Multimedia Tools and Applications (2020) 79:31739–31757
https://doi.org/10.1007/s11042-020-09551-9

# Symmetric keys image encryption and decryption using 3D chaotic maps with DNA encoding technique

Sakshi Patel[1] · Bharath K P[1] · Rajesh Kumar M[1]

Received: 26 July 2019 / Revised: 30 July 2020 / Accepted: 6 August 2020 /
Published online: 24 August 2020


**Abstract**
In present digital era, multimedia like images, text, documents and videos plays a vital role, therefore due to increase in usage of digital data; there comes high demand of security. Encryption is a technique used to secure and protect the images from unfair means. In cryptography, chaotic maps play an important role in forming strong and effective encryption algorithm. In this paper 3D chaotic logistic map with DNA encoding is used for confusion and diffusion of image pixels. Additionally, three symmetric keys are used to initialize 3D chaos logistic map, which makes the encryption algorithm strong. The symmetric keys used are 32 bit ASCII key, Chebyshev chaotic key and prime key. The algorithm first applies 3D non-linear logistic chaotic map with three symmetric keys in order to generate initial conditions. These conditions are then used in image row and column permutation to create randomness in pixels. The third chaotic sequence generated by 3D map is used to generate key image. Diffusion of these random pixels are done using DNA encoding; further XOR logical operation is applied between DNA encoded input image and key image. Analysis parameters like NPCR, UACI, entropy, histogram, chi-square test and correlation are calculated for proposed algorithm and also compared with different existing encryption methods.

**Keywords** Encryption · Decryption · 3D chaotic maps · DNA encoding · Symmetric keys · Logical operations

✉ Rajesh Kumar M
mrajeshkumar@vit.ac.in

Sakshi Patel
thesakshipatel@gmail.com

Bharath K P
bharathkp25@gmail.com

[1] School of Electronics Engineering, VIT University, Vellore, India

## 1 Introduction

Due to advancement in technology it has become easy to share information like images, data, documents, voice, videos in seconds. As this information is shared over a single band of frequency therefore it can be a questionable issue for security of personal information from the end user. Techniques used in order to secure the information plays an important role in maintaining integrity, privacy and authentication of the data to protect from unauthorized users. In this present era of technology and digitization, security is a vital aspect; therefore, encryption is one of the ways to secure the data from being hacked. Immense spreading of communication gives rise to digitized data leading to information abuse. One of the most important multimedia information carriers is images, containing critical and sensitive data. Securing these images from unauthorized users becomes the most important task in this digital world. Images play a vital role in many fields such as sharing information, authorization, Google maps, satellite, medical, military, etc. [21]. Cryptography is a method that can help in secure communication between two users. Basic four principles of cryptography are Confidentiality, Data Integrity, Authentication, and Non-Repudiation [4]. Images can be secured using different encryption algorithms, so that they can be accessed by genuine users only. Image encryption techniques are studied frequently in order to meet the demands of real-time information security when data is being transferred over internet. Traditional algorithm i.e., Data Encryption Standard (DES) has many disadvantages like low-level efficiency with large multimedia, therefore it is used for data encryption and not for multimedia [18]. There are basically two methods in encryption: pixel permutation and pixel diffusion. In pixel permutation, position of pixel is changed whereas in pixel diffusion intensity values is changed which spreads in the entire image [22].

In this paper, multilevel encryption algorithm is developed to provide several levels of security. Among various image encryption algorithms, chaos theory has a family of techniques that are good for cryptography, as it provides high speed, reasonable computation, and good security [24]. Here, 3D chaotic logistic map is used for pixel diffusion in 1st level of encryption. This method is believed to be good for multilevel security giving high chaotic behavior at certain parameters. The 3D map uses three dimensions i.e. three maps simultaneously. This map is extremely sensitive to initial conditions and its parameters. It can generate a confusion matrix that shows noisy behavior but it is exactly deterministic [3]. As the chaotic system is totally unpredictable, therefore it's output resemblance as noise. For initial conditions of the 3D chaotic map three symmetric keys are used as follows:

1 Key used for first dimension initial condition is chaotic Chebyshev map. This map generates a confusion matrix similar to logistic map with the help of certain parameters. Then logical operations are applied to get a single decimal value.
2 Initial condition for second dimension is generated using prime key. This will again generate confusion using three prime numbers; further hash function is applied to get a decimal value.
3 32 bit ASCII key is used for third dimension initial condition. Bits undergo logical operations like "bit-XOR" to generate a single decimal value.

For 2nd level of encryption, row and column pixel permutation is applied to increase randomness in image. In 3rd level pixel diffusion is done using DNA encoding technique. This technique changes the value of pixels and creates total confusion in image using encoding

rules. DNA encoding is combined with chaotic logistic map to make the encryption stronger and provide better level of security. In [28], pixels are diffused based on DNA rules, applied a random number of times. Then pixel confusion is done using 1D chaotic map. In the 4th level, logical operation is applied between the key image generated using third dimension of logistic map and DNA encoded image. Finally after four level of security encrypted image is obtained.

In 1997, chaos theory was first applied for image encryption by Fridrich, "Image encryption based on chaotic maps". Chao theory was used in order to secure large amount of data such as images or documents by applying basic steps of cryptography i.e., permutation-substitution that ensures confusion and diffusion for secure cipher [10]. M. Brindha [6], applied chaotic logistic map for multiple stages image encryption by using various functions. Six user-defined functions were used for encryption and decryption process. The pixel values were diffused using different keys for all six functions. In [20], sequence keys are used, K1 and K2 and generated third key K3 by XORing K1 with K2, the obtained K3 key is again XORed with the original image in order to obtain encrypted image. To generate these keys double chaotic logistic map equation is used and also analyzed the algorithm using histogram. Discrete wavelet transform (DWT) which uses 2D chaotic logistic map for key generation is used in [19]. Image sub-bands are created using Haar wavelet transform which are then used in encryption as well as decryption algorithm. Authors in [8] applied improved 2D chaotic system for pixel permutation and diffusion process to obtain secure encryption. 3D non-linear chaotic logistic map is used for pixel position permutation and value transformation where, row and column permutation is done, followed by logical operations in order to generate a cipher image [11]. Chaotic neural network based system is developed to control the operations of the encryption algorithm. Additionally, pixel permutation and DNA encoding rules are used for bit substitution and create confusion in image pixels [16]. Bit plane slicing is done to obtain 3D DNA matrix, which is then used permutation and diffusion of image pixels in encryption algorithm. Chaotic sequence is generated to permutate the pixels in DNA matrix. Further the matrix is divided into sub-blocks and XOR logical operation is applied with the key DNA matrix to get the final encrypted image [7]. Multiple layers encryption algorithm is developed by [5], which uses SHA-256 hash function to produce secret key for chaotic map [9], followed by pixel permutation, substitution, diffusion and DNA encoding to enhance the security of input image.

