Journal of Information Security and Applications 50 (2020) 102428

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Colour image encryption algorithm combining Arnold map, DNA sequence operation, and a Mandelbrot set

K.C. Jithin[1], Syam Sankar[1,*]

Department of Computer Science and Engineering, NSS College of Engineering, Palakkad, Kerala, India

ARTICLE INFO

Article history:
Available online 7 December 2019

Keywords:
DNA encryption
Chaotic encryption
Image encryption
Mandelbrot set

ABSTRACT

The third party misuse and manipulation of digital images is a threat to security and privacy of human subjects. The requirements to fulfil the security needs of digital images have led to the development of good encryption techniques. The proposed method of encryption scheme combines the techniques of chaotic image encryption and DNA (Deoxyribonucleic Acid) sequence operations. The encryption mechanism is applied separately on three channels (R, G and B) of the colour image with a suitable chaotic map. The map selection algorithm selects a good chaotic map based on image properties and other parameters. A method of DNA encoding is also applied after chaotic encryption. A new Mandelbrot Set based conditional shift algorithm is introduced to apply confusion effectively on R, G and B channels. Simulation and security analysis confirm that the proposed image encryption scheme demonstrates extraordinary performance in terms of security.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

The fast development of digital technologies and communication networks causes to generate more and more digital data [1]. Security and privacy of the images captured from electronic devices are of paramount importance with the increase in velocity, volume and veracity of the acquired multi-modal data. One effective well-known method to keep integrity is to encrypt the images by turning it from a meaningful version to a meaningless one [2]. Images have large data capacities, and it is known that a high correlation between adjacent pixels. So the traditional cryptosystems, such as RSA (Rivest–Shamir–Adleman), DES (Data Encryption Standard), AES (Advanced Encryption Standard), are not suitable for image encryption [3]. Recently, different kinds of image encryption algorithms based on optical transform, chaotic systems, Fourier transform, cellular automata, wavelet transform, magic cube, etc. have been studied by groups of researchers from all over the world [4].

There exist two groups in image encryption techniques based on the method applied to design the encryption and decryption system. Chaotic cryptography is the application of the mathematical chaos theory (that is, the chaotic sequences produced by chaotic maps) in the generation of cryptographic algorithms. The chaos-based image encryption mainly contains two steps called the confusion and the diffusion stage [5]. The confusion step is generally a pixel permutation part in which the pixel positions are getting randomly exchanged without changing the actual pixel values. This permutation process makes the image untraceable for attackers. It is not very secure for an image to have only this one stage since it can be found out by the attackers if they try hard [6].

To improve security, there comes the next step of the encryption process called diffusion. It mainly focuses on substituting the value of each pixel in the entire image with other values or by performing some operations on those pixels. The diffusion is performed through some chaotic maps. It is carried out by modifying the actual values of pixels sequentially with the random values generated from chaotic systems. This two-step confusion-diffusion process iterates for a certain number of times to enhance the security. The property which makes the chaotic maps appropriate for image encryption is its randomness.

Since first being investigated by Robert Matthews in 1989, the use of chaos in cryptography has attracted much interest, however, long-standing concerns about its security and implementation speed continue to limit its implementation. Many chaos-based image encryption algorithms have been proposed by various researchers. Wang et al. [2] proposed an image encryption procedure based on CML (Coupled Map Lattice) and DNA encryption. The method has an extended hamming distance calculation to improve the ability to resist plaintext attacks. Recently, the CML based image encryption systems [7–10] get more attention among

[*] Corresponding author.
E-mail address: syam.sankar8@gmail.com (S. Sankar).
[1] Both authors contributed equally to this work.

https://doi.org/10.1016/j.jisa.2019.102428
2214-2126/© 2019 Elsevier Ltd. All rights reserved.

researchers. The DNA computing process has lots of good characteristics [11,12] such as massive parallelism, huge storage and ultra-low power consumption. Many researchers have combined the properties of chaos and DNA encoding techniques to enhance the security of images in all aspects [13,14].

The Mondal and Mandal [15] proposed a cryptographic scheme which comprises the pseudo-random number (PRN) generation and Deoxyribonucleic Acid (DNA) computation. However, many algorithms have proven to be insecure. Dynamic properties of some of the chaotic maps degrade rapidly when coordinating with DNA computation. So, it is very essential and necessary to select an appropriate chaotic system [13].

When dealing with cryptanalysis [16–19], every encryption system is supposed to overcome all forms of attacks as we know that security is of utmost importance. Initially, the cryptanalyst has the knowledge of algorithm design and working of the whole security system except the secret key being used. In the end, the vulnerability of the system is utilized effectively to find hidden information. Every encryption system should be checked against at least a chosen-plaintext attack as a minimum criterion for a system to be declared as a secure one.

In this work, a new efficient image encryption technique is presented. The encryption system is based on three ingredients: Arnold map, Mandelbrot set, and DNA computation. The main contribution of this work is to present a DNA based chaotic image encryption system which can resist all forms of classical types of attack, and also to improve all the evaluation parameters so that the images can be transmitted effectively having no chance of being disclosed/decoded by the attackers. Since the keyspace is large, the system is free from brute-force attacks. The system can encrypt images of any size.

The rest of the paper is organized as follows. Section 2 describes various related works in chaotic image encryption and their weaknesses. Preliminary works associated with the proposed scheme are carried out in Section 3. Section 4 shows the proposed encryption strategy. Experimental results and security analysis are made in Section 5 and 6, respectively. Section 7 describes the conclusion of the work.

2. Related works

Algorithms like IDEA (International Data Encryption Algorithm), Advanced Encryption Standard and Data Encryption Standard are not suitable for image encryption because of two major factors: high correlation among pixels in images and redundant pixel values[20]. So, many algorithms using chaotic maps and DNA encoding aiming to encrypt images securely are reported by various researchers all over the world. An overview of the recently proposed image encryption methods is given hereafter.

In the paper [21], a new map named 2D-HSM (Two-Dimensional Henon-Sine Map) has been designed to apply pixel permutation and the DNA encoding scheme is implemented to perform diffusion on pixel values. Chai X et al. [1] proposed a novel encryption scheme in which wave-based permutation and row-by-row diffusion operations are applied on a DNA matrix, which is obtained after encoding the plain image. This scheme possesses a good security feature and can resist various attacks like chosen-plaintext attacks and more. The encryption scheme proposed by Wang et al. [22] applied DNA sequence operations in encrypting images and their extended hamming distance method improved the ability to resist known and chosen-plaintext attacks. Though the new schemes in this area further improved all evaluation parameters, the keyspace is comparatively poor.

The encryption schemes explained in the papers [1,21,22] are designed only for grey images. So before applying encryption, there comes an extra burden of converting colour images and other forms of data to the same form of grey images. Both the papers [13,22] have used CML (Coupled Map Lattice) in their encryption process to perform confusion or diffusion operation and their evaluation results are comparatively same but the scheme used in [13] is for colour images. Recently, Enayatifar et al. [23] proposed an encryption scheme combining cellular automata (CA) and DNA sequence operation to encrypt multiple images. It has an advantage of improving execution time, but the scheme is only for grey images and no significant changes in the values of evaluation parameters have been identified when comparing with the existing methods. Many algorithms in image encryption [24–28] are weak to resist the traditional chosen-plaintext attacks or known-plaintext attacks [20]. Initial conditions applied to chaotic maps play an important role in deciding its chaotic behaviour. In the paper Liu H et al. [29], adopted MD5(Message-Digest) hash to produce initial conditions of the chaotic maps, and followed by DNA encoding. Through the paper Norouzi et al. [30], reported that along with DNA encoding, the chaotic sequences generated by Cellular Neural Network (CNN) are also used to modify the pixel grey level values and to break the correlations between adjacent pixels of an image. It is very much essential to have any encryption system not solely depends on the keys but also on the input plain image too. In the paper Wu X. et al. [31], proposed a method in which the key streams for image encryption are generated from a secret key and the plain image, making the system to behave differently for each input image. The method also resists known-plaintext and chosen-plaintext attacks. The entropy value is relatively low when compared with other works in the same field. The work proposed by Chai X. et al. [3] combined memristive hyper-chaotic system, cellular automata and DNA sequence operations to develop an encryption system on grey images and the system seems relatively complex in computation but it can resist known-plaintext and chosen-plaintext attacks. DNA additions, subtractions and XOR(exclusive OR) operation are followed in most of the works describing image encryption using DNA [2]. Hash functions like SHA-256 (Secure Hash Algorithm-256) are even used in various works [2,32] to update the initial conditions used for generating key streams. Major weaknesses identified are as follows:

- None of the works specified how the selection of chaotic maps is performed so that the chosen map can be used for further processing.
- Many of the works solely depend on key streams only.
- No significant improvement is identified in the values of Shannon entropy, even in recent works.
- Most of the researchers use a maximum of four to five test images for experimentation.
- Not all works mention the behaviour of system when the cipher image is affected with noise of various kinds.
- Not all works state the uniformity of ciphered images generated by the system (variance of histogram).
- Many of the works seem computationally complex and the running speed of the corresponding encryption algorithm is not discussed.

Inspired by the above discussions, our work proposes a novel colour image encryption scheme based on chaotic maps, DNA sequence operations and Mandelbrot set.

First, key streams are generated with the help of chosen Arnold map and are encoded with DNA. Hamming distance calculation is performed between key streams and R-G-B components and is again encoded with DNA. Channel wise DNA encoding is applied by following a mechanism. This mechanism comprises both diffusion and confusion steps. Diffusion is performed with XOR operation and confusion of pixel values is applied with the help of a new conditional shift algorithm. This algorithm takes input from Mandelbrot set and again a diffusion operation is performed finally to get the cipher image.

2.1. Two other security defects

2.1.1. Low sensitivity with respect to changes in plaintext

A cryptosystem should be sensitive with respect to the plaintext. After carefully analysing different image encryption algorithms, it is seen that the existing algorithms stand very far from the desired property. The property is termed as the avalanche effect. This property is important for secure image encryption algorithms. This avalanche effect is quantitatively measured by estimating how many ciphertext bits will be changed when only one plaintext bit is changed [33].

2.1.2. Low sensitivity with respect to changes in secret key

An image encryption algorithm has to have a strong sensitivity with respect to changes in the secret key. This property can resist the known plain-text attack. The key sensitivity means that the system completely depends on the actual key values and parameter values. A small change in the secret key should definitely produce an entirely different output when the decryption is applied [34].

3. Preliminary works

3.1. Chaotic maps

Prior to the presentation of the proposed algorithm, it is more important to discuss the properties of different chaotic maps and to choose the best among them for the algorithm. Chaotic maps are some mathematical function which shows some sort of chaotic features. One of the major challenges faced during this work is about the selection of the map to be used along with the encryption-decryption system. There are different varieties of chaotic maps, each of which has its own properties and vulnerabilities. Then it is decided to choose the best map from the mainly used ones by applying simple encryption on a standard image with each of the maps. Then we analyse every map according to the randomness of the values it generates, i.e. the chaotic range and also the values of evaluation parameters. The following chaotic maps are used for the analysis.

3.1.1. Henon map

It is a discrete time dynamical system which shows good chaotic behaviour.

Henon map chooses a point (x_n, y_n) and changes it to a new point as follows [21]:

$$x_{n+1} = 1 - ax^2 + y_n \tag{1}$$

$$y_{n+1} = 1 - bx_n \tag{2}$$

This map depends on two parameters: *a and b*, and it becomes chaotic only when the parameters receive the values 1.4 and 0.3 respectively.

3.1.2. Logistic map

Logistic map is mathematically expressed as [35]

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

Here the parameter used is γ and the values must be in the range of [0, 4]. It is proven experimentally that the randomness (chaotic behaviour) is very large when γ takes the value between 3.5 and 4.

3.1.3. Arnold map

Arnold cat map is a chaotic map which is mainly used for the confusion of pixels. The Arnold cat map is a transformation as per the following form [36]

$$\tau(x, y) \to (2x + y, x + y) \mod (1) \tag{4}$$

3.1.4. Duffing map

Duffing map is a discrete time dynamical system which shows chaotic behaviour. This map is also known as the Holmes map. Duffing equation which named after George Duffing is used to design damped oscillators. It is defined as follows [37]

$$x_{n+1} = y_n \tag{5}$$

$$y_{n+1} = -bx_n + ay_n - yn^3 \tag{6}$$

Here the map depends on two constants a and b. And the system produces chaotic behaviour on the values $a = 2.75$ and $b = 0.2$.

