B}$).
R, G, and B ← Split ($image_{R,G,B}$) image for three colored bands
H ← height of the image, W ← width of the image, S ← H * W, the Size input image
Reshape R, G, and B bands to three vectors (VR, VG, and VB)
Create four chaotic sequences ($x_i, y_i, z_i,$ and $w_i$) using the iteration chaotic system (1) to obtain (k1, k2) and the sizes of each of them $>= S$
Fetch two values ($X_n + 1$ ) and ($Y_n + 1$) from the proposed cryptosystem
Create a dynamic dithering box is called $DdithBox$ [256]
$initial\ Ddith = (x_n + 1)\ mod\ 256\ , 0 \le n \le size\ of\ image$
$Loop\ j = dith\ [initial\ Ddith] to\ dith\ [255]$
$Ddith\ Box[i] = \ dith[j], where\ 0 \le i \le 255$
$Loop\ j = dith[0] to\ dith\ [initial\ Ddith - 1]$
$Ddith\ Box[i] = dith\ [j]$
Rearrange the pixel values based on the initial point ($X_n + 1$).
$Firstseqment = image_{VR,VG,VB}\ [0 : \ initial\ point]$
$Secondseqment = image_{VR,VG,VB}[initial\ point : (H * w) - 1]$
$Tem - image = \ Secondseqment + Firstseqment$
Apply Equation Step Confusion and diffusion Dynamic dithering with XOR K1 On three vectors (VR, VG, and VB) are respectively
$Loop\ Tem - image_{VR,VG,VB}[0]\ to\ Tem - image_{VR,VG,VB}\ [h * w]\ , jumper\ 256\ pixel$
$Loop\ i = \ 0\ to\ 255$
$Tem - image_{VR,VG,VB}[i] = DdithBox\left[Tem - image_{VR,VG,VB}\ [i]\right]\ XOR\ X_{n+1}\ [i]$
use the value from the proposed chaotic system ($Y_n + 1$) ,then use the modulo to select a specific pixel to stared stream cipher.
$specific\ pixel = (Y_n + 1)\ mod\ (H * w), H\&\ w\ for\ size\ image$
Rearrange the pixel values based on the initial point ($X_n + 1$).
$Firstseqment = Tem - image_{VR,VG,VB}\ [0 : initial\ point]$
$Secondseqment = Tem - image_{VR,VG,VB}[initial\ point : (H * w) - 1]$
$Tem - image = \ Secondseqment + Firstseqment$
```

```
Apply confusion operations to the input image using ($Y_n + 1$ ), and represent $k_2$ respectively
$Loop\ i = specific\ pixel\ to\ H * W$
$Tem - image_{VR,VG,VB}[i] = Tem - image_{VR,VG,VB}[i] XOR\ k_2\ [i]$