In this paper, symmetric keys image encryption and decryption using 3d chaotic maps with DNA encoding algorithm is proposed, which includes four levels of encryption process. Three keys namely Chebyshev, prime and ASCII is used to initialize the x, y and z parameters of 3D chaotic logistic map. The generated x and y sequence of the map is then used for pixel permutation of input image. The resultant image is then DNA encoded using eight complementary rules. A key image is generated by normalizing the values of z sequence of the map, and DNA encoded for further steps. Bit XOR logical operation is applied between the encoded input image and key image. The resultant image is then decoded back to decimal values in order to obtain final encrypted image.

Rest of the paper is discussed as follows; in Section 2 the chaotic logistic map is introduced. Proposed methodology and encryption decryption algorithms are discussed in Section 3. Results are shown in Section 4 and finally concluding the work in Section 5.

## 2 Chaotic logistic map

Chaos theory is widely used in emerging technologies such as neurology, cardiology, control and circuit theory, weather prediction, etc. Chaos -"when the present determine the future but the approximate present cannot approximately determine the future". In chaos even small change in initial conditions can lead to totally uncorrelated sequence. It has been said and proved that chaos functions can be used in encryption algorithms for giving good results.

Logistic function is one of the members in chaos family with high sensitivity to initial conditions and generates non periodic pseudo random sequence. It will be entirely unpredictable if proper choice of bifurcation parameter 'u' is taken into consideration. Implementing image encryption by using chaotic theory is simple, computationally faster and impregnable. Logistic map function given in Eq. 1 produces non-periodic chaotic sequences $\{x_i\}$, where $x_i$ lies between 0 and 1 and is random in nature.

$$x_{n+1} = ux_n(1-x_n) \tag{1}$$

Here 'u' is bifurcation parameter having range from 0 to 4. $x_0$ is the initial value between $0 < x < 1$ and $\{x_1, x_2, x_3 \ldots . x_n\}$ is the chaotic sequence generated by the map. It is proved that logistic system will have chaotic nature when 'u' ranges from $3.56994 < u \leq 4$ [2]. Figure 1 shows the bifurcation diagram of logistic map taking the initial condition $x_0 = 0.6$. The diagram shows the variation of u from 0 to 4 in 400 iterations, proving that the map is mostly chaotic when values are nearing 4.

In order to obtain higher level of randomness, 2D chaotic logistic map is introduced in [23]. The two dimensions of the map is shown in Eq. 2:

$$\begin{cases} x_{n+1} = u_1 x_n(1-x_n) + r_1 y_n^2 \\ y_{n+1} = u_2 y_n(1-y_n) + r_2\left(x_n^2 + x_n y_n\right) \end{cases} \tag{2}$$

The above map increases quadratic coupling in three values $x_n, y_n, y_n^2$ and $x_n^2$ and makes it complex and secure. When the parameters fall in the following range: $2.75 < u_1 \leq$

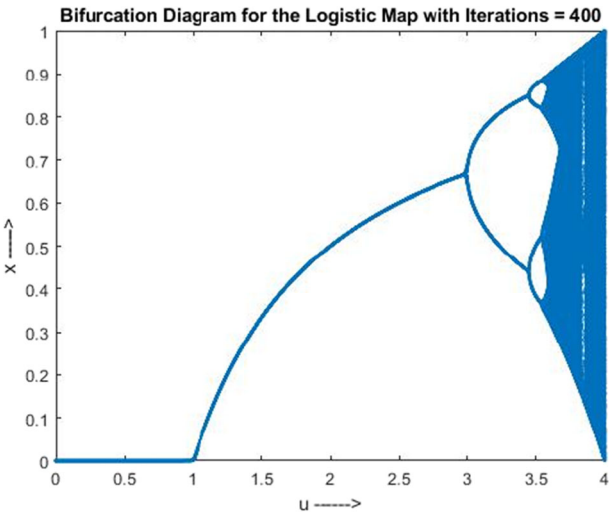


**Fig. 1** Chaotic behavior of logistic map

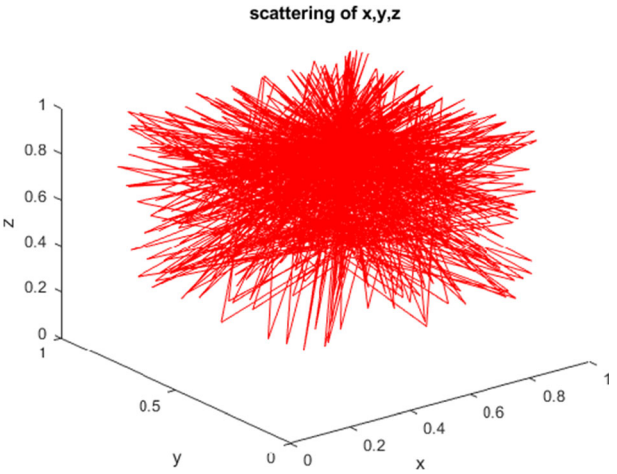


**Fig. 2** Chaotic behavior of 3D logistic map

3.4, $2.7 < u_2 \leq 3.45$, $0.15 < r_1 \leq 0.21$ and $0.13 < r_2 \leq 0.15$ the system becomes chaotic in nature from value [0, 1].

Authors in [12] extended 2D chaotic map to three dimensions in order to achieve more complexity. The 3D logistic chaotic is given in Eq. 3:

$$\begin{cases} x_{n+1} = ux_n(1-x_n) + by_n^2 x_n + az_n^3 \\ y_{n+1} = uy_n(1-y_n) + bz_n^2 y_n + ax_n^3 \\ z_{n+1} = uz_n(1-z_n) + bx_n^2 z_n + ay_n^2 \end{cases} \tag{3}$$

The 3D map further increases unpredictability as one more dimension has been added in the system to introduce chaotic nature to a greater extent. To get this chaotic behavior, parameters should have values between $3.53 < u < 3.81$, $0 < b < 0.022$, $0 < a < 0.015$ and the initial condition for x, y and z should have value between 0 and 1.

Figure 2 shows the chaos behavior of 3D chaotic logistic map taking the initial condition as x(1) = 0.2350, y(1) = 0.3500, z(1) = 0.7350, a(1) = 0.0125, b(1) = 0.0157, u(1) = 3.7700. The diagram shows randomness of x, y and z dimensions in 700 iterations. It also depicts that the map shows far more chaotic nature than 1D or 2D chaotic map.