3.1.5. Tent map

Tent map [33] is a real-valued function f_μ defined by [38]

$$f_\mu = \mu * \min\{x,\ 1 - x\} \tag{7}$$

Here the parameter μ is a positive real constant whose value ranges in the interval [0, 2]. The name *Tent map* is due to its tent-like structure of the graph defined by f_μ.

3.2. Chaotic map selection

An appropriate map selection is one of the important phases in the encryption mechanism. The chaotic nature of the sequences produced by the map enhances the security and prevents the encrypted images from being disclosed or breached by the attackers. The selection of map affects the quality of the encryption and our requirement is to choose the best maps so that the original pattern of image information is concealed in a better way.

Our method has taken entropy as the evaluation criteria for selecting the best maps out of *'N'* maps (chaotic map pool), which we have chosen randomly by assessing its property of chaotic behaviour. The entropy value measured for an encrypted image indicates the random distribution of pixel values ranging from 0 to 255, uniformly covering the entire region of the image such that the image information is totally hidden. The theoretical value of entropy is eight. The reason for selecting entropy as the criteria is that, an effective encryption algorithm should make the information entropy tend to 8 [14] and as a result, the pattern of image information is concealed in a better way. The selection of proper map which ultimately results higher entropy is very desirable.

We apply a simple *Map Selection()* procedure [39] on a standard image (Lena-256×256) with each of the chosen maps as input and compare the entropy values of the corresponding encrypted images. In our experiment, chaotic map pool is created with five ($N = 5$) maps: Arnold map, Logistic map, Tent map, Henon map, and Duffing maps. Thus, we get five encrypted Lena images and their entropy values are compared. Table 1 shows the maps, the initial values and other parameters used in our experiment. The maps which give high entropy value (Fig. 1) for the encrypted image after applying the *Map Selection* procedure are selected for further processing. The *Map Selection* procedure is defined as an algorithm (Algorithm 1) below

The entropy value is obtained as per the formula,

$$H(m) = -\sum_{i=0}^{255} P(x_i).\ \log[P(x_i)] \tag{8}$$

Where x_i represents the i^th grey value and $P(x_i)$ is the emergence probability of x_i. From Fig. 1, it is clear that Arnold map

Algorithm 1 Map selection.

```
1. Read the standard Lena image (P) of any size;
2. Decompose the three components of P and denoted the component
   matrices as R, G and B;
3. Convert the three matrices into 1D vectors: $R_{1D}$,$G_{1D}$and$B_{1D}$;
   while (1) do
       1. Select a map from the chaotic map pool;
       2. Generate three set of chaotic sequences: ($X_1$, $X_2$ and $X_3$)
          from the map, corresponding to each channel;
       3. Generate the three key streams: $S_1$, $S_2$ and $S_3$;
             $S_i = Mod\ ((X_i * 10^{10}), 256)\quad (for\ i = 1, 2\ and\ 3)$      (9)
       4. Perform XOR operation between each of the three channels
          ($R_{1D}$, $G_{1D}$ and$B_{1D}$) and the key streams ($S_1$, $S_2$ and $S_3$);
       5. Calculate the entropy of the encrypted image obtained after XOR
          and its value is recorded;
   end
4. Choose a maps which gives highest entropy value so that it can be used
   in the encryption system for further processing;
```

Table 1
Chaotic maps and parameter values.

NO	Chaotic Map	Equations	Initial values	Parameters
1	Arnold map	$\tau(x, y) \to (2x + y, x + y)\ \ \mathrm{mod}\ \ (1)$	x_0=0.105795019, y_0=0.2685999	
2	Logistic Map	$x_{n+1} = \gamma x_n(1 - x_n)$	$x_0 = 0.81$	$\gamma = 3.99$
3	Henon map	$x_{n+1} = 1 - ax^2 + y_n\ y_{n+1} = 1 - bx_n$	x_0= 0.631, y_0=0.189	$a = 1.4,\ b = 0.3$
4	Duffing Map	$x_{n+1} = y_n\ y_{n+1} = -bx_n + ay_n - yn^3$	$x_0 = 0.3,\ y_0 = 0.1$	$a = 2.750,\ b = 0.2$
5	Tent map	$f_\mu = \mu^*\min\{x,\ 1 - x\}$	$x0 = 0.5,\ \mu = 1.99$	

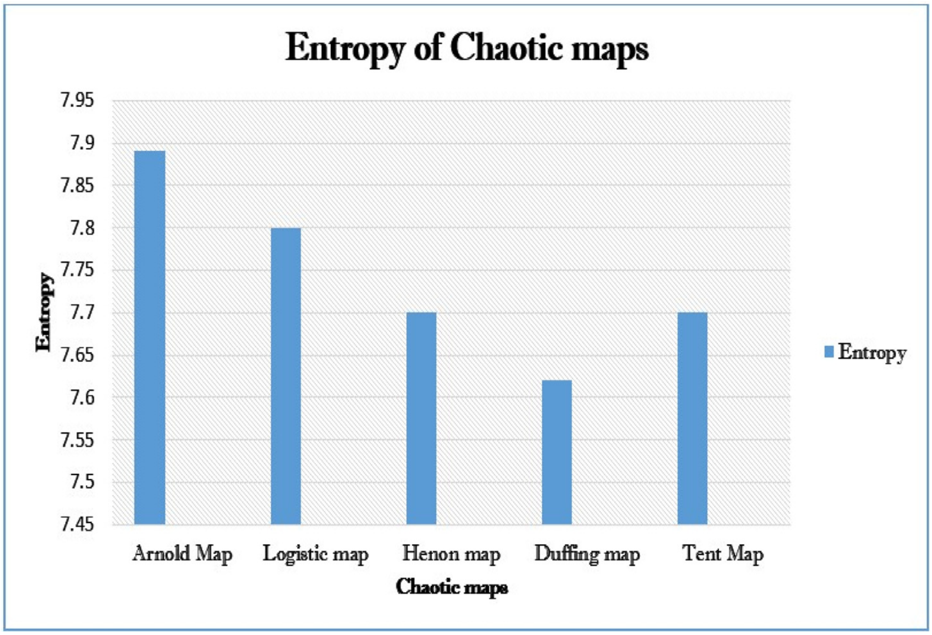


Fig. 1. Evaluation of entropy.

Table 2
DNA encoding rule.

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

Table 3
DNA XOR table.

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

produces high entropy value when processed with a standard Lena image. So our system uses Arnold map for the cryptographic operation

3.3. DNA encoding operation (Encode())

DNA encoding operation is the process used to map the sequence of binary values into DNA bases such as adenine *(A)*, thymine *(T)*, guanine *(G)*, and cytosine *(C)*, the building blocks of genetic code. The selection of *A, T, G* and *C* is accomplished with a DNA encoding rule [40]. The encoding is applied by taking two binary digits at a time. There are 24 types of DNA encoding rules are available to encode 00, 01, 10, and 11. Only 8 of them satisfy the Watson-Crick complementary rule [13], as shown in Table 2.

In our work, we use DNA encoding rule 01 to encode binary sequences. This rule can be used to encode images such that their pixel values (binary form) are replaced with the corresponding DNA sequences. Consider an 8-bit pixel value, 120 (01,111,000), can be expressed as" CTGA" in its encoded form.

4. Choose a maps which gives highest entropy value so that it can be used in the encryption system for further processing;

Let *Encode()* be the procedure to do this operation. For a DNA sequence," TGAC", the decoded binary form according to rule 01 is 11,100,001 (255 in decimal form). Let *Decode()* be the procedure to do this operation and is called in the subsequent algorithm.

3.3.1. XOR operation of DNA sequence

We have eight kinds of DNA XOR operation rules as we have 8 DNA rules satisfying Watson-Crick complementary rule [41]. DNA XOR operation as per DNA encoding rule 01 is shown in Table 3. For example, when we apply an XOR operation on the two DNA sequences," CTGA*" and" TGAC*", the result obtained is" GCGC".

3.4. Mandelbrot set

Mandelbrot set is a set of points in the complex plane. A point in the plane can be described using a complex number $c \in C$ written on the form $c = x+yi$ where $x,y \in R$. If we let the points belonging to the Mandelbrot set to be coloured in grey, we get the structure shown in Fig. 2. The values generated by the Mandelbrot set is used in the process of shifting in our encryption system [42].

$$\lim_{n\to\infty} Z_{(n+1)} = Z_n^2 + C, \quad Where\ Z_0 = 0 \tag{10}$$

To elude the set of all black pixels (zero values) in Fig. 2 we follow a procedure (Algorithm 2) as specified below to obtain a modified Mandelbrot output as shown in Fig. 3. Let W(i,j) corresponds to the pixel value in the location *(i, j)* in Fig. 2.

Where *C* is constant with a value ($=10^{14}$) in our experiment

Algorithm 2 Modified Mandelbrot set.

```
Initialization: Read the image (Fig. 2)
Output: Obtain the image (Fig. 3)
If $W(i,j)$ equals 0 then
    $W(i,j) = [( i*j) + C] \bmod 256;$
end
```

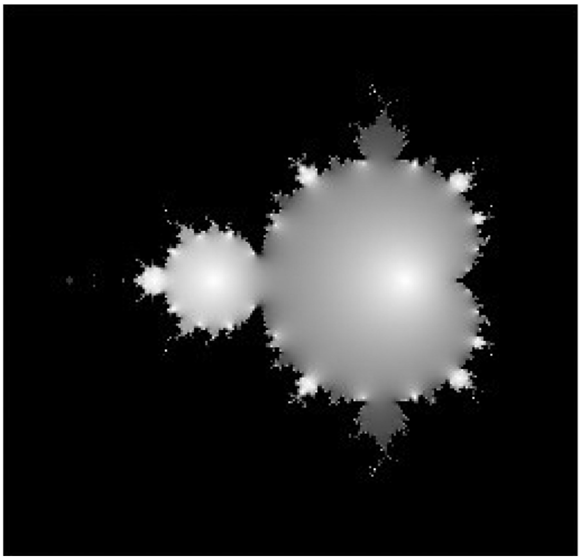


Fig. 2. Original Mandelbrot set image.

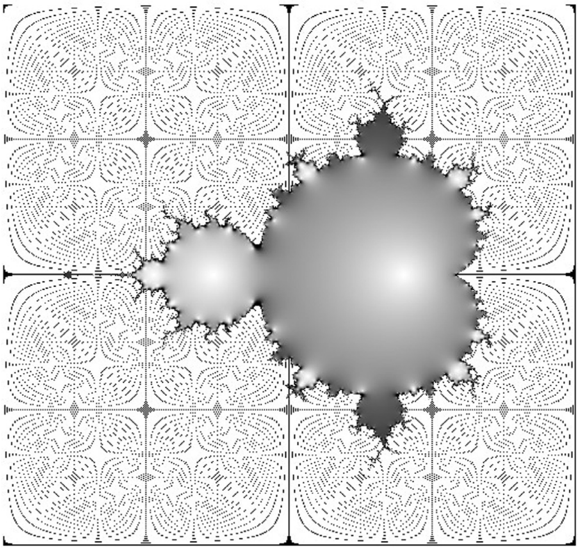


Fig. 3. Modified Mandelbrot set image.

4. Proposed system

The proposed encryption system combines three primary components such as chaotic map sequences, the set of points generated by Mandelbrot set and the DNA sequence operation in an ordered way such that the encryption algorithm produces a most encrypted form of the plain image unbreakable by the attackers while transmitting through a medium.