$Loop\ i = 0\ to\ specific\ pixel$
$Tem - image_{VR,VG,VB}[i] = Tem - image_{VR,VG,VB}[i] XOR\ k_2\ [i]$
Reshape VR, VG, and VB vectors to three (R, V, and B) color bands
$En - image$ ← Merge ($Tem - image_{VR,VG,VB}$) image for three colored bands
End.
```

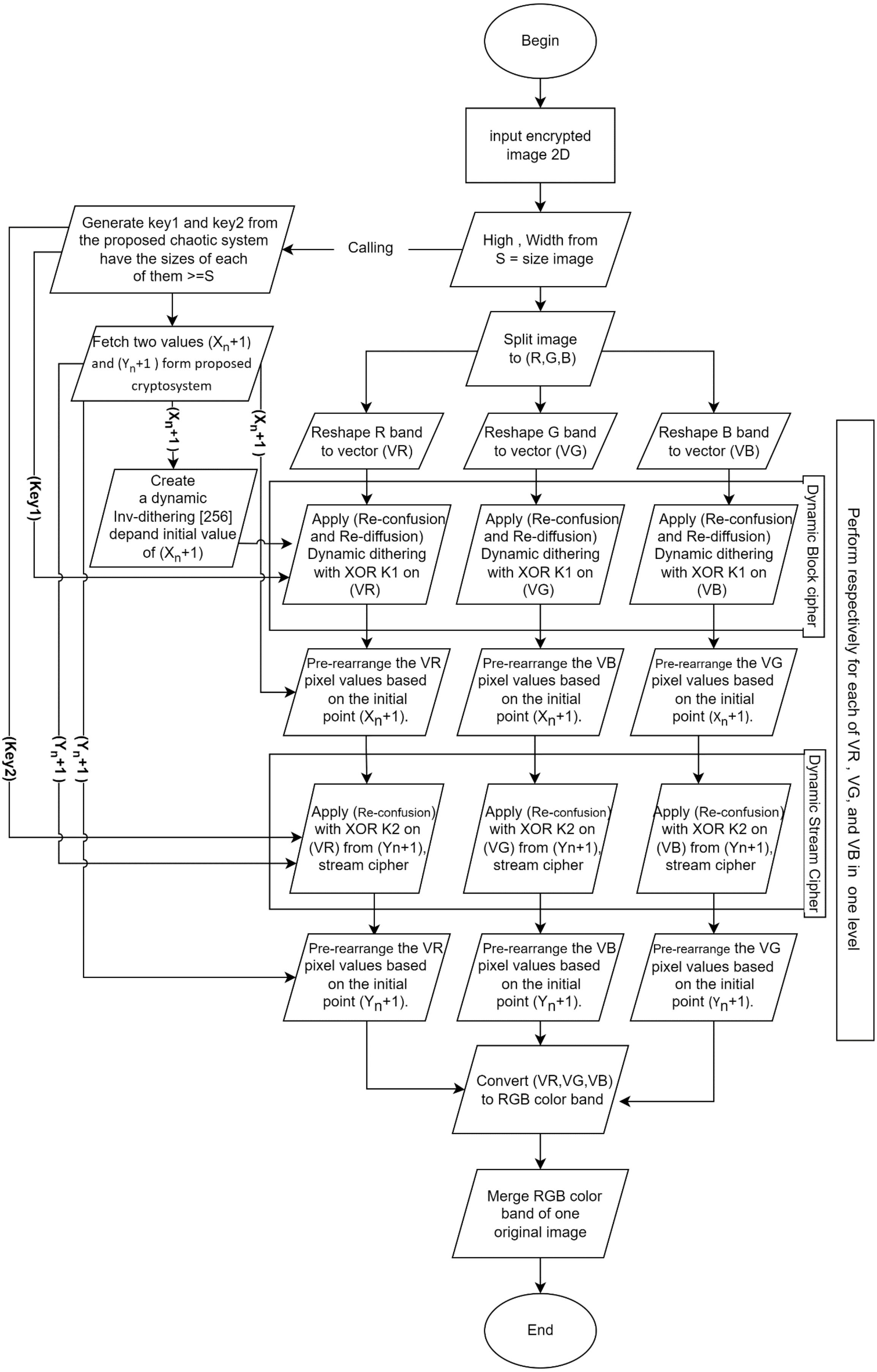


Begin
input color image 2D
High , Width from S = size image
Calling
Generate key1 and key2 from the proposed chaotic system have the sizes of each of them >=S
Fetch two values (Xn+1) and (Yn+1) form proposed cryptosystem
Split image to (R,G,B)
(Xn+1)
(Xn+1)
Reshape R band to vector (VR)
Reshape G band to vector (VG)
Reshape B band to vector (VB)
Create a dynamic dithering [256] depand initial value of (Xn+1)
Rearrange the VR pixel values based on the initial point (Xn+1).
Rearrange the VB pixel values based on the initial point (Xn+1).
Rearrange the VG pixel values based on the initial point (xn+1).
Dynamic Block cipher
Apply (Confusion and diffusion) Dynamic dithering with XOR K1 on (VR)
Apply (Confusion and diffusion) Dynamic dithering with XOR K1 on (VG)
Apply (Confusion and diffusion) Dynamic dithering with XOR K1 on (VB)
(Key1)
(Key2)
(Yn+1)
(Yn+1)
Rearrange the VR pixel values based on the initial point (Yn+1).
Rearrange the VB pixel values based on the initial point (Yn+1).
Rearrange the VG pixel values based on the initial point (Yn+1).
Dynamic Stream Cipher
Apply (Confusion) with XOR K2 on (VR) from (Yn+1), stream cipher
Apply (Confusion) with XOR K2 on (VG) from (Yn+1), stream cipher
Apply (Confusion) with XOR K2 on (VB) from (Yn+1), stream cipher
Perform respectively for each of VR , VG, and VB in one level
Convert (VR,VG,VB) to RGB color band
Merge RGB color band of one encrypted image
End

Figure 7. Flowchart for encryption steps

3.5 Decryption algorithm for proposed cryptosystem

This step involves returning the encrypted image to its original image without losing any pixel value in the Decryption image, as shown in Figure 8.