## 3 Proposed methodology

The proposed symmetric keys image encryption using 3D chaotic maps with DNA encoding technique has four level of encryption process as follows:

Level 1. **:** Symmetric keys generation to initialize the 3D chaotic map.

a Chebyshev key
b Prime key
c ASCII key

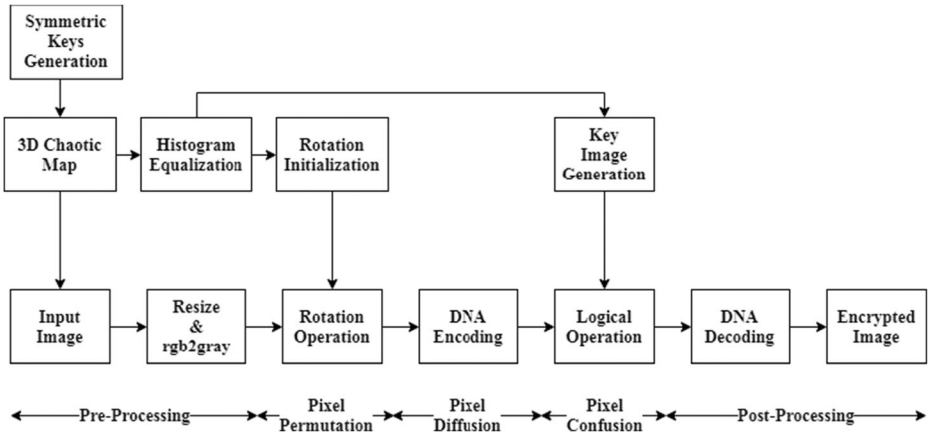


**Fig. 3** Proposed Block Diagram

Level 2. **:** Pixel permutation through 3D chaotic logistic map.

a First dimension of chaotic for row permutation.
b Second dimension for column permutation.

Level 3. **:** Pixel diffusion through DNA encoding.

Level 4. **:** Key image generation using third dimension and pixel confusion through XOR logical operation.

Figure 3 shows the proposed block diagram of the encryption process. The different levels of proposed encryption algorithm are explained in further subsections.

### 3.1 Encryption process

In this section the entire proposed encryption algorithm is explained:

#### 3.1.1 Preprocessing

This is the initial stage of the proposed algorithm, where the input image is checked for its nature and size. If the image is a color image then it is converted to gray scale and sent for further stages of encryption. If required, the image is resized to $256 \times 256$.

#### 3.1.2 Level-1: Symmetric key generation and 3D chaos generation

The proposed algorithm has three symmetric keys to initialize x, y and z dimension of 3D chaotic map. Symmetric keys in cryptography are basically those keys which are same for encryption of plain image as well as decryption of cipher image. This algorithm requires both sender and receiver to have same secret keys. Therefore, in the proposed work receiver should have same keys to decrypt the encrypted image. Three different algorithms are proposed using specific methods to

generate symmetric keys for map initialization. Each algorithm randomly generates a decimal value between 0 and 1 that helps in initializing x, y and z dimensions of 3D chaotic logistic map.

**MD5 hash function** Message Direct Algorithm 5 (MD5) is a cryptographic hash function which produces 128 bit hash value. This algorithm is widely used for security purposes as even one bit change in the key gives a significant change in the hash function generated [13]. In this work MD5 hash function is used to generate a single checksum value from 32 bits. The checksum expression is given in Eq. 4,

$$\text{checksum} = \text{mod}\left(d_1 \otimes d_2 \otimes d_3 \otimes d_4, 2^{16}\right) / \left(2^{16}-1\right) \tag{4}$$

Where, $d_1$, $d_2$, $d_3$, $d_4$ are the sequences of 8 bit each making total of 32 bits, which has to be given input to the MD5 algorithm. 'mod' is a modulo function. The final checksum value lies between 0 and 1, which is then used as the initial values for chaotic map parameters.

**Chebyshev key** This key is used to initialize the first dimension 'x' of 3D chaotic map. Chebyshev polynomial chaotic map is one of the family members of chaos maps. It uses the concept of Chebyshev polynomial which is basically a sequence of orthogonal polynomials. This map posses' chaotic behavior therefore used in such encryption algorithms [15]. Polynomial of degree n is defined in Eq. 5 [29]:

$$k_{n+1} = \cos(p * \arccos(k_n)) \tag{5}$$

Where, $k_1 \leq 1$, $2 \leq p \leq 6$, n = size of image.

**Algorithm to generate Chebyshev key**

1 Applying Chebyshev polynomial formula given in Eq. 5 using above parameters $k_1$ and p for n number of times. This map then generates a $1 \times 65{,}536$ size vector having random positive and negative decimal chaotic values.
2 To normalize the vector, each value is multiplied with 255 (highest gray level) and taken absolute of the number obtained.
3 The above vector is converted from decimal to binary values.
4 The vector is circularly shifted by one.
5 The matrix is converted back to decimal values.
6 Bit XOR logical operation is applied between the matrixes obtain in step 2 and step 5.
7 All the values obtained are added and the cumulative is converted to 32 bit binary.
8 The obtained 32 bit binary is divided into four parts and each 8 bits are converted to decimal value.
9 MD5 hash function is applied and using Eq. 4 a checksum value is obtained.

The checksum value obtained from the Chebyshev key generation algorithm is used for initializing the 'x' parameter of 3D chaotic map.

**Prime key** This key is used to initialize the second dimension 'y' of 3D chaotic map.

**Algorithm to generate prime key:** The proposed algorithm to generate prime key is executed 'n' number of times i.e. the size of input image.

1 Initialize three different prime numbers, for example: let us say p, q and m are prime numbers.
2 Multiply the first two numbers and take remainder when product is divided by 10.
3 Now 1 is added to the above obtained remainder and multiplied by the third prime number 'm'. This will be the new 'p' value for the next loop.
4 Now the first two numbers are divided and round off to floor value. Add the loop index to the number obtained. This will be the new 'q' value for the next loop.
5 After running the first three steps for n number of times, a matrix of size $1 \times 65{,}536$ is obtained.
6 All the values obtained are added and the cumulative is converted to 32 bit binary.
7 The obtained 32 bit binary is divided into four parts and each 8 bits are converted to decimal value.
8 MD5 hash function is applied and using Eq. 4 a checksum value is obtained.

The checksum value obtained from the prime key generation algorithm is used for initializing the 'y' parameter of 3D chaotic map.

**ASCII key** This key is used to initialize the third dimension 'z' of 3D chaotic map.
**Algorithm to generate ASCII key:**

1 Enter 32 bit ASCII characters.
2 The entered characters are converted into 128 bit binary value.
3 Only 32 bits are taken and divided into four parts. Each 8 bits are converted to decimal value.
4 MD5 hash function is applied and using Eq. 4 a checksum value is obtained.

The checksum value obtained from the ASCII key generation algorithm is used for initializing the 'z' parameter of 3D chaotic map.