Fig. 4 shows the architecture diagram of the proposed encryption system. The encryption mechanism can be applied to

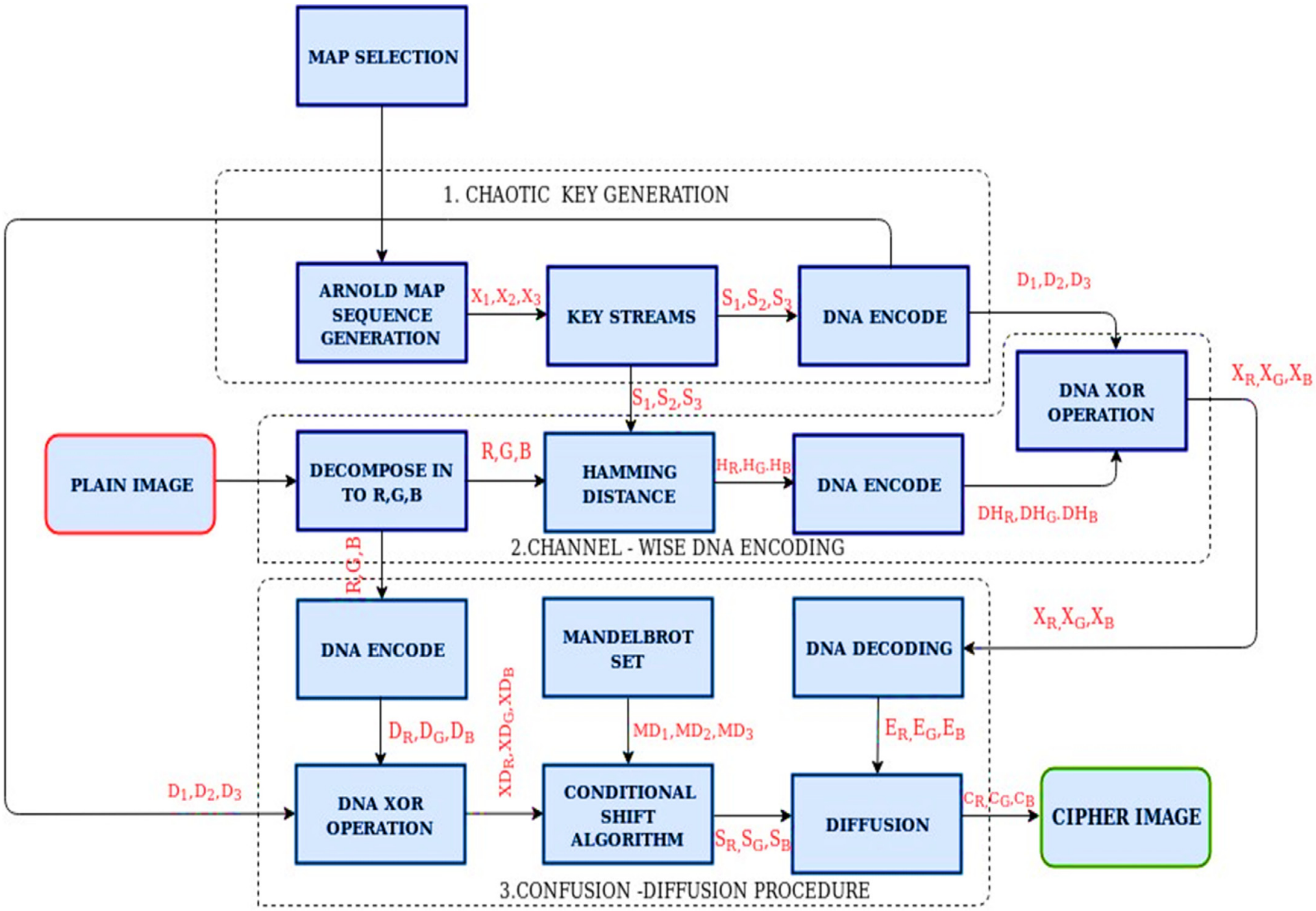


Fig. 4. Architecture diagram of Encryption system.

any images irrespective of its size and other parameters. The encryption system comprises three major subsystems:

1. *Chaotic key stream generation subsystem*
2. *Channel-wise DNA encoding subsystem*
3. *Confusion-diffusion procedure subsystem*

The whole encryption process can be done in seven steps under three subsystems as described below.

4.1. Chaotic key generation subsystem

The *Map Selection* procedure identifies Arnold map as the best map for encryption. So our system makes use of it. This subsystem comprises two steps (Step1 and Step2).

Step 1:

Obtain the key streams generated from the sequences (X_i) produced by the Arnold map as S_1, S_2 *and* S_3 *(Iterate the map t times and obtain* S_i *by Eq. (8))*

Step 2:

Apply DNA encoding operation on S_1, S_2 *and* S_3

$$D_i = Encode(S_i) \tag{11}$$

Each S_i is converted to its binary form and then DNA Encoding Operation, *Encode()*, is applied to generate a sequence of DNA bases (D_1, D_2 *and* D_3). If the size of image is 256 × 256, then size of S_i is also 256 × 256.

4.2. Channel-wise DNA encoding subsystem

This subsystem comprises the following steps: (Step 3 through Step 6).

Step 3:

The plain image is given as input to the subsystem and its R, G and B components are separated.

Step 4:

Hamming distance calculation is applied between component matrices and the key streams (S_i):

$$H_R(i,j) = Ham\_dist(R(i,j),\ S1(i,j)) \tag{12}$$

$$H_G(i,j) = Ham\_dist(R(i,j), S2(i,j)) \tag{13}$$

$$H_B(i,j) = Ham\_dist(R(i,j), S3(i,j)) \tag{14}$$

The function *Ham dist*(x, y) between two numbers x and y returns the number of bits which are different at same position in both x and y.

Step 5:

Apply DNA encoding operation on H_R, H_G *and* H_B to obtain the DNA sequence matrices: D_{HR}, D_{HG} *and* D_{GB}.

$$D_{HR} = Encode(H_R) \tag{15}$$

$$D_{HG} = Encode(H_G) \tag{16}$$

$$D_{HB} = Encode(H_B) \tag{17}$$

Step 6:

XOR operation is applied between the above DNA encoded matrices and the DNA encoded key streams obtained from chaotic key generation subsystem.

$$X_R = XOR(D_{HR}, D1) \tag{18}$$

$$X_G = XOR(D_{HG}, D2) \tag{19}$$

$$X_B = XOR(D_{HB}, D3) \tag{20}$$

4.3. Confusion-diffusion procedure subsystem

The whole process of this subsystem can be defined in a single step (Step 7)

Step 7: (Final Step)

Apply confusion-diffusion operation by invoking the procedure *Confusion-Diffusion-algorithm ()* (Algorithm 3). Finally, the cipher image components: $\mathrm{C_R}$, $\mathrm{C_G}$ and $\mathrm{C_B}$ are obtained to form the encrypted image for transmission.

Algorithm 3 Confusion-diffusion algorithm.

```
1. Read the DNA encoded R, G, and B channels of the plain image
        $D_R = Encode(R)$    (21)
        $D_G = Encode(G)$    (22)
        $D_B = Encode(B)$    (23)
2. Read the DNA encoded key streams as $D_1$, $D_2$, and $D_3$.
3. Perform the DNA XOR operation between DNA encoded RGB channels
   and the DNA encoded key streams
        $X_{DR}(i) = DNA\_xor(D_1(i), D_R(i));$    (24)
        $X_{DG}(i) = DNA\_xor(D_2(i), D_G(i));$    (25)
        $X_{DB}(i) = DNA\_xor(D_3(i), D_B(i));$    (26)
4. Apply conditional shift algorithm (Algorithm 4) on $X_{DR}$, $X_{DG}$ and, $X_{DB}$ to
   generate $S_R$, $S_G$ and $S_B$
5. Apply DNA decoding on the result produced by Channel-wise DNA
   encoding subsystem
        $E_R = Decode(X_R)$    (27)
        $E_G = Decode(X_G)$    (28)
        $E_B = Decode(X_B)$    (29)
6. Perform Bit XOR operation(Diffusion)
        $C_R(i) = (bitxor(S_R(i), E_R(i))) \bmod 256$    (30)
        $C_G(i) = (bitxor(S_G(i), E_G(i))) \bmod 256$    (31)
        $C_B(i) = (bitxor(S_B(i), E_B(i))) \bmod 256$    (32)
```

The confusion-diffusion procedure subsystem has an important role in image encryption mechanism. The actual encryption starts from here. The confusion procedure involves rearrangement or reordering of the pixel values without changing the actual value. The diffusion process intends to modify pixel values. Confusion and diffusion are the two major steps followed in every encryption mechanism. These steps are necessary to conceal the original image information from the attackers. These two processes can be done in any manner such that the original image has to be recovered at the decryption side. That is, each of them must be reversible too. In our encryption system, a conditional shift algorithm (Algorithm 4) is developed to meet the necessity of confusion and a Bit XOR operation takes the role of diffusion. In the end, the component channels are merged to form the final cipher image as indicated in the architecture diagram Fig. 4.

In the case of Arnold map, even if the state can be periodic after some iteration, the proposed system can resist this drawback in such a way that key streams are not applied directly with the images. Instead, the hamming-distance calculation between channels and key streams, followed by its DNA encoding can nullify this effect. So the hamming-distance calculation step is introduced purposefully because of the presence of periodic key streams generated by Arnold map after some iteration.

5. Experimental result

In order to fully demonstrate the advantages of our algorithm, we choose some standard colour images as shown in Table 4, each with size 256 × 256. All the experiments are implemented using a computer which has the following hardware environment: 2.16 GHz *CPU*, 4 GB memory and Windows 10 operating system. Matlab R2017a is used as the compiling software. The constants and initial values used for various maps are shown in Table 1.

5.1. Data set

Table 4 shows the standard colour images selected for the image encryption process. The quality of our encryption system is analysed by performing encryption on different images with different intensity values. Table 5 shows the simulation results of the proposed work.

6. Security analysis

6.1. Key space

A good encryption scheme should have a large key space to resist the brute force attack [34].

For an encryption system to be secure and strong, the keyspace should be large enough and usually, it should not be smaller than 2^{100} to make brute-force attacks infeasible. As per IEEE 754 floating-point standard (double), significant precision is defined to be 53 bits (including the hidden bit) and 15 decimal digits are required to represent the same.

- Initial values: X_0,Y_0 (Arnold map) to generate sequence for each channel, values allowed: [0, 1] and an iteration number t.
- The hamming distance matrix for each channel: H_R, H_G, H_B. (Let the size of each matrix of hamming distance be 256 by 256 for the input image of size 256 × 256)

The number of different values possible for each X_0, Y_0 is around $(2 \times 10^{15})^3$ and the same for the counter t is assumed to be 10^2. Each matrix has 65,536 elements. Each element location can have 256 (0–255) different values possible. The number of different values possible for three hamming distance matrix is around $256^{(65{,}536\times3)}$.

So the total keyspace is around $(2 \times 10^{15})^3 \times 10^2 \times (256)^{(65{,}536\times3)}$, which is definitely greater than 2^{100}, making brute-force attacks infeasible.

6.2. Key sensitivity

The encryption scheme must be sensitive to the initial and parameter values of the map taken [43]. The cryptosystem should produce different output for a slight change in the keys. Table 6 shows the analysis of key sensitivity. After applying encryption on the test images with the actual keys (keyset-1), as given in Table 1, we slightly change one of the initial values of Arnold map (the value $X_0 = 0.105795019$ is changed to $X_0 = 0.105795020$) and formed a keyset-1′, and then tried to decrypt the images with the changed key set (keyset-1′). The resultant decrypted images as shown in the fourth column of Table 6. It is clear that the decrypted images obtained with the changed keys (keyset-1′) are quite different, not the actual image produced though a small change is applied to keys. This shows that the proposed cryptosystem has perfect sensitivity to the keys and thereby preventing it from various attacks. In Table 6, Col. 2 and Col. 3 indicate the corresponding columns of the table. The third column shows the encrypted form of images with

Algorithm 4 Conditional shift algorithm.

```
1. Let M corresponds to the modified Mandelbrot set image matrix depicted in
   Fig. 3.
2. Read $X_{DR}$, $X_{DG}$ and $X_{DB}$

while i =1 to n do
    // n denotes the total number of columns in M
    Find the maximum value of $i^{th}$ column elements of M and denote it as $max_i$
    Find the maximum value of $i^{th}$ row elements of $X_{DR}$, $X_{DG}$ and $X_{DB}$ and
    denote them as $max_{ri}$, $max_{gi}$, and $max_{bi}$ respectively.
    Apply shifting operation as follows;

    case 1 do if ($max_i \le max_{bi}$) then
        $i^{th}$ row elements of $X_{DR}$ are confused by applying left cyclic shift
        $max_i$ times.
        else $i^{th}$ row elements of $X_{DR}$ are confused by applying right cyclic
            shift $max_i$ times.
        end if
    end

    case 2 do if ($max_i \le max_{ri}$) then
        $i^{th}$ row elements of $X_{DG}$ are confused by applying left cyclic shift
        $max_i$ times.
        else $i^{th}$ row elements of $X_{DG}$ are confused by applying right cyclic
            shift $max_i$ times.
        end if
end
    case 3 do
        if ($max_i \le max_{gi}$) then
            $i^{th}$ row elements of $X_{DB}$ are confused by applying left cyclic shift
            $max_i$ times.
        else
            $i^{th}$ row elements of $X_{DB}$ are confused by applying right cyclic shift
            $max_i$ times.
        end
    End

3. Let $S_R$, $S_G$ and $S_B$ denote the final shifted matrices corresponding to $X_{DR}$, $X_{DG}$
   and $X_{DB}$ respectively.
```

keyset-1′. The last column proves that, even if a small change is made in the keys (keyset-1 is changed to keyset-1′), we obtain two different cipher images. That difference in values is represented as images and it clearly shows how sensitive our system is.