Begin
input encrypted image 2D
High , Width from S = size image
Calling
Generate key1 and key2 from the proposed chaotic system have the sizes of each of them >=S
Fetch two values (X_n+1) and (Y_n+1) form proposed cryptosystem
(X_n+1)
Create a dynamic Inv-dithering [256] depand initial value of (X_n+1)
(Key1)
(Key2)
(Y_n+1)
Split image to (R,G,B)
Reshape R band to vector (VR)
Reshape G band to vector (VG)
Reshape B band to vector (VB)
Apply (Re-confusion and Re-diffusion) Dynamic dithering with XOR K1 on (VR)
Apply (Re-confusion and Re-diffusion) Dynamic dithering with XOR K1 on (VG)
Apply (Re-confusion and Re-diffusion) Dynamic dithering with XOR K1 on (VB)
Dynamic Block cipher
Pre-rearrange the VR pixel values based on the initial point (X_n+1).
Pre-rearrange the VB pixel values based on the initial point (X_n+1).
Pre-rearrange the VG pixel values based on the initial point (X_n+1).
Apply (Re-confusion) with XOR K2 on (VR) from (Yn+1), stream cipher
Apply (Re-confusion) with XOR K2 on (VG) from (Yn+1), stream cipher
Apply (Re-confusion) with XOR K2 on (VB) from (Yn+1), stream cipher
Dynamic Stream Cipher
Pre-rearrange the VR pixel values based on the initial point (Y_n+1).
Pre-rearrange the VB pixel values based on the initial point (Y_n+1).
Pre-rearrange the VG pixel values based on the initial point (Y_n+1).
Perform respectively for each of VR , VG, and VB in one level
Convert (VR,VG,VB) to RGB color band
Merge RGB color band of one original image
End

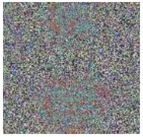
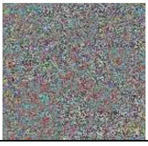
Figure 8. Flowchart for decryption steps

Decryption algorithm

```
Input: Encrypted image ($En - image_{R,G,B}$) has size $h \times w \times 3$, initial conditions ($x_0, y_0, z_0,$ and $w_0$) and parameters ($a, b, c, d, e, f, g, h$ and $i$)
Output: original image ($Dec - image$)
Begin
Read image ($image_{R,G,B}$).
R, G, and B ← split ($image_{R,G,B}$) image for three colored bands
H ← height of the image, W ← width of the image, S ← H * W, the Size input image
Reshape R, G, and B bands to three vectors (VR, VG, and VB)
Create four chaotic sequences ($x_i, y_i, z_i,$ and $w_i$) using the iteration chaotic system (1) to obtain (k1, k2) and the sizes of each of them $>= S$
Fetch two values ($X_n + 1$ ) and ($Y_n + 1$) form proposed cryptosystem
Create a dynamic inverse dithering box is called $DdithBox$ [256]
$Loop\ i = Ddith\ Box[0]\ to\ Ddith\ Box$ [255]
    $inv - Ddith\ Box[Ddith\ Box\ [i]] = 0,1,2, \dots n, 0 \leq n \leq 255$
Use the value from the proposed chaotic system ($Y_n + 1$), then use the modulo to select a specific pixel to start the stream cipher.
    $specific\ pixel = (Y_n + 1)\ mod\ (\ H * w), H\ \&\ w\ for\ size\ image$
Apply Re-confusion operations to the input image using ($Y_n + 1$ ), and k$_2$ are respectively
    $Loop\ i =\ specific\ pixel\ to\ H * W$
    $Tem - image_{VR,VG,VB}[i] = En - image_{VR,VG,VB}[i] XOR\ k_2\ [i]$
$Loop\ i = 0\ to\ specific\ pixel$
    $Tem - image_{VR,VG,VB}[i] = En - image_{VR,VG,VB}[i] XOR\ k_2\ [i]$
Pre-rearrange the pixel values based on the initial point ($Y_n + 1$).
    $Secondpart = abs(Lengh(Tem - image_{VR,VG,VB}) - initial\ point)$
    $Secondseqment = Tem - image_{VR,VG,VB}\ [Secondpart\ :\ (H * w) - 1]$
    $Firstseqment = Tem - image_{VR,VG,VB}[0\ :\ Secondpart\ ]$
    $Tem - image_{VR,VG,VB} = Firstseqment + Secondseqment$
Apply Equation Step Re-confusion and Re-diffusion Dynamic inverse dithering with XOR K1 On three vectors (VR, VG, and VB) are respectively
$Loop: Tem - image_{(VR,VG,VB)}[0]\ to\ Tem - image_{(VR,VG,VB)}\ [h * w]\ , jumper\ 256\ pixel$
    $Loop: i =\ 0\ to\ 255$
        $Tem - image_{VR,VG,VB[i]} = Inv - Ddith\ box\big[Tem - image_{VR,VG,VB}\ [i]\big]\ XOR\ X_{n+1}\ [i]$
Pre-rearrange the pixel values based on the initial point ($X_n + 1$).