Using these three algorithms the dimensions of the chaotic map is initialized. According to Eq. 3 the remaining parameters should have values between $3.53 < u < 3.81$, $0 < b < 0.022$, $0 < a < 0.015$. After all the values are initialized, the map is iterated for $T_1$ number of times, $T_1 > 10{,}000$ in order to obtain high chaotic values. The obtained three vectors x, y and z are normalized using equalization formula given in Eq. 6.

$$\begin{cases} x = \left\lceil mod\left((x*T_2), image\ height\right)\right\rceil \\ y = \left\lceil mod\left((y*T_2), image\ height\right)\right\rceil \\ z = \left\lceil mod\left((z*T_2), image\ height\right)\right\rceil \end{cases} \tag{6}$$

Where, $T_2$ is a large number and image height = 256 for $256 \times 256$ image size.

### 3.1.3 Level-2: Pixel permutation

Pixel position permutation is a method to reposition the pixels of an image to make it unpredictable. This process can be done randomly or by using permutation key. The proposed algorithm uses the first and second dimension generated sequence of the 3D chaotic map from Level 1 as permutation keys for row and column permutation of the

**Fig. 4** Input Matrix

I =

| | | | | |
|---|---|---|---|---|
| 4 | 2 | 3 | 1 | $R_1$ |
| 3 | 2 | 4 | 1 | $R_2$ |
| 2 | 1 | 4 | 3 | $R_3$ |
| 3 | 4 | 2 | 1 | $R_4$ |

input image respectively. The detailed explanation of the row and column permutation is given in the following subsections.

**Row permutation** To illustrate the permutation method more clearly, it is explained with an example of 4 × 4 matrixes. Let's say that the obtained chaos values of first dimension 'x' from Level 1 of size 1 × 16 is: x = [1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4].

Figure 4 shows the input matrix I. Some starting values of chaotic sequence 'x' are left unused because they are less chaotic then the later values. Now randomly values are chosen from 'x' vector and stored in variable 'k' of size [1× (number of rows in input matrix)]. This variable is the initializing vector for row permutation. For this particular example, k = [3 4 1 2] is chosen from vector x = [1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4]. The condition applied here is: If value of 'k' is even then apply right shift to row in input image, otherwise left shift. Using Eq. 7 the row permutation is applied.

$$P(i, j + k(i)) = I(i, j) \tag{7}$$

Here, P is the row permutated matrix, I is input matrix and (i, j) is row and column index of input matrix. For this example row permutation is done according to Eq. 7 as:

- The values of $R_1$ are shifted left 3 times.
- The values of $R_2$ are shifted right 4 times.
- The values of $R_3$ are shifted left 1 time.
- The values of $R_4$ are shifted right 2 times.

Final row permutated matrix is given in Fig. 5.

**Column permutation** Let's say that the obtained chaos values of second dimension 'y' from Level 1 of size 1 × 16 is: y = [4 3 2 1 4 3 2 1 4 3 2 1 4 3 2 1]. Randomly values are chosen from 'y' vector and stored in variable 'l' of size [1× (number of columns in input matrix)]. This variable is the initializing vector for column permutation. For this particular example, l = [3 2 1 4] is chosen from vector y = [4 3 2 1 4 3 2 1 4 3 2 1 4 3 2 1]. The condition applied here is: If

**Fig. 5** Row Permutated Matrix

P =

| $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|
| 1 | 4 | 2 | 3 |
| 3 | 2 | 4 | 1 |
| 1 | 4 | 3 | 2 |
| 2 | 1 | 3 | 4 |

**Fig. 6** Column Permutated Matrix

$$C = \begin{bmatrix} 3 & 4 & 3 & 3 \\ 1 & 1 & 2 & 1 \\ 2 & 4 & 4 & 2 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

value of 'l' is even then apply upwards shift to column in row permutated image, otherwise downward shift. Using Eq. 8 the column permutation is applied.

$$C(i + l(j), j) = P(i, j) \tag{8}$$

Here, C is the column permutated matrix, P is row permutated matrix and (i, j) is row and column index. For this example column permutation is done according to Eq. 8 as:

- The values of $C_1$ are shifted down 3 times.
- The values of $C_2$ are shifted up 2 times.
- The values of $C_3$ are shifted down 1 time.
- The values of $C_4$ are shifted up 4 times.

Final column permutated matrix is given in Fig. 6.

### 3.1.4 Level-3: DNA encoding

Deoxyribonucleic Acid (DNA) is a biological material found in all living body. DNA maps the nucleotide sequence to form a strand. Basically, every DNA sequence consists of four bases, A (adenine), C (cytosine), G (guanine), T (thymine). According to the rule of complementary each pair should be complementary of each other, where A binds with T and G binds with C [27]. Therefore, the complementary of the four bases works as 00 and 11, 01 and 10. These bases can have 4! = 24 combinations but only 8 are suitable for complimentary rule, as shown in Table 1. This idea of DNA is used to form a sequence code for pixel diffusion. Diffusion is a method to alter the values of image pixels and encode them in unreadable format.

For 8 bit gray scale image, the pixel value varies from 0 to 255. For example, if we randomly take any pixel value like 188, it is converted into binary bits sequence [10111100]. Randomly a DNA encoding rule is chosen from Table 1, say 7 is chosen and the sequence is encoded as CAAT. For decoding the sequence back to binary bits at the receiver side, same complementary rule is used which was used while encoding the bits.

**Table 1** Rules of DNA encoding

| | Rule1 | Rule2 | Rule3 | Rule4 | Rule5 | Rule6 | Rule7 | Rule8 |
|---|---|---|---|---|---|---|---|---|
| **A** | 00 | 00 | 01 | 01 | 10 | 10 | 11 | 11 |
| **T** | 11 | 11 | 10 | 10 | 01 | 01 | 00 | 00 |
| **G** | 01 | 10 | 00 | 11 | 11 | 11 | 01 | 10 |
| **C** | 10 | 01 | 11 | 00 | 00 | 00 | 10 | 01 |

After applying pixel permutation on the input image, DNA encoding method is applied to diffuse the pixel values using complementary rules as explained earlier. Single rule is not used for the entire image in the proposed algorithm, to make the encryption strong. For each pixel value, randomly DNA rules are chosen and encoding is applied. The random selection of rules is done using 1D chaotic logistic map given in Eq. 1. The initialization of the map is done by the Chebyshev key explained in Section 3.1.2.1. The values obtained from the chaotic map lies between 0 and 1. These values are normalized in the range 1 to 8, in order to choose eight rules of DNA for each pixel.

### 3.1.5 Level-4: XOR logical operation

In order to encrypt the pixels more and making it unpredictable, XOR logical operation is applied between the DNA encoded image and a key image. Key image is generated using the third dimension 'z' of 3D chaotic map. The chaotic sequence generated by dimension 'z' is reshaped into [M × N] matrix. Here M and N is the size of input image. DNA encoding is applied on the obtained key image using eight complementary rules as done in the 3rd level of proposed encryption algorithm. Now bit XOR logical operation is performed between the DNA encoded key image and the image obtained from Level 3. Table 2 shows the XOR operation for DNA sequences.