It is clear that the decrypted images obtained with the wrong key (changed key) are quite different, not the actual image produced though a small change is applied to keys. This shows that the proposed cryptosystem has perfect sensitivity to the keys.

Table 4
Dataset.

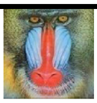
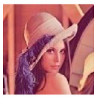
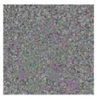
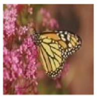
NO	Image Name	Standard colour images(256×256)
1	Baboon	
2	Barbara	
3	Cornfield	
4	Flower	
5	Lake	
6	Lena	
7	Monarch	
8	Peppers	
9	Soccer	
10	Yacht	

Table 5
Simulation result.

Input images	Cipher images	Decrypted images
Baboon		
Barbara		
Cornfield		
Flower		
Lake		
Lena		
Monarch		
Peppers		
Soccer		
Yacht		

6.3. *Histogram analysis*

An image histogram shows the distribution of the pixel intensity values, and it provides some statistical information of the image. A secure image encryption system can make the encrypted image have a uniform histogram to resist any statistical attacks. Table 7 shows the histogram of plain and cipher images. The plain image distribution differs significantly from the distribution of the encrypted image. An image histogram shows the distribution of the pixel intensity values, and it provides some statistical information of the image. A secure image encryption system can make the encrypted image to have a uniform histogram to resist any kind of statistical attacks [44]. Table 7 shows the histogram of plain and cipher images. The cipher image distribution differs significantly from the distribution of the plain image. So our system has applied a uniform pixel distribution on the encrypted image, concealing the actual image pattern. It is clear from the figures that there are no sequences/patterns of any kind visible in the corresponding encrypted images.

6.3.1. *Variance of histogram*

We use a measure called variance of the histogram to test the uniformity of ciphered images. We generate two different ciphered images encrypted with two different keys, and its variance of histograms is calculated. In order for the two ciphered images to be uniform, the values of the variances must be close enough. The closeness in the values shows strong uniformity while keys are varying. The variance of the histogram is as follows:

$$VAR(Z) = \left(\frac{1}{N^2}\right)\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{1}{2}\right)\times\left(z_i - z_j\right)^2 \tag{33}$$

where Z is the vector of the histogram values and $Z = \{z_1,z_2,\ldots,z_{256}\}$, z_i *and* z_j denote the number of pixels where grey values are equal to i and j, respectively.

Table 6
Key sensitivity analysis.

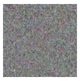
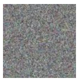
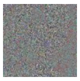
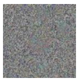
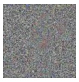
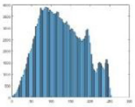
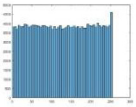
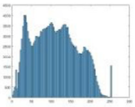
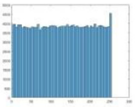
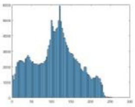
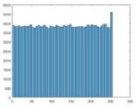
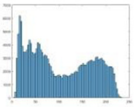
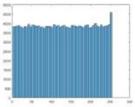
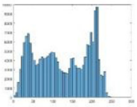
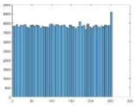
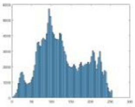
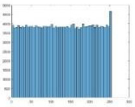
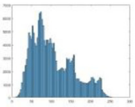
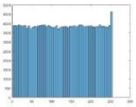
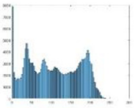
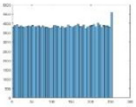
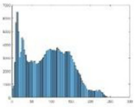
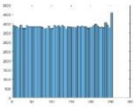
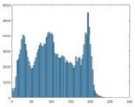
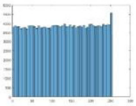
Plain Image	Encrypted using original keyset-1	Encrypted using keyset -1′	Decrypted-Col.2 cipher images- using keyset-1′	Difference between encrypted images in Col.2 and Col.3
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Table 7
Histogram analysis.

Plain Image -No	Histogram of plain image	Histogram of cipher image
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

We have taken the Lena image (256 × 256) for experimentation. Let $K = x, y, t,..$ be the key-set used to encrypt the plain Lena image. One of the keys in the keyset K is changed to form another key-set and encrypt the plain image again to produce the second ciphered image. Now we can analyse the change in the values of variance. If there is no big difference between them, the ciphered images are concluded as uniform even if they are encrypted with different keys. As the Lena is a colour image, the variance of the histogram of each of the channels (R, G and B) is calculated. In Table 8, the first column gives the values of variance measured with the plain image. The second column shows the values of variance obtained for the image encrypted with the key-set K. The third column K_x indicates that only the key x in key-set K is slightly varied and after encryption with the key-set K_x, the variance is calculated. In the second column, the keyset K_y indicates that only y of the key-set K is varied and obtained the variance after encryption with it. Similarly, one of the keys in K is changed to form the new key-set in the rest of the columns, and the corresponding variances are indicated. Table 8 shows that the variance of the ciphered images is very low when compared with the variance of the plain image. It is clearly evident that the differ-

Table 8
Variance analysis.

Image	Channels	Plain Image	K	K_x	K_y	K_t
Baboon	R	21,215.875	260.8515	278.9062	230.7343	234.75
	G	33,584.2109	287.8515	262.8125	261.9921	216.7031
	B	18,980.0234	275.2890	247.75	270.4140	271.6328
Barbara	R	20,170.4062	240.9296	272.9375	230.5468	269.1484
	G	33,742.5937	274.3515	266.0625	254.1171	241.8515
	B	30,490.6562	259.7109	243	240.1640	270.7890
Lena	R	60,506.8515	271.1796	265.4375	259.7578	249.7265
	G	30,923.625	272.7031	293.1562	262.0234	257.4453
	B	82,885.6796	241.6875	275.3437	262.0234	256.1875

ence in variance between K and $K_{x,}$ K and K_y, K and $K_{t,}$ is not high. So the ciphered images are uniform in nature. It is also concluded that, in our proposed encryption system, the difference in variance value shows that the histogram depends on the plain image too.

6.4. Correlation analysis

In every image, some level of correlation is maintained between every pair adjacent pixels. Good encryption schemes are expected to avoid or hide such correlations among pixels to protect the data from different attacks [45]. To find out the correlations among pixel pairs, it is needed to select certain adjacent pixels of the input image along with the three directions, that is, in horizontal (H), vertical (V), and diagonal (D).

The correlation among the pixel pairs can be calculated as follows:

$$r_{xy} = \frac{N^2.\mathrm{cov}(x,y)}{\sum_{i=1}^{N}(x_i - E_x)^2.\sum_{i=1}^{N}(y_i - E_y)^2} \tag{34}$$

$$E_x = \frac{\sum_{i=1}^{N} x_i}{N}$$

$$\mathrm{cov}(x,y) = E((x - E_x)(y - E_y))$$

where (x, y) is the two of horizontal, vertical or diagonal adjacent pixels sequences, and N denotes the size of the image.

Table 9 shows the correlation distribution of each pair of pixels (in each R, G and B) for the image Baboon in three directions (Horizontal -H, Vertical -V) and Diagonal- D) and also the distribution of corresponding cipher image. Similarly, Tables 10 and 11 show the correlation distribution of the test images, Lena and Pepper, respectively. The values of correlation coefficients of all the cipher images are shown in Table 12. It is clearly evident from Table 12 that the correlation coefficients between each pair pixels of all the cipher images are very low in all the three directions (H, V and D). So, it is concluded that the cipher images conceal all forms of pattern in it making unbreakable by the attackers.

6.5. Shannon entropy

Image entropy is a term which is used to define the crowdedness of an image, i.e. the amount of data or information which is hidden in an image using any kind of algorithms. Shannon Entropy describes a measure of randomness of an image [46]. Shannon entropy for an 8-bit image is obtained as given below:

$$H(m) = -\sum_{i=0}^{255} P(x_i) \times \log P(x_i) \tag{35}$$

where x_i is the *ith*grey value and $P(x_i)$ is the probability of x_i in an image. A good encryption scheme should have an entropy value close enough to 8. Our encryption method gives the highest entropy value for various images. Table 13 shows the entropy values of different images.

6.6. SSIM

The structural similarity (SSIM) index is a method for measuring the similarity between two images (here, the actual plain image and the decrypted image at the receiver side are tested for similarity). Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close [36]. These dependencies carry important information about the structure of the objects in the visual scene. The resultant SSIM index is a decimal value between −1 and 1. Table 14 shows the SSIM value of images. For a better encryption method, the SSIM value must be close enough to 1.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{36}$$

- μ_x is the average of x
- μ_y is the average of y
- ${\sigma_x}^2$ is the variance of x
- ${\sigma_y}^2$ is the variance of y
- σ_{xy} the covariance of x and y
- $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$, two variables to stabilize the division with weak denominator
- L the dynamic range of the pixel-values
- $k_1 = 0.01$ and $k_2 = 0.03$

6.7. Robustness against differential attack

6.7.1. NPCR and UACI

Sometimes, an opponent may try to make a minute change in the original image which is used for encryption and observe the change in encryption results (that is, cipher image of the original and the cipher image of original with a minute change). In this way, the opponent tracks the relationship between the original image and the two ciphered images [47]. The process which helps in decrypting an image is called differential cryptanalysis. So, it is obvious that our system must be anti-differential, which means that it should be difficult for the attackers to identify how the plain image is related to cipher image. The two main parameters used for this are the Number of Changing Pixel Rate (NPCR) and the Unified Averaged Changed Intensity (UACI). These parameters are defined as follows:

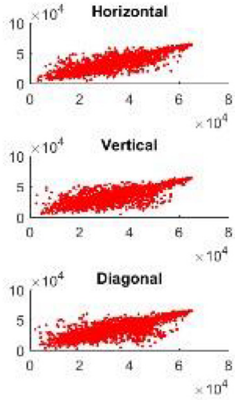
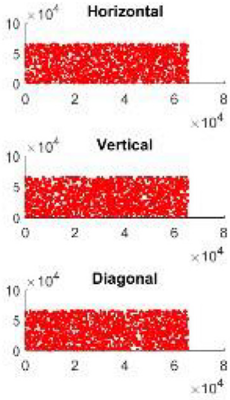
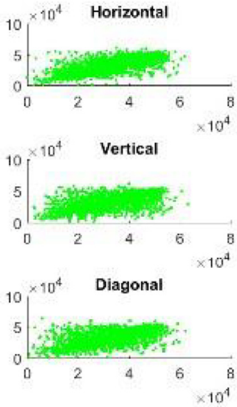
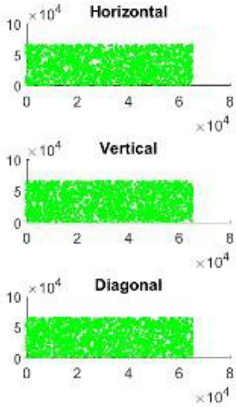
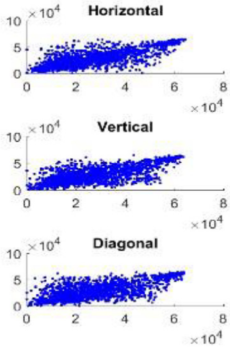
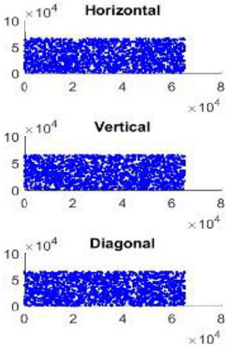
$$NPCR = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j)}{m \times n} \times 100 \tag{37}$$

$$UACI = \frac{1}{255 \times m \times n}\left[\sum_{i=1}^{m}\sum_{j=1}^{n} C(i,j) - C1(i,j)\right] \times 100 \tag{38}$$

$D_{i,j} = =0,\ if\ C(i,j) = C1(i,j), D_{i,j} = =1,\ if\ C(i,j) \neq C1(i,j)$, where *C* and *C1* are the two ciphered images corresponding to the plain image before and after slightly change respectively. The values *m* and *n* indicate the width and height of the image. The proven value of UACI score is nearly 0.33. When the NPCR score is closer to 1, it means that the encryption method is more sensitive to the input image. Thus, these schemes will resist differential attack to a large extent. Table 15 shows the NPCR and UACI values of various images in the data set. All the values are very close to the theoretical values.