    $Secondpart = abs(Lengh(Tem - image_{VR,VG,VB}) - initial\ point)$
    $Secondseqment\ = Tem - image_{VR,VG,VB}\ [Secondpart\ :\ (H * w) - 1]$
    $Firstseqment = Tem - image_{VR,VG,VB}[0\ :\ Secondpart]$
    $Tem - image_{VR,VG,VB} = Firstseqment + Secondseqment$
Pre-reshape VR, VG, and VB vectors to three (R, V, and B) color bands
$Dec - imagee_{R,G,B}$ ← Merge ($Tem - image_{VR,VG,VB}$) image for three colored bands
End.
```

4. RESULT AND DISCUSSION

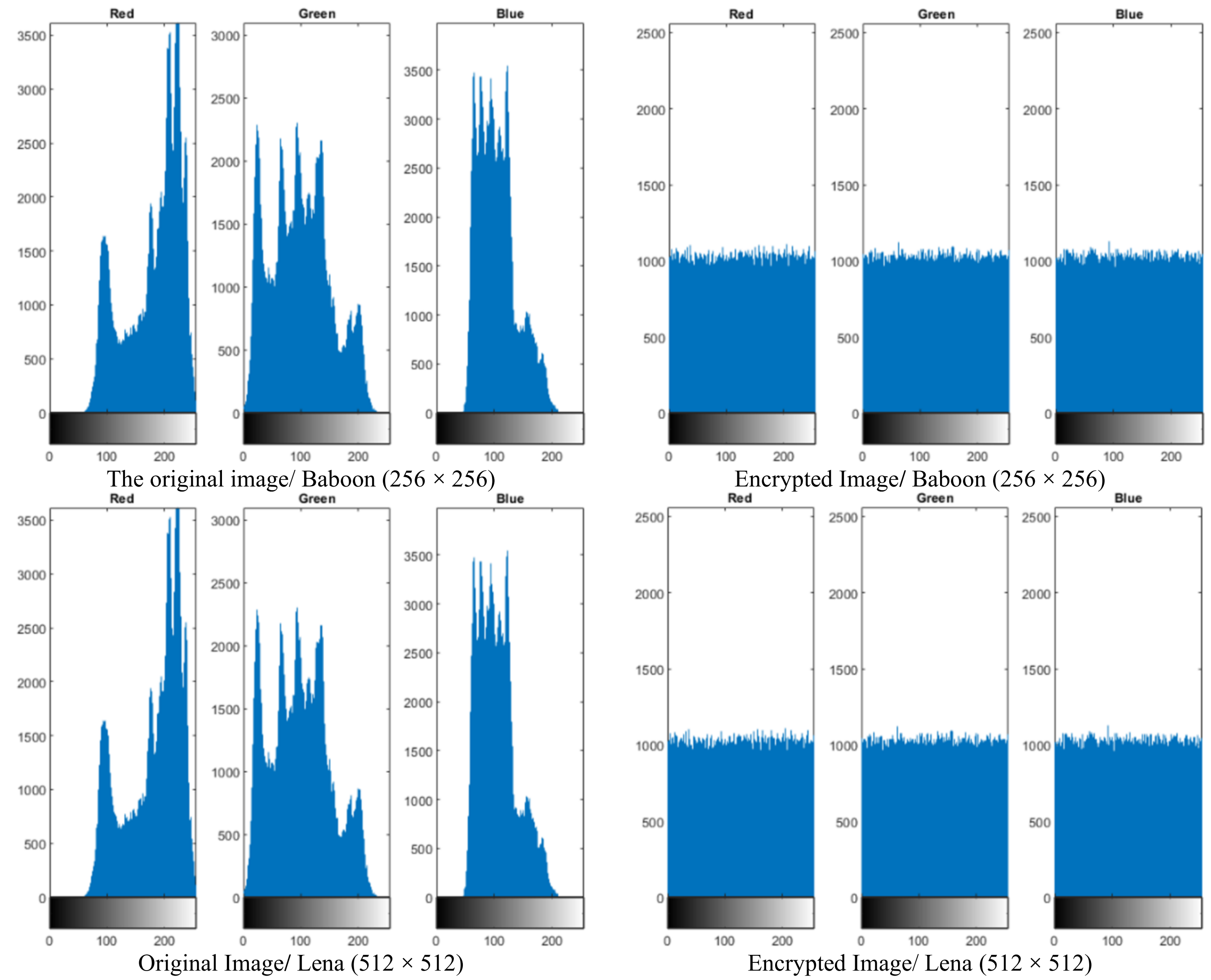
Table 1. Encrypted and decrypted image sizes (512 × 512) for 24 bits

Image Name	Plain Image	Encrypted Image	Decrypted Image
Baboon			
Lena			

We tested the encryption efficiency of the proposed algorithm of image encryption. The images used in this manuscript (such as Lena and Baboon) shown in Table 1 were simulated and examined, preferentially chosen from the USC-SIPI picture dataset, enabling fair comparisons with the currently suggested image encryption scheme based on chaotic systems. All tests showed significant differences compared to the other algorithms (Histogram, Entropy, and Correlation Analysis). This proposed algorithm uses MATLAB R2021a with an Intel® Core™ i7-13650HX at 2.60 GHz, 16 GB memory, running on Windows 11. Table 1 shows the encrypted images using the suggested cryptosystem. Visual distortion is generally acceptable since they are blurry and distorted.