After applying logical operation a new DNA encoded matrix is obtained, which is then DNA decoded to decimal values. It is done by applying same DNA complementary rules on the pixels to convert them into binary bits, which can further be converted into decimal numbers. The resultant image after applying four levels of encryption is the cipher image which is transmitted over the channel to the receiver.

Flow chart of entire proposed encryption algorithm is shown in Fig. 7. To summarize, the steps for the algorithm are given below:

1 Generate three symmetric keys (Chebyshev, prime and ASCII) using specific algorithms as discussed in Section 3.1.2.
2 Initialize the 3D chaotic logistic map dimensions (x, y and z) with the generated keys and other required parameters.
3 Iterate the map for $T_1$ number of times, to get high chaotic values.
4 Equalize the values of all the dimensions in the range 0 to 255.
5 Preprocessing is done on the input image.
6 Apply pixel permutation on this image using the first two dimensions 'x' and 'y' of the chaotic map.
7 Then apply DNA encoding complementary rules in order to diffuse the image pixels.

**Table 2** XOR operation for DNA sequences

| XOR | A | T | G | C |
|---|---|---|---|---|
| **A** | A | T | G | C |
| **T** | T | A | C | G |
| **G** | G | C | A | T |
| **C** | C | G | T | A |

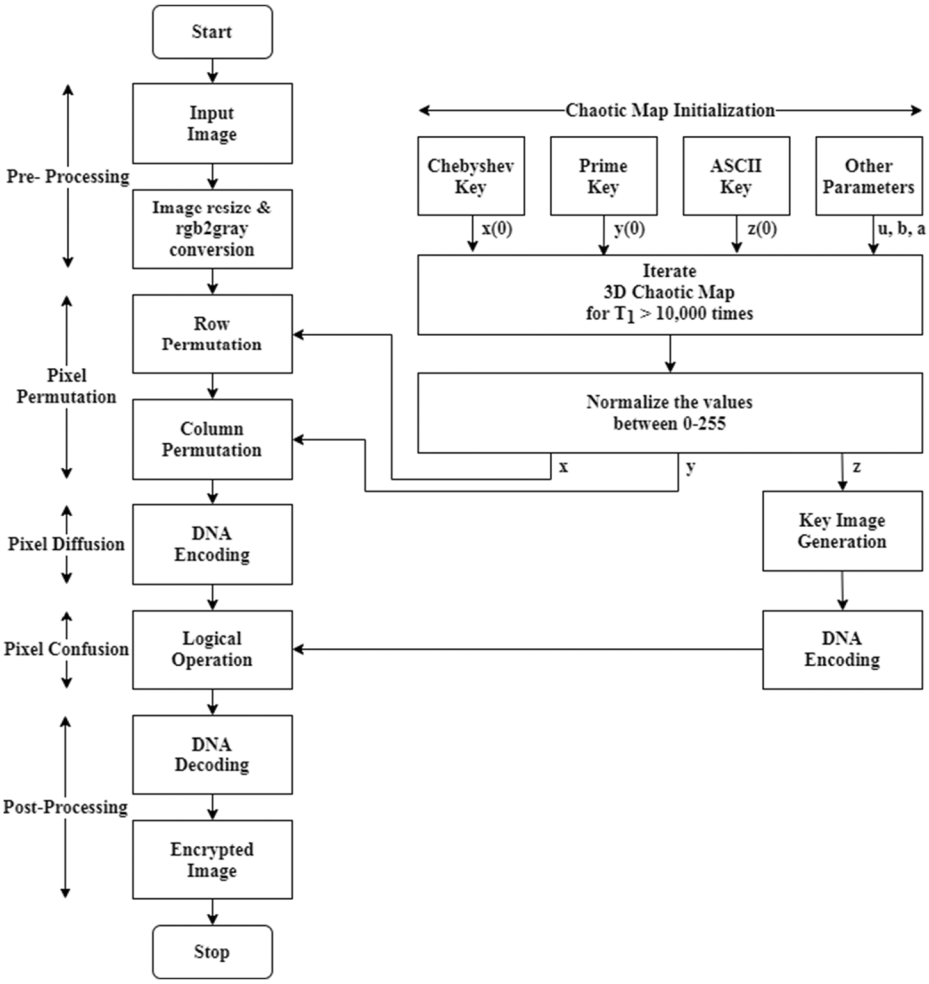


**Fig. 7** Flow Chart of Proposed Algorithm

8 Generate key image with the help of third dimension 'z' of the chaotic map and apply DNA encoding.
9 Bit XOR logical operation is applied between the encoded key image and the image obtained from Step 7.
10 Finally, cipher image is obtained after decoding the resultant of Step 9.

### 3.2 Decryption

Reverse operations are performed in order to retrieve back the image at the receiver side. As the proposed algorithm deals with symmetric keys, therefore at the receiver

side same keys are used which were applied at the transmitter side. The decryption steps are given below:

**Algorithm for decryption algorithm**

1 Generate 3D chaotic maps with the symmetric keys and other required parameters.
2 Apply DNA encoding complementary rule on both cipher image and key image.
3 Then apply logical XOR operation between the encoded cipher and encoded key image.
4 Decode the result of step 3 from DNA encoded sequence to decimal values.
5 Now apply pixel row and column permutation to retrieve back input image.

## 4 Simulation results

In the proposed work 5 grayscale test images are used with size 256 × 256. The simulation results obtained for encryption algorithm are explained in following subsections.

### 4.1 Encryption analysis

The proposed algorithm is analyzed on the 5 test images, the input and encrypted images are as shown in Fig. 8. The encrypted images are obtained by applying four level of encryption process on the input image which includes: 3D chaos generation, pixel permutation, DNA encoding and XOR logical operation. The proposed algorithm also uses three symmetric keys namely Chebyshev, prime and 32 bit ASCII, to initialize 3D chaotic map for pixel permutation stage in the algorithm. Then pixel permutation of row and column is applied using the first two dimensions 'x' and 'y' of the chaotic map. Further complementary rules of DNA encoding technique is applied to obtain diffused image.

For XOR operation, key image is generated using the third dimension 'z' of the chaotic map. This is then DNA encoded to obtain encoded key image. Confusion matrix is obtained by applying the XOR operation between encoded input and key image. Figure 8 (a) to (e) shows 5 grayscale test images and (f) to (j) shows their respected encrypted images for the proposed algorithm.