In Table 15, for the image, Baboon, choose a random pixel location (128, 28, 2). The value 2 in the triplet indicates the green (G) component of the image stating that the location (128, 28) is taken from the G component. Here 95 is the original pixel value in that location. We make a slight change in the pixel value and

Table 9
Correlation analysis of Baboon.

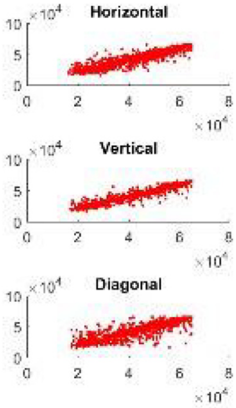
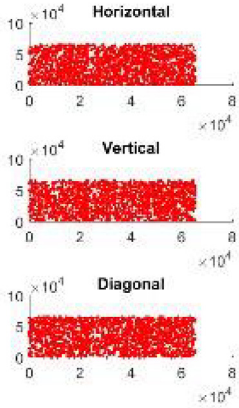
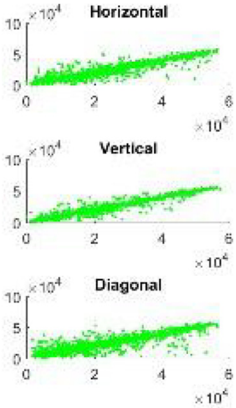
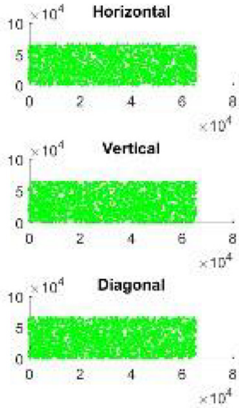
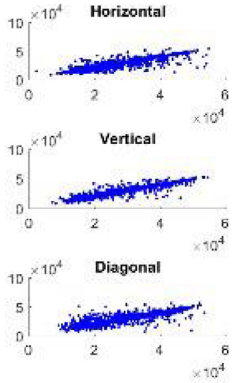
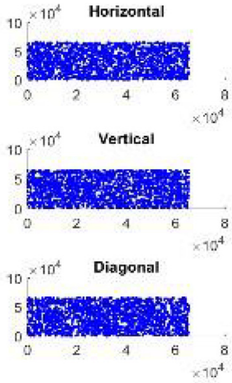
Channels	Plain image	Cipher image
RED		
GREEN		
BLUE		

obtain 94 as the new value in that location. Then the modified image (change in one-pixel value only) is encrypted with our system and calculates NPCR and UACI values as 0.9957 and 0.3347 respectively and the values show its closeness to the theoretical values.

6.8. Noise attack analysis

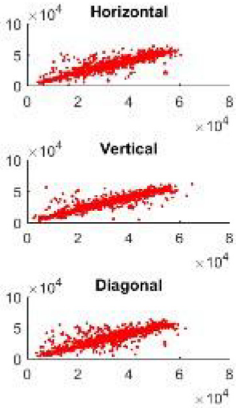
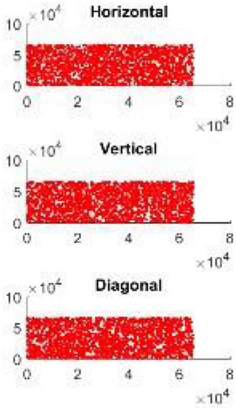
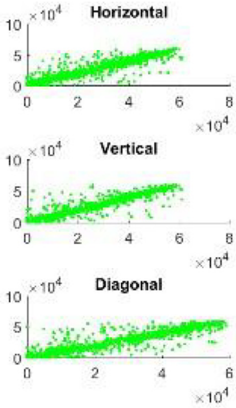
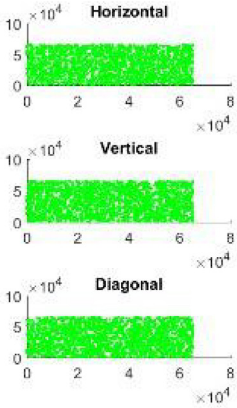
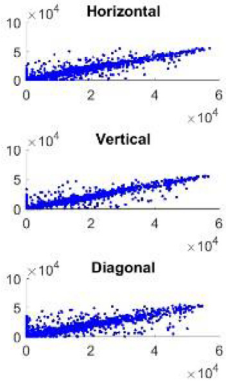
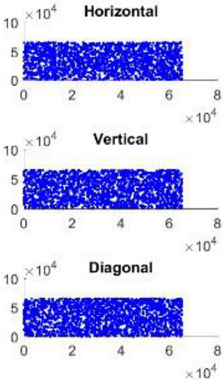
In this section, we analyse how our encryption-decryption system behaves with noises. The transmission channel always contains some form of noise. During transmission, the image in

Table 10
Correlation analysis of Lena.

Channels	Plain image	Cipher image
RED		
GREEN		
BLUE		

the encrypted form will definitely be badly affected by some noises. So our decryption algorithm should be capable to resist the noise in such a way that the decrypted images should be understandable or human-readable form even if it is contaminated with noise during transmission. So we have to prove that the decryption system is efficient enough to generate the recognizable image from the encrypted image containing noise. Consider the following noises for the analysis:

Table 11
Correlation analysis of Pepper.

Channels	Plain image	Cipher image
RED		
GREEN		
BLUE		

6.8.1. Gaussian noise

Principal sources of Gaussian noise in digital images arise during acquisition. The sensor has an inherent noise because of the level of illumination and its own temperature, and also, the electronic circuits connected to the sensor inject their own share of electronic circuit noise [48]. A typical model of image noise is gaussian, additive, independent at each pixel, and independent of the signal intensity.

In Table 16, the first, second and third columns show different cipher images affected with default Gaussian noise, low Gaussian

Table 12
Values of correlation coefficients.

NO		H	V	D
1	R	0.0005	0.0059	0.0014
	G	0.0078	0.0042	−0.001
	B	0.0021	−0.0039	−0.0114
2	R	−0.0039	−0.0047	−0.0058
	G	−0.0067	0.0037	−0.0048
	B	−0.0035	0.0007	−0.0068
3	R	0.0029	0.0047	−0.0026
	G	0.0005	−0.0005	0.0023
	B	−0.0029	0.0007	−0.0043
4	R	0.0001	0.0019	−0.0028
	G	−0.0015	−0.0078	−0.0042
	B	−0.0071	−0.0006	−0.0037
5	R	−0.005	−0.0054	0.0006
	G	−0.0003	0.0003	−0.0077
	B	0.0025	0.0052	−0.005
6	R	0.0021	0.0018	−0.0026
	G	−0.0006	0.0004	0
	B	−0.005	0.001	−0.0104
7	R	−0.0047	0.0022	−0.0061
	G	0.0004	0.0009	−0.0071
	B	−0.002	0.0004	−0.007
8	R	−0.004	−0.0042	−0.0009
	G	−0.0007	0.0012	−0.0069
	B	−0.0019	0.0021	−0.0041
9	R	−0.006	−0.0003	−0.0075
	G	−0.0073	−0.0058	−0.0012
	B	−0.0063	−0.0008	−0.0083
10	R	0.0064	−0.0044	−0.0024
	G	−0.0052	−0.0002	−0.008
	B	−0.0007	0.0035	−0.0033

Table 13
Shannon entropy.

Image name	Cipher image entropy (proposed)	Plain image entropy
Baboon	7.99911	7.7749
Barbara	7.99908	7.7176
Cornfield	7.9991	7.7431
Flower	7.9991	7.6828
Lake	7.99904	7.7451
Lena	7.99924	7.7532
Monarch	7.99917	7.5442
Peppers	7.99914	7.7124
Soccer	7.99901	7.6319
Yacht	7.99909	7.6549

Table 14
SSIM.

Image name	SSIM (structural similarity)
Baboon	0.9281
Barbara	0.8722
Cornfield	0.9273
Flower	0.8619
Lake	0.8346
Lena	0.9354
Monarch	0.892
Peppers	0.9206
Soccer	0.9034
Yacht	0.8857

noise and high Gaussian noise respectively. The last three columns indicate the decrypted form of the corresponding cipher images. Table 16 shows that the decrypted images are recognizable even if the corresponding encrypted images are affected with various forms of Gaussian noises.

Table 15
NPCR and UACI.

Image name	Pixel position	Pixel value	Changed value	NPCR	UACI
Baboon	(1,1,1)	127	126	0.9957	0.3347
	(1256,1)	153	152	0.99568	0.33468
	(1,1,2)	113	112	0.9957	0.3347
	(128,28,2)	95	94	0.9957	0.3347
	(28,128,3)	42	41	0.9957	0.3347
Barbara	(1,1,1)	255	254	0.99614	0.333857
	(1256,1)	255	254	0.9961	0.3338
	(1,1,2)	255	254	0.9961	0.3338
	(128,28,2)	172	171	0.99614	0.33385
	(28,128,3)	28	27	0.9961	0.3338
Cornfield	(1,1,1)	90	89	0.99615	0.33362
	(1256,1)	82	81	0.9962	0.3336
	(1,1,2)	109	108	0.9962	0.3336
	(128,28,2)	100	99	0.9962	0.3336
	(28,128,3)	195	194	0.9962	0.3336
Flower	(1,1,1)	37	36	0.9964	0.3337
	(1256,1)	61	60	0.9964	0.3337
	(1,1,2)	13	12	0.99642	0.33367
	(128,28,2)	167	166	0.99642	0.33367
	(28,128,3)	19	18	0.9964	0.3337
Lake	(1,1,1)	94	93	0.9962	0.335
	(1256,1)	131	130	0.9962	0.335
	(1,1,2)	38	37	0.9962	0.335
	(128,28,2)	24	23	0.9962	0.335
	(28,128,3)	205	204	0.9962	0.335
Lena	(1,1,1)	223	222	0.9957	0.3333
	(1256,1)	208	207	0.9957	0.3333
	(1,1,2)	135	134	0.9957	0.3333
	(128,28,2)	145	144	0.9957	0.3333
	(28,128,3)	165	164	0.9957	0.3333
Monarch	(1,1,1)	109	108	0.9962	0.3345
	(1256,1)	91	90	0.9962	0.3345
	(1,1,2)	98	97	0.9962	0.3345
	(128,28,2)	89	88	0.9962	0.3345
	(28,128,3)	141	140	0.9962	0.3345
Peppers	(1,1,1)	38	37	0.9961	0.334
	(1256,1)	42	41	0.9961	0.334
	(1,1,2)	42	41	0.9961	0.334
	(128,28,2)	83	82	0.9961	0.334
	(28,128,3)	35	34	0.9961	0.334
Soccer	(1,1,1)	134	133	0.996	0.3343
	(1256,1)	63	62	0.996	0.3343
	(1,1,2)	102	101	0.996	0.3343
	(128,28,2)	15	14	0.996	0.3343
	(28,128,3)	179	178	0.996	0.3343
Yacht	(1,1,1)	83	82	0.9962	0.3353
	(1256,1)	100	99	0.9962	0.3353
	(1,1,2)	103	102	0.9962	0.3353
	(128,28,2)	137	136	0.9962	0.3353
	(28,128,3)	12	11	0.9962	0.3353

6.8.2. *Poisson noise/shot noise*

The dominant noise in the darker parts of an image from an image sensor is typically that caused by statistical quantum fluctuations, that is, variation in the number of photons sensed at a given exposure level [49]. This noise is known as photon shot noise. Shot noise has a root-mean-square value proportional to the square root of the image intensity, and the noises at different pixels are independent of one another. Shot noise follows a Poisson distribution. In Table 17, the first, third and fifth columns show different cipher images affected with default Poisson noises. The second, fourth and sixth columns indicate the decrypted form of the corresponding cipher images. Table 17 clearly proves that the decrypted images are recognizable even if the encrypted images are affected with Poisson noise.