4.1 Histogram analysis

The image's histogram displays the intensity of light for each pixel in (R, G, B) color mode. A uniform histogram would be an effective photo encryption test [14]. The histogram analysis of the plain and encrypted images is displayed in Figure 9.



The original image/ Baboon (256 × 256)

Encrypted Image/ Baboon (256 × 256)

Original Image/ Lena (512 × 512)

Encrypted Image/ Lena (512 × 512)

Figure 9. Histogram analysis of plain and encrypted color images

4.2 Analysis of entropy

This test is used to compute the randomness of data. The data entropy will be strong if it is near 8 if the data is made up of randomly dispersed components [15]. Table 2 shows an entropy testing comparison between the proposed image encryption algorithm and other algorithms for plain & cipher color images.

Table 2. Comparing Lena's entropy with other image encryption methods (256 × 256, 24 bits) (256 × 256, 24 bits)

Algorithm	Plain Image	Encryption Image
Proposed	7.24525	7.9998
Ref. [4]	7.75155	7.99987
Ref. [16]	7.7516	7.99857
Ref. [4]	---	7.9993
Ref. [3]	7.75155	7.9992
Ref. [7]	---	7.9991
Ref. [17]	7.3871	7.9993
Ref. [10]	---	7.9972
Ref. [11]	7.4475	7.9994

4.3 Analysis of correlation coefficients

A high degree of correlation between neighboring pixels in horizontal, vertical, and diagonal orientations is one of the key properties of an original image. The correlation test is the degree of similarity between two pixels measured [18]. Table 3 shows a correlation coefficient comparison between the proposed cryptosystem and other image encryption algorithms for plain & cipher of color images.

Table 3. Correlation comparison with plain & encryption of Lena image (24 bits)

Algorithm	Plain Correlation$_{(R.G.B)}$		
	H	V	D
Proposed	0.9896	0.995	0.982
Ref. [10]	0.96438	0.96438	0.94832
Ref. [15]	0.96438	0.98031	0.94832
Ref. [14]	-	-	-
Ref. [16]	0.9712	0.9858	0.9588
Ref. [5]	0.9964	0.9988	0.9950
Ref. [17]	0.9858	0.9801	0.9669
Ref. [9]	0.9876	0.9760	0.9626
Algorithm	**Plain Correlation$_{(R.G.B)}$**		
	H	**V**	**D**
Proposed	0.0011	-0.0011	-0.00012
Ref. [10]	0.00305	0.00305	-0.00191
Ref. [15]	-0.00020	-0.00045	-0.00474
Ref. [14]	-0.003	-0.002	0.007
Ref. [16]	0.0033	-0.0022	0.0035
Ref. [5]	0.0693	0.0610	-0.0242
Ref. [17]	0.0019	-0.0024	0.0011
Ref. [9]	-0.0010	-0.0059	0.0072

4.4 Security differential attack test

Any proposed cryptosystem should be able to repel differential attacks by testing metrics such as the Unified Average Changing Intensity (UACI) and Number of Pixels Change Rate (NPCR). The UACI values should be around 33%, while the NPCR values should be greater than 99% [10]. Table 4 shows a UACI and NPCR testing comparison between the proposed image encryption algorithm and other image encryption algorithms for plain & cipher of color images.