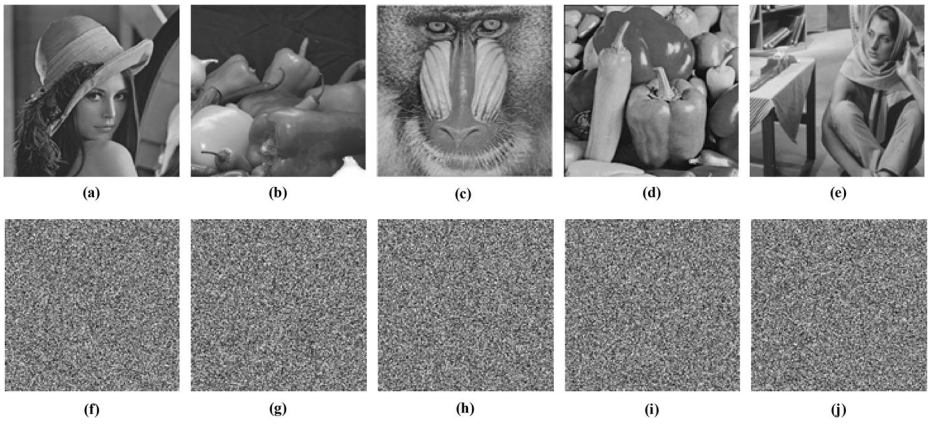


**Fig. 8** Encryption analysis: (a), (b), (c), (d), (e) are the 5 input images and (f), (g), (h), (i), (j) are 5 respective encrypted images

### 4.2 Statistical analysis

In image encryption algorithms there are serious statistical attacks from the adjacent pixels due to high correlation between them. Therefore, the statistical analysis can be performed with histogram analysis, correlation analysis, entropy analysis and chi-square test analysis. The detailed study is shown in below subsections.

#### 4.2.1 Histogram analysis

Histogram is basically a graphical plot of number of pixels at different grayscale values. A true encrypted image should have a flat histogram for all gray level values. Figure 9 (a) to (e) shows the histogram plot of the input test images and (f) to (j) shows the histogram plot of the encrypted image respectively. It shows how change in the input image pixel can bring a large difference on the histogram of the encrypted image.

#### 4.2.2 Correlation analysis

This parameter tells about the two adjacent pixels in an image, for an encrypted image these values should be as low as possible to reduce the correlation between the pixels. Table 4 shows the correlation between the horizontal, vertical and diagonal neighboring pixels in the cipher image. Figure 10 shows the correlation plot neighboring pixels of input and encrypted images. The equation for correlation is as follows:

$$r_{x,y} = \frac{cov(x,y)}{\sqrt{d_x}\sqrt{d_x}} \tag{9}$$

$$d(x) = \frac{1}{n}\sum_{j=1}^{n}\left(x_j - \frac{1}{n}\sum_{j=1}^{n}x_j\right)^2 \tag{10}$$

$$cov(x,y) = \frac{1}{n}\sum_{j=1}^{n}\left(x_j - \frac{1}{n}\sum_{j=1}^{n}x_j\right)\left(y_j - \frac{1}{n}\sum_{j=1}^{n}y_j\right) \tag{11}$$

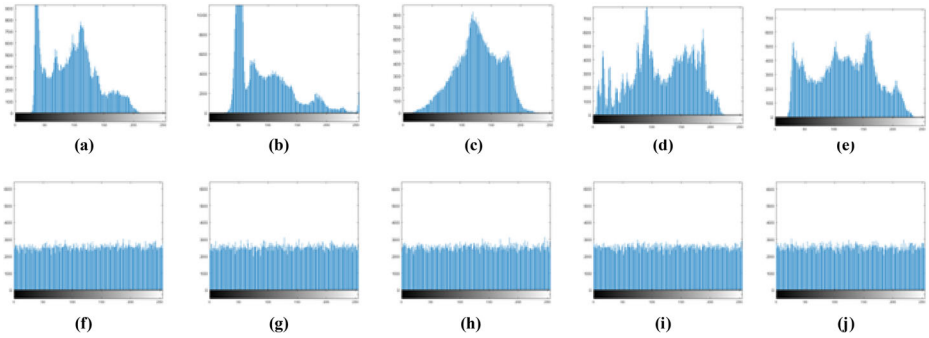


**Fig. 9** Histogram Analysis: (a), (b), (c), (d), (e) are the 5 input image histograms and (f), (g), (h), (i), (j) are 5 respective encrypted image histograms

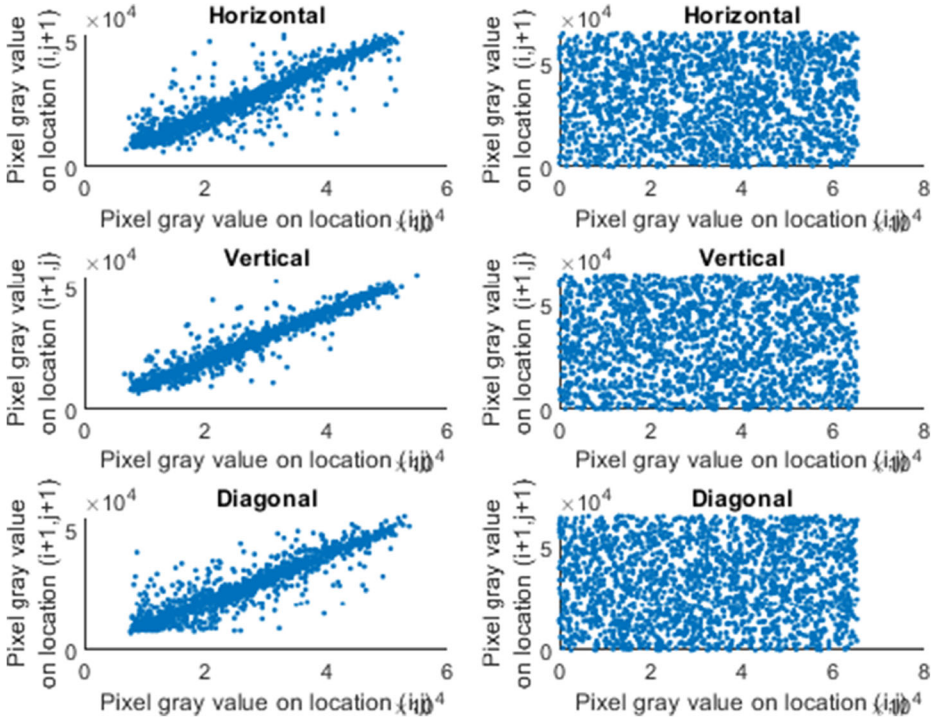


**Fig. 10** Correlation Analysis (a) Correlation of input image (b) Correlation of encrypted image

### 4.2.3 Entropy analysis

Entropy is defined as the degree of uncertainty in the image information. It should be low for the encrypted image as information uncertainty should be high and for original image it should be high as uncertainty of information is very low. Table 4 shows the entropy of 5 test images. The equation for entropy is as follows:

$$H(m) = \sum_{i=0}^{M-1} P(m_i) log_2 \frac{1}{P(m_i)} \tag{12}$$

### 4.2.4 Chi-square test analysis

Chi-square analysis gives the statistical representation of pixel uniformity throughout the gray level values. Similar analysis is done while plotting the histogram of encrypted images, which gives visual interpretation of pixel uniformity. The formula to calculate chi-square values of encrypted image is given below:

$$\chi^2 = \sum_{L=0}^{255} \frac{(observed - expected)^2}{expected} \tag{13}$$

**Table 3** Chi-square Analysis

| Images | Without DNA encoding | With DNA encoding |
|---|---|---|
| **Test image 1** | 233.4844 | 237.1172 |
| **Test image 2** | 232.1484 | 257.2188 |
| **Test image 3** | 240.5313 | 279.9609 |
| **Test image 4** | 237.4375 | 264.7734 |
| **Test image 5** | 239.8438 | 242.6250 |