6.8.3. *Salt and Pepper noise*

Fat-tail distributed or “impulsive” noise is sometimes called salt-and-pepper noise or spike noise [50]. An image containing

Table 16
Gaussian noise analysis.

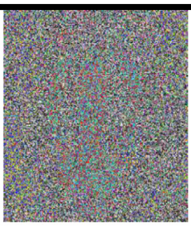
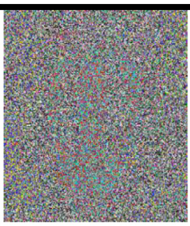
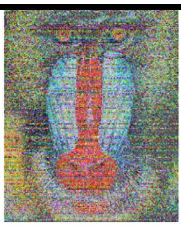
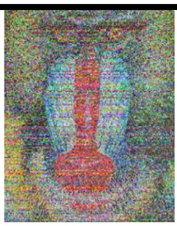
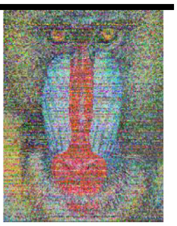
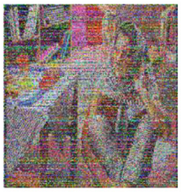
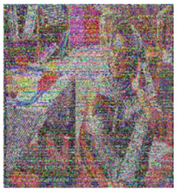
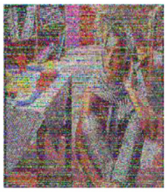
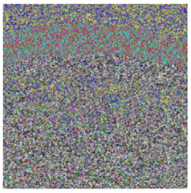
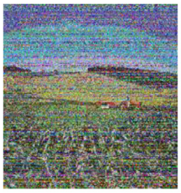
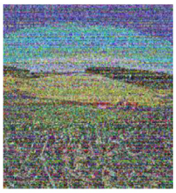
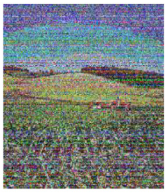
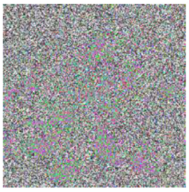
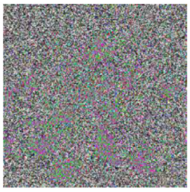
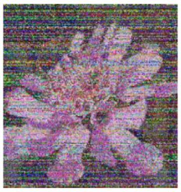
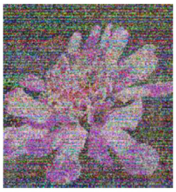
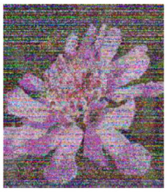
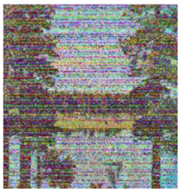
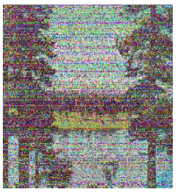
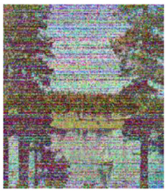
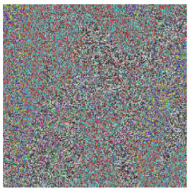
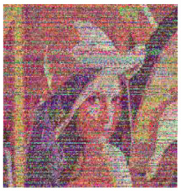
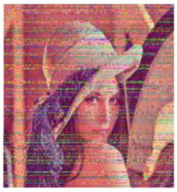
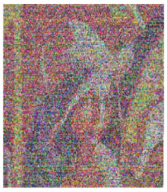
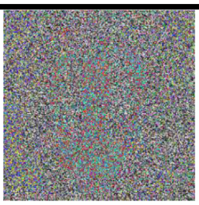
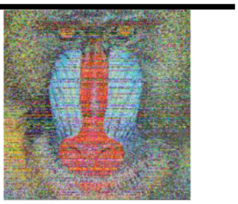
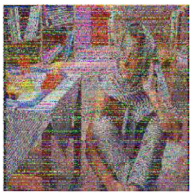
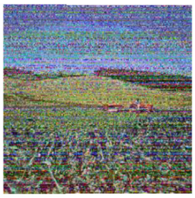
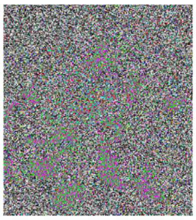
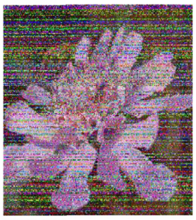
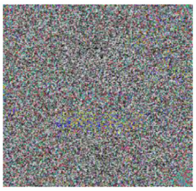
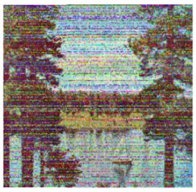
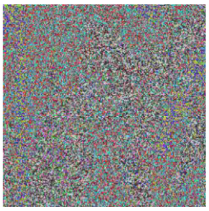
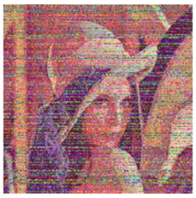
No.	Cipher noise- default	Cipher noise -low	Cipher noise- high	Decrypted- noise default	Decrypted- noise low	Decrypted- noise high
1						
2						
3						
4						
5						
6						

Table 17
Poisson noise analysis.

No	Cipher noise default	decrypted noise default
1		
2		
3		
4		
5		
6		

salt-and-pepper noise will have dark pixels in bright regions and bright pixels in dark regions. This type of noise can be caused by analog-to-digital converter errors, bit errors in transmission, etc. It can be mostly eliminated by using dark frame subtraction, median filtering, combined median and mean filtering and interpolating around dark/bright pixels. In Table 18, the first, second and third columns show different cipher images affected with low Salt and Pepper noise, default Salt and Pepper noise and high Salt and Pepper noise respectively. The last three columns indicate the decrypted form of the corresponding cipher images. Table 18 shows that the decrypted images are recognizable even if the encrypted images are affected by Salt and Pepper noise.

6.8.4. Speckle noise

Speckle is a granular 'noise' that inherently exists in and degrades the quality of the images [51]. Speckle noise results from these patterns of constructive and destructive interference shown as bright and dark dots in the image.

In Table 19, the first, second and third columns show different cipher images affected with low speckle noise, default speckle noise and high speckle noise respectively. The last three columns indicate the decrypted form of the corresponding cipher images. Table 19 shows that the decrypted images are recognizable even if the encrypted images are affected with Speckle noise.

6.8.5. PSNR

A higher value of PSNR(Peak signal-to-noise ratio) is good because of the superiority of the signal to that of the noise [52]. The larger the PSNR the better the image separation result. Table 20 shows the PSNR values of images. For a grey scale image, the PSNR is computed as,

$$PSNR = \log \frac{255 \times 255}{MSE} \tag{39}$$

MSE is the mean square error value and is defined as follows,

$$MSE = -\frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} [I(i,j) - C(i,j)]^2 \tag{40}$$

where *Im*×*n and Cm*×*n are the plain* image and its corresponding cipher image respectively

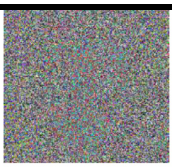
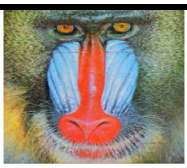
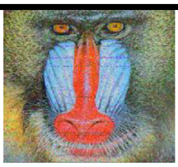
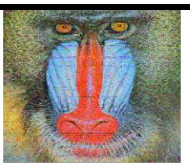
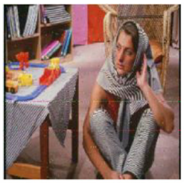
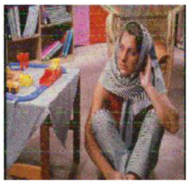
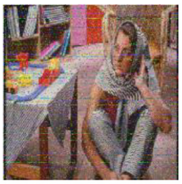
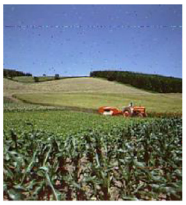
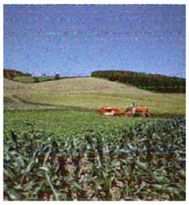
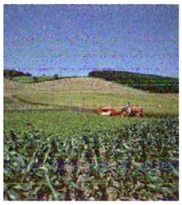
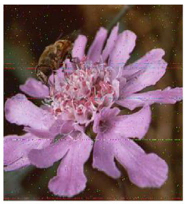
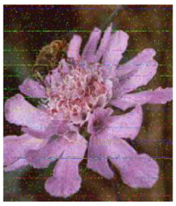
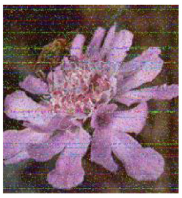
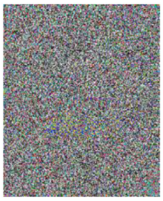
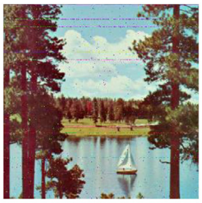
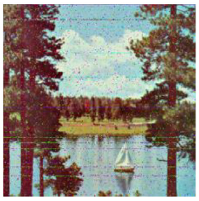
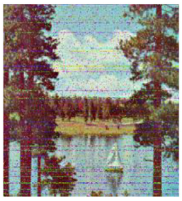
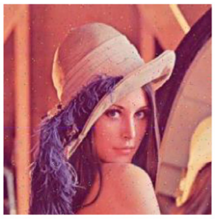
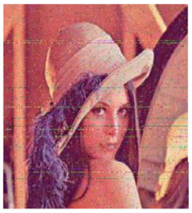
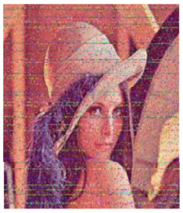
6.9. Classical types of attack

The encryption-decryption algorithm that every one designs is made publicly available at the end. Even hackers could be able to analyse the steps in the algorithm as it is available under the public domain, except the keys shared between the sender and receiver side to perform the encryption or decryption operation. There are four classical types of attacks: ciphertext only, known-plaintext, chosen-plaintext and chosen-ciphertext. It is clearly evident that chosen-plaintext attack is the most crucial or horrible attack since the hacker somehow has obtained temporary access to the environment of the encryption-decryption system and he/she could be able to generate the corresponding ciphertext for a chosen-plaintext. We could be able to resist the remaining three attacks if our cryptosystem can defeat the chosen plain text attacks.

Our proposed work is sensitive to the parameters (or constant values) and the initial values used with Arnold map and also sensitive to the constant values used with Mandelbrot set. Most importantly, there is a step in the encryption system called hamming distance calculation, which calculates hamming distance between the channels of the plane image and the corresponding key streams.

The hamming distance matrix plays a major role in the further processing of our encryption system. So our system not only depends on keys but also on the plain image too. The iteration count *t* (a key-value) can be set to different values randomly for each input image. So the Arnold cat map generates entirely different output when the number of times it iterates (iteration count) changes. So finally, we conclude that even if the opponent is able to obtain some samples of plain-cipher text pairs, our system is sufficient to resist chosen-plain text attack.

Table 18
Salt and Pepper noise analysis.