Table 4. UACI and NPCR analyses with other algorithms for Lena, 24 bits

Algorithm	NPCR (Avg)	UACI (Avg)
Proposed	99.607	33.447
Ref. [12]	99.607	33.366
Ref. [16]	99.6367	99.6367
Ref. [4]	99.608	33.4558
Ref. [7]	99.5677	33.4353
Ref. [3]	99.6043	33.4642
Ref. [17]	99.6113	33.4682
Ref. [10]	99.60	33.47
Ref. [11]	99.6060	33.4626

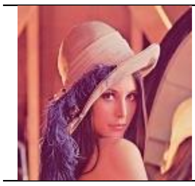
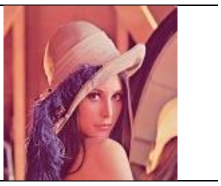
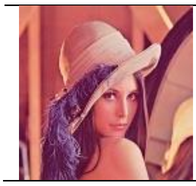
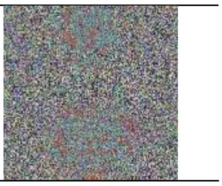
4.5 Analysis of key space

To avoid a brute-force attack, the keyspace must first be large enough [4]. At least 10^{128} bits are needed to avoid brute-force attacks. The parameters and initial conditions available for the operation of the encryption scheme mean that the keyspace size could exceed $(10^{15})^{13} = 10^{195} \approx 2^{424}$, which is greater than 2^{128}. The larger the keyspace of a cryptosystem, the more resistant the encryption is to brute-force attacks [17]. Table 5 shows a keyspace comparison between the proposed cryptosystem and other cryptosystems.

Table 5. Compares keyspace with other algorithms

Algorithm	Key Space
Proposed	10^{195}
Ref. [12]	10^{210}
Ref. [16]	10^{627}
Ref. [4]	10^{270}
Ref. [17]	10^{165}
Ref. [10]	10^{79}

Table 6. The sensitivity of the key

Plain image	Encrypted Image	Decrypted Image with x = 2.5
		
Plain Image	**Encrypted Image**	**Decrypted Image with x = 2.500000000000001**
		

4.6 Analysis of sensitivity

A good algorithm for encrypting images should be very sensitive to keys. Table 6 shows the decryption processes for Lena with a small change in the initial value (X_0). As a result, it can be said that the cryptosystem used in this research is extremely key-sensitive [4].

4.7 Analysis of MSE and PSNR Ratio.

A standard requirement for any cryptosystem technique is that the cipher image differs greatly from the original image. The plain and encrypted images may be compared using two metrics, the Peak Signal-to-Noise Ratio (PSNR), and Mean Square Error (MSE) [19].

$$MSR = \frac{1}{M \times N}\sum\nolimits_{i,j}(p_0(i,j) - p_1(i,j))^2 \tag{30}$$

$$PSNR = 20\log_{10}\left(\frac{255}{\sqrt{MSR}}\right) dB \tag{31}$$

Eqs. (30) and (31) are applied to the plain and encrypted images of the proposed encryption system, and the results as shown in Table 7.

Table 7. PSNR and MSE of Lena's image for the encryption algorithm

Proposed	The Plain and Encrypted Image	Plain and Decrypted Image
MSE	8.865950	0
PSNR	8.6536	INF

4.8 χ^2 test

Eq. (32) defines χ2, this is used to quantify how far the image deviates from a completely uniform distribution.

$$\chi^2 = \sum_{i=0}^{255}(\frac{p_i - \bar{p})^2)}{\bar{p}}) \tag{32}$$

where, $\bar{p}$ represents all pixels of an average frequency $(\bar{p} = \frac{(M\times N)}{256})$. The frequency of pixels in the image is represented by p_i.

The value of χ_2 is smaller, the more uniform the distribution of pixels in the data. Table 8, describes the values χ_2 between the plain and cipher image. The χ_2 value of the encrypted image is considerably smaller than that of the plain image [3].