Where L is the total gray level values. The expected value of chi-square is 256 for a 256 × 256 image. The chi-square values for both algorithms including DNA encoding and without DNA are given in Table 3. The analysis shows that the chi-square values obtained for the algorithm including DNA encoding shows better results that the algorithm which do not have DNA method in the encryption process. Therefore, DNA encoding distributes the pixels of image uniformly throughout the grayscale values.

### 4.3 Differential analysis

This analysis shows the sensitivity of the plain image towards any key change in the encryption algorithm. The various tests involves: NPCR and UACI.

#### 4.3.1 NPCR: Number of pixel change rate

This parameter tells the rate of change of pixels of the encrypted image when a single pixel of the input image is changed. It shows the percentage of change in the pixel values between the encrypted image and the cipher image when there is single bit change in input image. The equation for NPCR is as follows:

$$NPCR = \frac{\sum_{i,j} A_{(i,j)}}{MXN} \times 100\% \tag{14}$$

$$A(i,j) = \begin{cases} 1 \; if \; B_1(i,j) \neq B_2(i,j) \\ 0 \; if \; B_1(i,j) = B_2(i,j) \end{cases} \tag{15}$$

Where, $B_1$ and $B_2$ are the cipher image and input image respectively. M × N defines the size of the input image.

**Table 4** Simulation results for symmetric keys and 3D chaotic with XOR algorithm

| Images | NPCR | UACI | Entropy | Correlation | | |
|---|---|---|---|---|---|---|
| | | | | Horizontal | Vertical | Diagonal |
| **Test image 1** | 99.6002 | 29.4682 | 7.9899 | 0.0035 | 0.0048 | −0.0012 |
| **Test image 2** | 99.6185 | 30.5432 | 7.9892 | 0.0085 | 0.0013 | 0.0006 |
| **Test image 3** | 99.6033 | 27.3904 | 7.9893 | −0.0010 | −0.0005 | 0.0007 |
| **Test image 4** | 99.5865 | 29.6307 | 7.9890 | 0.0089 | −0.0037 | −0.0042 |
| **Test image 5** | 99.5956 | 29.4244 | 7.9894 | 0.0010 | 0.0055 | −0.0024 |

**Table 5** Simulation results for symmetric keys using 3D chaotic with XOR and DNA encoding

| Images | NPCR | UACI | Entropy | Correlation | | |
|---|---|---|---|---|---|---|
| | | | | Horizontal | Vertical | Diagonal |
| **Test image 1** | 99.6994 | 31.5592 | 7.9996 | −0.0083 | 0.0003 | −0.0002 |
| **Test image 2** | 99.6353 | 30.6439 | 7.9893 | 0.0048 | 0.0052 | −0.0006 |
| **Test image 3** | 99.6033 | 27.2931 | 7.9895 | −0.0048 | 0.0087 | −0.0011 |
| **Test image 4** | 99.5743 | 29.6893 | 7.9890 | −0.0022 | −0.0016 | 0.0046 |
| **Test image 5** | 99.6078 | 29.5717 | 7.9893 | 0.0004 | 0.0000 | 0.0004 |

### 4.3.2 UACI: Unified average changing intensity

UACI tells the average change in the intensity of the pixels between the encrypted image and original image. The equation for UACI is as follows:

$$UACI = \frac{1}{M \times N} \sum_{i,j} \frac{|B_1(i,j) - B_2(i,j)|}{255} \times 100\% \tag{16}$$

The results of the algorithm which includes symmetric keys and 3D chaotic with XOR operation are tabulated in Table 4 for all the 5 test images. Table 5 represents the simulation results for the algorithm including DNA encoding technique.

It is observed that from Tables 4 and 5, the proposed algorithm which includes symmetric keys using 3D chaotic with XOR and DNA encoding technique algorithm provides better results compared to the algorithm which do not include DNA encoding technique. For all the test images the proposed algorithm having DNA technique provides better NPCR, UACI, entropy, chi-square test and correlation analysis.

The proposed approach provides better results as compared to the other previous works as shown in Table 6. Whereas the CM and CF from [17] and MMC [1] provides better UACI compare to proposed and other methods but, the proposed method gives better results under NPCR and other metrics like entropy and for correlation analysis. From this it shows that the proposed algorithm; symmetric keys using 3D chaotic with XOR and DNA encoding shows less correlation coefficient value between pixel in the encrypted image and also gives better NPCR value.

**Table 6** comparison results of proposed algorithm with other methods

| Method | NPCR | UACI | Entropy | Correlation | | |
|---|---|---|---|---|---|---|
| | | | | Horizontal | Vertical | Diagonal |
| **CM and CF [26]** | 99.6000 | 33.5400 | – | −0.0003 | 0.0014 | – |
| **NPWLCM[25]** | 99.6292 | 28.5050 | 7.9975 | −0.0159 | −0.0195 | – |
| **MMC[24]** | 99.5100 | 33.4500 | 7.9997 | 0.0047 | 0.0030 | – |
| **3D chaos[13]** | 99.6048 | 33.5044 | 7.9890 | −0.0043 | 0.0014 | – |
| **2D chaos permutation/diffusion[12]** | 99.6057 | 30.4172 | 7.9797 | 0.0048 | 0.0065 | 0.0045 |
| **Proposed** | **99.6994** | **31.5592** | **7.9996** | **−0.0083** | **0.0003** | **−0.0002** |

## 5 Conclusion and future scope

In proposed work a novel symmetric keys using 3D chaotic with XOR and DNA encoding technique image encryption method is developed. Here, 3D chaotic map is used for various operations in encryption process. Three symmetric keys are developed to initialize the map. In this paper, pixel row & column permutation, diffusion and confusion of pixels is applied to make the algorithm strong enough to withstand any attacks. Two dimensions 'x' and 'y' of the 3D chaotic map are used simultaneously in the permutation of the row and column of the input matrix. The third dimension 'z' is used for generating the key image for further levels of encryption. The input and key image is DNA encoded using complementary rules. Now XOR logical operation is applied between encoded input image and key image. The resultant image is DNA decoded to decimal values to get final encrypted image. Detailed simulation analyses are done to test the proposed algorithm. Differential parameters like NPCR and UACI are being calculated, which tells the effect on encrypted image when pixel change is applied on the original image. Further statistical analysis such as histogram, chi-square test, entropy and correlation is calculated to check the uncertainty of information and relation of pixel in the encrypted image. The proposed algorithm which includes DNA encoding is compared with the results obtained from symmetric keys and 3D chaotic with XOR algorithm. The proposed work gives better results, further this algorithm is also compared with other state of art techniques. It was found that the proposed system gives better NPCR, chi-square test, entropy and correlation value. This algorithm can be applied on RGB as well as gray images in order to provide higher security in multimedia transmission.