No	Cipher- noise Low	Cipher - noise default	Cipher -noise high	Decrypted- noise low	Decrypted -noise default	Decrypted-noise high
1						
2						
3						
4						
5						
6						

6.10. *Occlusion attack*

When digital images are transmitted through networks, some data may be lost because of congestion in the network or malicious destruction [1]. Occlusion attack analysis is used to test the capacity of recovering original images from cipher images even if some part of it has been lost or occluded. The occlusion analysis of images is shown in Table 21. In Table 21, the columns, Cipher-Occlusion-1, Cipher-Occlusion-2, Cipher-Occlusion-3 indicate the loss of image information in various regions which may likely happen while transmission. The corresponding decrypted or recovered images are shown in the rest of the columns. It is clearly evident from the tables that the images can be decrypted in a readable or understandable form even if some parts of cipher image are lost.

Table 19
Speckle noise analysis.

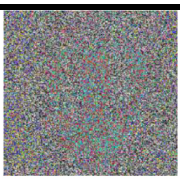
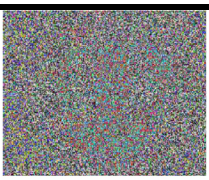
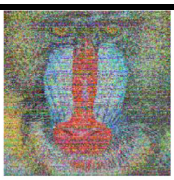
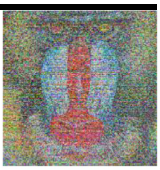
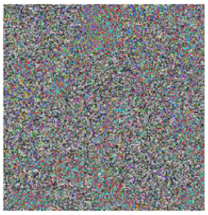
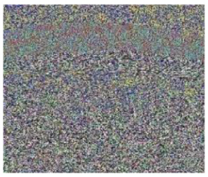
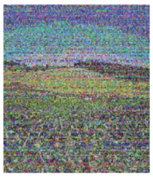
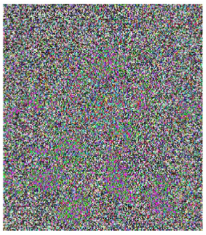
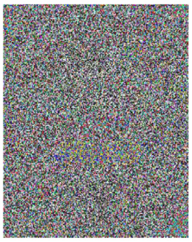
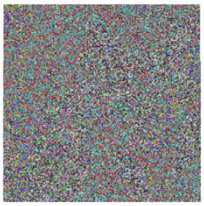
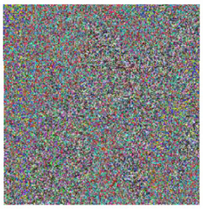
No	Cipher noise Low	Cipher noise default	Cipher noise high	Decrypted noise low	Decrypted noise default	Decrypted noise high
1						
2						
3						
4						
5						
6						

Table 20
PSNR.

Image name	PSNR (Peak Signal Noise Ratio)
Baboon	37.534
Barbara	37.1564
Cornfield	32.3818
Flower	34.3599
Lake	33.9109
Lena	36.0656
Monarch	34.6709
Peppers	33.6039
Soccer	36.8346
Yacht	36.8512

6.11. Running speed

A good encryption algorithm is expected to have a quick running speed. The colour images of different size have been used as examples to measure the encryption/decryption rate of the proposed image encryption scheme [53]. Our experiment runs on desktop PC with Intel(R) Core(TM) 2.16 *GHz CPU,* 4GB RAM and 500GB hard drive. The operating system and computational platform are Microsoft Windows 10 and Matlab R2017a respectively. The average encryption/decryption time taken by the algorithm for processing the images with size 256 × 256, 512 × 512 and 1024 × 1024 are 8.2 s, 16.43 s and 38 s respectively. Considering its high level of security, an acceptable running speed is achieved.

6.12. Performance comparison with recent works

This section analyses how well the various evaluation parameters of our system are improved over recent works. The Tables 22–27 show the comparison with recent popular works and all evaluation parameters like entropy, PSNR, correlation, etc. are improved.

The proposed algorithm has high sensitivity to the plain images, and it can resist the chosen/known-plaintext attacks, proving that our scheme is more secure than other schemes.

Table 21
Occlusion attack analysis.

Cipher occlusion 1	Cipher occlusion 2	Cipher occlusion 3	Decrypted occlusion 1	Decrypted occlusion 2	Decrypted occlusion 3

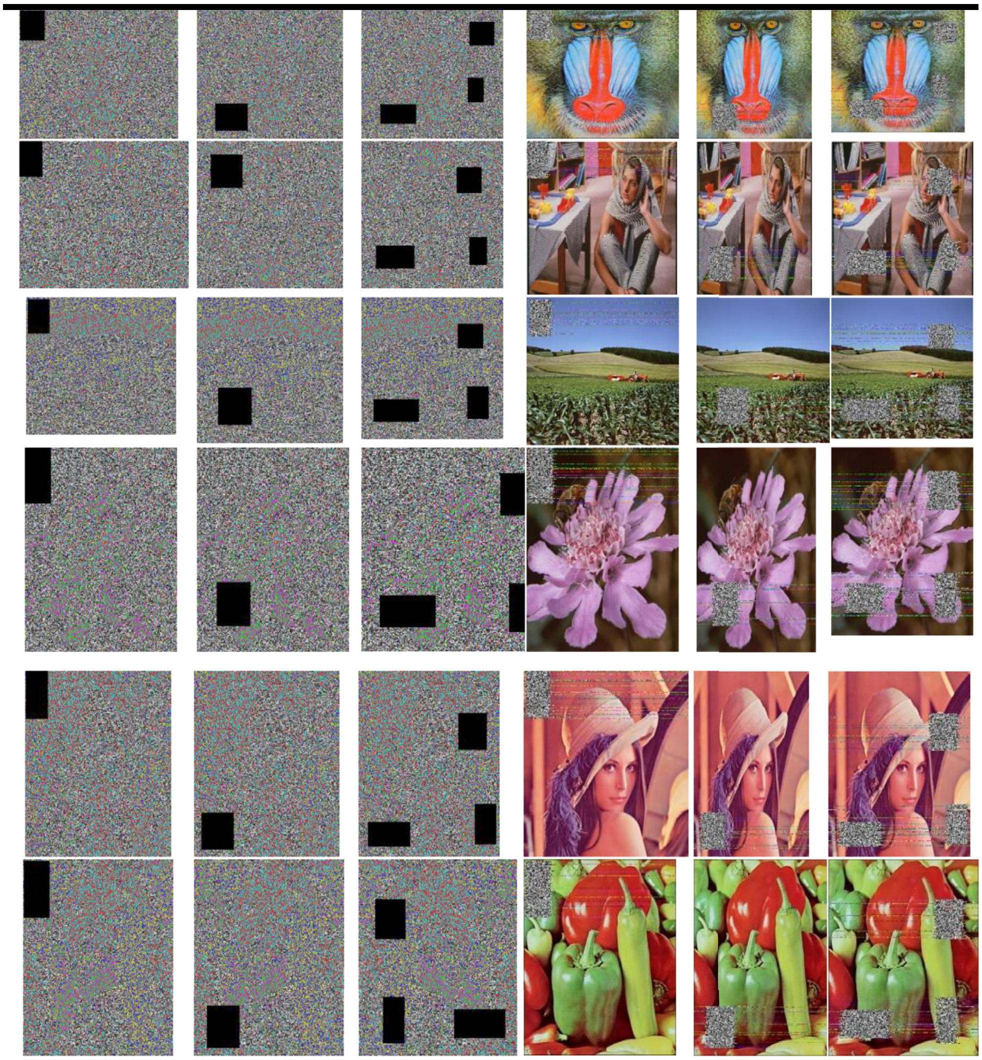


Table 22
PSNR analysis.

Existing Works	Image Name	PSNR value	Proposed
[54]	Lena	30.8397	**36.0656**
[55]	Monarch	28.6653	**34.6709**
[55]	Peppers	28.4145	**33.6039**

Table 23
Structural similarity analysis.

Image name	[36]	Proposed
Lena	0.5241	**0.9354**

Table 24
Entropy analysis -Lena.

Lena	256×256	512×512	Image Type
[1]		7.9993	GREY
[2]	7.9896		RGB
[3]	7.9971		GREY
[6]	7.9975	7.9994	GREY
[13]	7.9972		RGB
[40]	7.9983		RGB
[56]	7.990966		RGB
[57]	7.9972		RGB
[58]	7.9878		RGB
[59]	7.9952		RGB
[60]	7.9895		RGB
[61]	7.9927		RGB
[62]	7.9970		RGB
[31]	7.9896		RGB
[39]	7.9980		RGB
Proposed	**7.99924**	**7.9998**	**RGB**

Table 25
Entropy analysis –other images.

Existing Works	Image Name (256×256 – RGB)	Entropy	Entropy (Proposed Work)
[39]	Baboon	7.9990	7.99911
[63]	Barbara	7.9979	7.99908
[64]	Flower	7.9990	7.9991
[39]	Peppers	7.9977	7.99914

Table 26
Correlation analysis-Lena.

Lena	H(Horizontal)	V(Vertical)	D(Diagonal)
[3]	−0.0016	−0.0033	0.0130
[21]	0.0056	0.0037	0.0032
[54]	0.0129	0.0118	0.0088
[6]	0.0023	0.0019	0.0011
[56]	0.0041	0.0021	0.0009
[57]	0.0061	0.0116	0.0018
Proposed	**−0.00116**	**0.00106**	**−0.0043**

Table 27
NPCR and UACI analysis.

Existing Works	Image Name	NPCR	NPCR (Proposed Work)	UACI	UACI (Proposed Work)
[64]	Flower	0.9959	**0.9964**	0.333	**0.334**
[65]	Lena	0.9925	**0.9957**	0.333	**0.338**

7. Conclusion

In this work, an improved image encryption scheme based on chaotic map and DNA encoding is proposed. We have selected Arnold map as the best map to be used with our encryption system. The selection of the map is accomplished by devising a simple and proper map selection strategy. Here the encryption is applied separately on each of the three channels of the colour image to enhance the security. Chaos in image encryption has improved the security of transmitting image data. The cipher image generated by our algorithm is not possible to decrypt by the attacker since the encryption is done using the randomly generated sequences from chaotic maps and the large keyspace eliminates brute-force attacks. The analysis part of this work includes Histogram analysis, correlation analysis, Noise, Entropy, NPCR and UACI, etc. These evaluation parameters which are taken into consideration give higher values than the previous works and it proves that the algorithm yields a more secure way to transmit image data. In future, we can further improve the running speed of the algorithm by incorporating the concept of parallel permutation or parallel diffusion.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

We are thankful to DST-FIST, Government of India, for providing us the infrastructure to carry out this research.

References

[1] Chai X, Chen Y, Broyde L. A novel chaos-based image encryption algorithm using dna sequence operations. Opt Lasers Eng 2017;88:197–213. doi:10.1016/j.optlaseng.2016.08.009.

[2] Wu X, Wang K, Wang X, et al. Color image DNA encryption using nca map-based cml and one-time keys. Signal Process 2018;148:272–87. doi:10.1016/j.sigpro.2018.02.028.

[3] Chai X, Gan Z, Yang K, et al. An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations. Signal Process Image Commun 2017;52:6–19. doi:10.1016/j.image.2016.12.007.

[4] Chai X, Zheng X, Gan Z, et al. An image encryption algorithm based on chaotic system and compressive sensing. Signal Process 2018;148:124–44. doi:10.1016/j.sigpro.2018.02.007.

[5] Jacob G, Murugan A. DNA based cryptography: An overview and analysis. Int J Emerg Sci 2013;3:36–42.

[6] Enayatifar R, Abdullah AH, Isnin IF, et al. Image encryption using a synchronous permutation-diffusion technique. Opt Lasers Eng 2017;90:146–54. doi:10.1016/j.optlaseng.2016.10.006.

[7] Zhang YQ, He Y, Wang XY. Spatiotemporal chaos in mixed linear–nonlinear two-dimensional coupled logistic map lattice. Phys A Stat Mech Appl 2018;490:148–60. doi:10.1016/j.physa.2017.07.019.

[8] Zhang YQ, Wang XY. A new image encryption algorithm based on non-adjacent coupled map lattices. Appl Soft Comput J 2015;26:10–20. doi:10.1016/j.asoc.2014.09.039.

[9] Zhang YQ, Wang XY, Liu LY, et al. Spatiotemporal chaos of fractional order logistic equation in nonlinear coupled lattices. Commun Nonlinear Sci Numer Simul 2017;52:52–61. doi:10.1016/j.cnsns.2017.04.021.