Table 8. Compare χ^2 value between our proposed and using different methods for Lena's image (24 Bits)

Algorithm	Plain Image	Encrypted Image
Proposed	251294.3906	268.2487
Ref. [3]	158,880	308.0742
Ref. [11]	160,001	230.5898
Ref. [9]	--	291.2394

4.9 Complexity analysis of time

Good encryption algorithms require not only high-security performance but also need to be extremely fast [4]. The experimental environment for the test is Python 3.8, with an Intel® Core™ i7-13650HX at 2.60 GHz, 16 GB memory, running Windows 11. The proposed cryptosystem used the Lena image (512×512, 8 bits). Table 9 shows the total time consumed as well as the percentage of each process compared to other algorithms. In Table 9, generating chaotic system keys consumes 0.9578 sec, while generating dynamic dithering and inverse dynamic dithering boxes consumes 0.00014 sec, and dynamic dithering box operations consume 0.0568 seconds. In the first permutation, the pixel operation consumes 0.000278 seconds, while in the second permutation, it consumes 0.00035 seconds. The first confusion operation consumes 0.283692 seconds, while the second confusion operation consumes 0.24491 seconds. Table 10 compares the time consumption between the proposed cryptosystem and other algorithms.

Table 9. The execution time of the proposed cryptosystem

Process	Time [unit: sec]	Percentage
chaotic system keys	0.9578	62%
Gen- dynamic dith & inv-dih	0.00014	0.009%
dithering box operations	0.0568	3%
first & second permutation	0.000628	0.04%
first confusion	0.283692	18%
second confusion	0.24491	16%
Total	1.54397	100%

Table 10. Comparisons of time consumption with other algorithms [unit/sec]

Image Size	256×256	512×512
Proposed	0.9946	1.561223
Ref. [17]	1.170844	4.73389
Ref. [20]	0.22	0.85
Ref. [21]	0.498021	0.938217
Ref. [22]	5.556790	8.974393
Ref. [23]	7.73	31.59

5. CONCLUSION

This article marks the first time that an order-dithering matrix of 1/256 has been used with a chaotic system for image encryption. Before beginning the encryption steps, two highly complex keys (K1 and K2) are created using the proposed chaotic system, and then the proposed dithering matrix is reordered based on the chaotic system.

The encryption step consists of two stages. The first is called the Dynamic Block Cipher, which includes changing the locations of the pixels for each channel (R, G, B) based on the value from $(X_n + 1)$ and then applying the permutation operations using the proposed dynamic matrix, followed by an XOR operation using k1 on the pixels.

Finally, this step is called the Dynamic Stream Cipher, which involves reordering the locations of the pixels based on the value of $(Y_n + 1)$ from the proposed chaotic system, and then applying XOR operations based on K2.

The important feature of the proposed encryption algorithm is the speed of performance in encrypting data. The encryption speed of a color image (256×256) reaches 0.9946 milliseconds, and 1.561223 milliseconds for a color image (512×512). For security experiments and analyses, the keyspace length used (10^{195}) increases its resistance to brute-force attacks. Figure 9 shows the histogram analysis of the images before and after encryption, demonstrating the uniform frequency of each color in the image after encryption. The entropy analysis in Table 2 gives a value of 7.9998, which is close to 8, the ideal value for an encrypted image compared to modern algorithms in the same table.

Experiments and security analyses have proven that the proposed encryption system enjoys high protection and speed in the encryption and decryption of data. The proposed algorithm of encryption images can be used for encrypting all types of data, such as images, audio, and video, and can also be applied in 5G communications.

REFERENCES

[1] Mohammed, S.J., Mehdi, S.A. (2020). Web application authentication using ZKP and novel 6D chaotic system. Indonesian Journal of Electrical Engineering and Computer Science, 20(3): 1522-1529. https://doi.org/10.11591/ijeecs.v20.i3.pp1522-1529

[2] Zghair, H.K., Mehdi, S.A., Sadkhan, S.B. (2021). Bifurcation of novel seven-dimension hyper chaotic system. Journal of Physics: Conference Series, 1804(1): 012051. https://doi.org/10.1088/1742-6596/1804/1/012051

[3] Wang, X., Xue, W., An, J. (2020). Image encryption algorithm based on tent-dynamics coupled map lattices and diffusion of household. Chaos, Solitons & Fractals, 141: 110309. https://doi.org/10.1016/j.chaos.2020.110309