The future scope of the proposed work can be applied to other multimedia tasks, such as feature selection [14], image segmentation [25] and image dehaze [26, 30]. Further, the proposed work can use different maps from the chaotic theory; later different asymmetric keys can also be used in the image encryption algorithm to make it more secure and unpredictable.

## References


1. Abdulredha HH, Nasir Q (2011) Low Complexity High Security Image Encryption Based on Nested PWLCM Chaotic Map. IEEE Internatonal conference for Internet Technology and Secure Transactions, ISBN 978–1–4577-0884-8, pp 220–225
2. Akter MT, Chowdhury MAM (2018) Observation of Different Behaviors of Logistic Map for Different Control Parameters. International Journal of Applied Mathematics and Theoretical Physics. Vol. 4, No. 3, pp. 84–90
3. Al-Maadeed S, Al-Ali A, Abdalla T (2012) A new chaos-based image encryption and compression algorithm. J Electr Comput Eng 2012:15
4. Barakat M, Eder C, Hanke T (2018) An Introduction to Cryptography, Timo Hanke at RWTH Aachen University, pp. 1–145
5. Belazi A, Talha M, Kharbech S (2019) Novel medical image encryption scheme based on Chaos and DNA encoding, 2018 IEEE. Translations, 18576092, **Page(s):** 36667–36681, **ISSN:** 2169-3536.
6. Brindha M (2017) Multiple stage image encryption using chaotic logistic map. 2017 international conference on intelligent sustainable systems (ICISS), ISBN: 978-1-5386-1959-9
7. Chai XL, Gan ZH, Lu Y, Chen YR, Han DJ (2017) A novel image encryption algorithm based on the chaotic system and DNA computing. Int J Mod Phys C 28:1750069
8. Essaid M ; Akharraz I ; Saaidi A ; Mouhib A (2019) A novel image encryption scheme based on permutation/diffusion process using an improved 2D chaotic system", 2019 international conference on wireless technologies, embedded and intelligent systems (WITS), IEEE

9. Kamel Faraoun, “Chaos-based key stream generator based on multiple maps combinations and its application to images encryption”. The International Arab Journal of Information Technology, Vol. 7, No. 3, pp231–240, July 2011.
10. Fridrich J (1997) Image encryption based on chaotic maps, in IEEE Int. Conf. Systems, Man, and Cybernetics, 2, 1105{1110
11. Hossain MB, Rahman MT, Rahman A, Islam S (2014) A new approach of image encryption using 3d chaotic map to enhance security of multimedia component. In: 2014 international conference on informatics, Electronics & Vision (ICIEV). IEEE, pp. 1–6
12. D. Huo, D.-f. Zhou, S. Yuan, S. Yi, L. Zhang, and X. Zhou, “Image encryption using exclusive-OR with DNA complementary rules and double random phase encoding,” Physics Letters A, vol. 383, no. 9, pp. 915–922, Feb 28, 2019.
13. Pawan N. Khade and Prof. Manish Narnaware, “3D Chaotic Functions for Image Encryption", International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, PP. 323–328, May 2012.
14. Li H, He F, Liang Y, Quan Q (2019) A dividing-based many-objective evolutionary algorithm for large-scale feature selection. Soft Comput 1–20
15. Liu H, Wang X (2010) Color image encryption based on onetime keys and robust chaotic maps. Computers & Mathematics with Applications 59(10):3320–3327
16. Maddodi G, Awad A, Awad D, Awad M, Lee B (2018) A new image encryption algorithm based on heterogeneous chaotic neural network generator and DNA encoding. Multimed Tools Appl 77(19):24701–24725
17. Masmoudi A, Bouhlel MS, Puech W (2010) A New Image Cryptosystem Based On Chaotic Map And Continued Fractions 18th European Signal Processing Conference (EUSIPCO-2010),Aalborg, Denmark, pp 1504–1508
18. Ratinder Kaur VK (2012) Banga “Image Security using Encryption based Algorithm”: International Conference on Trends in Electrical, Electronics and Power Engineering (ICTEEP'2012) July 15–16, Singapore
19. Saffari RM, Mirzakuchaki S (2016) A Novel Image Encryption Algorithm Based on Discrete Wavelet Transform Using Two dimensional Logistic Map ,” 24th Iranian Conference on Electrical Engineering (ICEE), IEEE
20. Safi HW, Maghari AY (2017, October) Image encryption using double chaotic logistic map. In promising electronic technologies (ICPET), 2017 international conference on (pp. 66-70). IEEE
21. Abhinav Srivastava, "A survey report on Different Techniques of Image Encryption", International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 6, June 2012, pages: 163–167.
22. Tang Z, Yang Y, Xu S, Yu C, Zhang X (2019) Image encryption with double spiral scans and chaotic maps. Security and Communication Networks 2019:1–15
23. Wang XY, Shi QJ (2005) New Type Crisis, Hysteresisand Fractal in Coupled Logistic Map, Chinese Journal of Applied Mechanics, pp. 501–506
24. Wu Y, Yang G, Jin H, Noonan JP (2012) Image encryption using the two dimensional logistic chaotic map. J Electron Imaging 21(1):013014–013011
25. Yu H, He F, Pan Y (2020) A scalable region-based level set method using adaptive bilateral filter for noisy image segmentation. Multimed Tools Appl 79(9):5743–5765
26. Zhang S, He F (2019) DRCDN: learning deep residual convolutional dehazing networks. Vis Comput 1–12
27. Zhang Q, Wei X (2013) A novel couple images encryption algorithm based on DNA subsequence operation and chaotic system. Optik 124(23):6276–6281
28. Zhang J, Fang D, Ren H (2014) Image encryption algorithm based on DNA encoding and chaotic maps. Math Probl Eng 2014:917147
29. Zhang Z, Wang H, Gao Y (2015) 2015/10/01, Chebyshev chaotic map-based authentication protocol for RFID applications, personal and ubiquitous computing, SN - 1617-4917.
30. Zhang J, He F, Chen Y (2020) A new haze removal approach for sky/river alike scenes based on external and internal clues. Multimed Tools Appl 79(3):2085–2107