[10] Zhang YQ, Wang XY, Liu J, Chi ZL. An image encryption scheme based on the mlncml system using DNA sequences. Opt Lasers Eng 2016;82:95–103. doi:10.1016/j.optlaseng.2016.02.002.

[11] Belazi A, Hermassi H, Rhouma R, Belghith S. Algebraic analysis of a rgb image encryption algorithm based on DNA encoding and chaotic map. Nonlinear Dyn 2014;76:1989–2004. doi:10.1007/s11071-014-1263-y.

[12] Wang XY, Li P, Zhang YQ, et al. A novel color image encryption scheme using DNA permutation based on the lorenz system. Multimed Tools Appl 2018;77:6243–65. doi:10.1007/s11042-017-4534-z.

[13] yuan Wang X, li Zhang H, mei Bao X. Color image encryption scheme using cml and DNA sequence operations. BioSystems 2016;144:18–26. doi:10.1016/j.biosystems.2016.03.011.

[14] Zhang X, Wang X. Multiple-image encryption algorithm based on DNA encoding and chaotic system. Multimed Tools Appl 2019;78:7841–69. doi:10.1007/s11042-018-6496-1.

[15] Mondal B, Mandal T. A light weight secure image encryption scheme based on chaos & DNA computing. J King Saud Univ Comput Inf Sci 2017;29:499–504. doi:10.1016/j.jksuci.2016.02.003.

[16] Zhang LY, Liu Y, Wang C, et al. Improved known-plaintext attack to permutation-only multimedia ciphers. Inf Sci (Ny) 2018;430–431:228–39. doi:10.1016/j.ins.2017.11.021.

[17] Zhang LY, Zhang Y, Liu Y, et al. Security analysis of some diffusion mechanisms used in chaotic ciphers. Int J Bifurc Chaos 2017;27:1–13. doi:10.1142/S0218127417501553.

[18] ALVAREZ G, LI S. SOME basic cryptographic requirements for chaos-based cryptosystems. Int J Bifurc Chaos 2006;16:2129–51. doi:10.1142/S0218127406015970.

[19] Zhang LY, Liu Y, Pareschi F, et al. On the security of a class of diffusion mechanisms for image encryption. IEEE Trans Cybern 2018;48:1163–75. doi:10.1109/TCYB.2017.2682561.

[20] Norouzi B, Seyedzadeh SM, Mirzakuchaki S, Mosavi MR. A novel image encryption based on row-column, masking and main diffusion processes with hyper chaos. Multimed Tools Appl 2013;74:781–811. doi:10.1007/s11042-013-1699-y.

[21] Wu J, Liao X, Yang B. Image encryption using 2D hénon-sine map and DNA approach. Signal Process 2018;153:11–23. doi:10.1016/j.sigpro.2018.06.008.

[22] Wang XY, Zhang YQ, Bao XM. A novel chaotic image encryption scheme using DNA sequence operations. Opt Lasers Eng 2015;73:53–61. doi:10.1016/j.optlaseng.2015.03.022.

[23] Enayatifar R, Guimarães FG, Siarry P. Index-based permutation-diffusion in multiple-image encryption using DNA sequence. Opt Lasers Eng 2019;115:131–40. doi:10.1016/j.optlaseng.2018.11.017.

[24] Li C, Liu Y, Xie T, Chen MZQ. Breaking a novel image encryption scheme based on improved hyperchaotic sequences. Nonlinear Dyn 2013;73:2083–9. doi:10.1007/s11071-013-0924-6.

[25] Gao T, Chen Z. A new image encryption algorithm based on hyper-chaos. Phys Lett Sect A Gen Solid State Phys 2008;372:394–400. doi:10.1016/j.physleta.2007.07.040.

[26] Patidar V, Pareek NK, Sud KK. A new substitution-diffusion based image cipher using chaotic standard and logistic maps. Commun Nonlinear Sci Numer Simul 2009;14:3056–75. doi:10.1016/j.cnsns.2008.11.005.

[27] Li C, Li S, Lo KT. Breaking a modified substitution-diffusion image cipher based on chaotic standard and logistic maps. Commun Nonlinear Sci Numer Simul 2011;16:837–43. doi:10.1016/j.cnsns.2010.05.008.

[28] Lian S. Efficient image or video encryption based on spatiotemporal chaos system. Chaos Solitons Fractals 2009;40:2509–19. doi:10.1016/j.chaos.2007.10.054.

[29] Liu H, Wang X, Kadir A. Image encryption using DNA complementary rule and chaotic maps. Appl Soft Comput J 2012;12:1457–66. doi:10.1016/j.asoc.2012.01.016.

[30] Norouzi B, Mirzakuchaki S. An image encryption algorithm based on DNA sequence operations and cellular neural network. Multimed Tools Appl 2017;76:13681–701. doi:10.1007/s11042-016-3769-4.

[31] Wu X, Kan H, Kurths J. A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps. Appl Soft Comput J 2015;37:24–39. doi:10.1016/j.asoc.2015.08.008.

[32] ur Rehman A, X Liao, Ashraf R, et al. A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2. Optik (Stuttg) 2018;159:348–67. doi:10.1016/j.ijleo.2018.01.064.

[33] Jeng FG, Huang WL, Chen TH. Cryptanalysis and improvement of two hyper-chaos-based image encryption schemes. Signal Process Image Commun 2015;34:45–51. doi:10.1016/j.image.2015.03.003.

[34] Aqeel-ur-Rehman Liao X, Hahsmi MA, Haider R. An efficient mixed inter-intra pixels substitution at 2bits-level for image encryption technique using DNA and chaos. Optik (Stuttg) 2018;153:117–34. doi:10.1016/j.ijleo.2017.09.099.

[35] Ismail SM, Said LA, Radwan AG, et al. Generalized double-humped logistic map-based medical image encryption. J Adv Res 2018;10:85–98. doi:10.1016/j.jare.2018.01.009.

[36] Abbas NA. Image encryption based on independent component analysis and arnold's cat map. Egypt Inform J 2016;17:139–46. doi:10.1016/j.eij.2015.10.001.

[37] Hasan MM, Faruqi TM, Tazrean M, Chowdhury TH. Biometric encryption using duffing map. In: Proceedings of the fourth international conference on advanced electrical engineering ICAEE 2017 2018; 2018. p. 737–42. doi:10.1109/ICAEE.2017.8255452.

[38] Wu X, Zhu B, Hu Y, Ran Y. A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps. IEEE Access 2017;5:6429–36. doi:10.1109/ACCESS.2017.2692043.

[39] Sneha PS, Sankar S, Kumar AS. A chaotic colour image encryption scheme combining walsh–hadamard transform and arnold–tent maps. J Ambient Intell Humaniz Comput 2019. doi:10.1007/s12652-019-01385-0.

[40] Kalpana J, Murali P. An improved color image encryption based on multiple DNA sequence operations with DNA synthetic image and chaos. Optik (Stuttg) 2015;126:5703–9. doi:10.1016/j.ijleo.2015.09.091.

[41] Xu M, Tian Z. Security analysis of a novel fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. Optik (Stuttg) 2017;134:45–52. doi:10.1016/j.ijleo.2017.01.029.

[42] Sun YY, Kong RQ, Wang XY, Bi LC. An image encryption algorithm utilizing mandelbrot set. In: Proceedings of the international workshop on complex-systems for future technologies and applications IWCFTA; 2010. p. 170–3. doi:10.1109/IWCFTA.2010.70.

[43] Vaferi E, Sabbaghi-Nadooshan R. A new encryption algorithm for color images based on total chaotic shuffling scheme. Optik (Stuttg) 2015;126:2474–80. doi:10.1016/j.ijleo.2015.06.012.

[44] Zhu H, Zhao C, Zhang X, Yang L. An image encryption scheme using generalized arnold map and affine cipher. Optik (Stuttg) 2014;125:6672–7. doi:10.1016/j.ijleo.2014.06.149.

[45] Fu XQ, Liu BC, Xie YY, et al. Image encryption-then-transmission using DNA encryption algorithm and the double chaos. IEEE Photonics J 2018;10. doi:10.1109/JPHOT.2018.2827165.

[46] Wang X, Liu L, Zhang Y. A novel chaotic block image encryption algorithm based on dynamic random growth technique. Opt Lasers Eng 2015;66:10–18. doi:10.1016/j.optlaseng.2014.08.005.

[47] Xu L, Gou X, Li Z, Li J. A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion. Opt Lasers Eng 2017;91:41–52. doi:10.1016/j.optlaseng.2016.10.012.

[48] Li J. Asymmetric multiple-image encryption based on octonion fresnel transform and sine logistic modulation map. J Opt Soc Korea 2016;20:341–57. doi:10.3807/JOSK.2016.20.3.341.

[49] Pal G, Verma V. Image encryption techniques under various noise attacks: a survey. Int J Softw Hardw Res Eng 2016;4:48–56.

[50] Pal G, Kumar V. Image encryption using adaptive pixel masking under various noise attacks. Int J Comput Appl 2017;164:12–16. doi:10.5120/ijca2017913587.

[51] Li J, Xiang S, Wang H, et al. A novel image encryption algorithm based on synchronized random bit generated in cascade-coupled chaotic semiconductor ring lasers. Opt Lasers Eng 2018;102:1339–51. doi:10.1016/j.optlaseng.2017.11.001.

[52] Jaryal S. Comparative analysis of various İmage encryption techniques. Int. J. Comput. Intell. Res. 2017;13(2):273–84.

[53] Zhou G, Zhang D, Liu Y, et al. A novel image encryption algorithm based on chaos and line map. Neurocomputing 2015;169:150–7. doi:10.1016/j.neucom.2014.11.095.

[54] Chai X, Yang K, Gan Z. A new chaos-based image encryption algorithm with dynamic key selection mechanisms. Multimed Tools Appl 2017;76:9907–27. doi:10.1007/s11042-016-3585-x.

[55] Thada V, Shrivastava U. A novel hybrid digital image encryption technique. Int J Comput Sci Eng 2018;6:585–92. doi:10.26438/ijcse/v6i5.585592.

[56] Li C, Luo G, Qin K, Li C. An image encryption scheme based on chaotic tent map. Nonlinear Dyn 2017;87:127–33. doi:10.1007/s11071-016-3030-8.

[57] Yaghouti Niyat A, Moattar MH, Niazi Torshiz M. Color image encryption based on hybrid hyper-chaotic system and cellular automata. Opt Lasers Eng 2017;90:225–37. doi:10.1016/j.optlaseng.2016.10.019.

[58] Liu H, Wang X. Color image encryption using spatial bit-level permutation and high-dimension chaotic system. Opt Commun 2011;284:3895–903. doi:10.1016/j.optcom.2011.04.001.

[59] Faraoun KM. Fast encryption of rgb color digital images using a tweakable cellular automaton based schema. Opt Laser Technol 2014;64:145–55. doi:10.1016/j.optlastec.2014.05.012.

[60] Liu H, Kadir A. Asymmetric color image encryption scheme using 2D discrete-time map. Signal Process 2015;113:104–12. doi:10.1016/j.sigpro.2015.01.016.

[61] Wang X, Zhao Y, Zhang H, Guo K. A novel color image encryption scheme using alternate chaotic mapping structure. Opt Lasers Eng 2016;82:79–86. doi:10.1016/j.optlaseng.2015.12.006.

[62] Wang X, Zhang HL. A color image encryption with heterogeneous bit-permutation and correlated chaos. Opt Commun 2015;342:51–60. doi:10.1016/j.optcom.2014.12.043.

[63] Gopalakrishnan T, Ramakrishnan S, Balakumar M. An image encryption using chaotic permutation and diffusion. In: Proceedings of the international conference on recent trends in information technology ICRTIT 2014; 2014.

[64] Di X, Li J, Qi H, et al. A semi-symmetric image encryption scheme based on the function projective synchronization of two hyperchaotic systems. PLoS One 2017;12:1–29. doi:10.1371/journal.pone.0184586.

[65] Auyporn W, Vongpradhip S. A robust image encryption method based on bit plane decomposition and multiple chaotic maps. Int J Signal Process Syst 2014;3. doi:10.12720/ijsps.3.1.8-13.