[4] Jasim, O.A., Hussein, K.A. (2021). A hyper-chaotic system and adaptive substitution box (S-Box) for image encryption. In 2021 International Conference on Advanced Computer Applications (ACA), Maysan, Iraq, pp. 144-149. https://doi.org/10.1109/ACA52198.2021.9626793

[5] Lu, Q., Zhu, C., Deng, X. (2020). An efficient image encryption scheme based on the LSS chaotic map and single S-box. IEEE Access, 8: 25664-25678. https://doi.org/10.1109/ACCESS.2020.2970806

[6] Zhang, Y. (2020). The fast image encryption algorithm based on lifting scheme and chaos. Information Sciences, 520: 177-194. https://doi.org/10.1016/j.ins.2020.02.012

[7] Farah, M.B., Guesmi, R., Kachouri, A., Samet, M. (2020). A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation. Optics & Laser Technology, 121: 105777. https://doi.org/10.1016/j.optlastec.2019.105777

[8] Wang, X., Zhang, J., Cao, G. (2019). An image encryption algorithm based on ZigZag transform and LL compound chaotic system. Optics & Laser Technology, 119: 105581. https://doi.org/10.1016/j.optlastec.2019.105581

[9] Budiman, F., Andono, P.N., Setiadi, M. (2022). Image encryption using double layer chaos with dynamic iteration and rotation pattern. International Journal of Intelligent Engineering & Systems, 15(2): 57-67. https://doi.org/10.22266/ijies2022.0430.06

[10] Khalil, N., Sarhan, A., Alshewimy, M.A. (2021). An efficient color/grayscale image encryption scheme based on hybrid chaotic maps. Optics & Laser Technology, 143: 107326. https://doi.org/10.1016/j.optlastec.2021.107326

[11] Wang, X., Zhang, M. (2021). An image encryption

algorithm based on new chaos and diffusion values of a truth table. Information Sciences, 579: 128-149. https://doi.org/10.1016/j.ins.2021.07.096

[12] Shakir, H.R., Mehdi, S.A.A., Hattab, A.A. (2022). Chaotic-DNA system for efficient image encryption. Bulletin of Electrical Engineering and Informatics, 11(5): 2645-2656. https://doi.org/10.11591/eei.v11i5.3886

[13] Kitakubo, S., Hoshino, Y., Xu, S.B. (2004). Evaluation of graininess for digital halftone images based on human visual sensitivity. In 2004 International Conference on Digital Printing Technologies, pp. 435-438.

[14] Mehdi, S.A. (2021). Image encryption algorithm based on a novel 4D chaotic system. International Journal of Information Security and Privacy, 15(4): 118-131. https://doi.org/10.4018/IJISP.2021100107

[15] Rashid, A.A., Hussein, K.A. (2023). Image encryption algorithm based on the density and 6D logistic map. International Journal of Electrical & Computer Engineering, 13(2): 1903-1913. https://doi.org/10.11591/ijece.v13i2.pp1903-1913

[16] Shakir, H.R., Mehdi, S.A., Hattab, A.A. (2023). A new four-dimensional hyper-chaotic system for image encryption. International Journal of Electrical and Computer Engineering, 13(2): 1744-1756. https://doi.org/10.11591/ijece.v13i2.pp1744-1756

[17] Luo, Y., Ouyang, X., Liu, J., Cao, L. (2019). An image encryption method based on elliptic curve ElGamal encryption and chaotic systems. IEEE Access, 7: 38507-38522. https://doi.org/10.1109/ACCESS.2019.2906052

[18] Hussein, K.A., Kareem, T.B. (2019). Proposed parallel algorithms to encryption image based on hybrid enhancement RC5 and RSA. In 2019 International Engineering Conference (IEC), Erbil, Iraq, pp. 101-106. https://doi.org/10.1109/IEC47844.2019.8950593

[19] Mehdi, S.A., Jabbar, K.K., Abbood, F.H. (2018). Image encryption based on the novel 5D hyper-chaotic system via improved AES algorithm. International Journal of Civil Engineering and Technology, 9(10): 1841-1855.

[20] Zhu, S., Zhu, C., Wang, W. (2018). A new image encryption algorithm based on chaos and secure hash SHA-256. Entropy, 20(9): 716. https://doi.org/10.3390/e20090716

[21] Zhu, S., Zhu, C., Wang, W. (2018). A novel image compression-encryption scheme based on chaos and compression sensing. IEEE Access, 6: 67095-67107. https://doi.org/10.1109/ACCESS.2018.2874336

[22] Chen, J., Zhang, Y., Qi, L., Fu, C., Xu, L. (2018). Exploiting chaos-based compressed sensing and cryptographic algorithm for image encryption and compression. Optics & Laser Technology, 99: 238-248. https://doi.org/10.1016/j.optlastec.2017.09.008

[23] ur Rehman, A., Liao, X., Kulsoom, A., Abbas, S.A. (2015). Selective encryption for gray images based on chaos and DNA complementary rules. Multimedia Tools and Applications, 74: 4655-4677. https://doi.org/10.1007/s11042-013-1828-7