



# Metaheuristic-supported image encryption framework based on binary search tree and DNA encoding

Mousomi Roy<sup>1</sup> · Shouvik Chakraborty<sup>1</sup> · Kalyani Mali<sup>1</sup>

Received: 30 November 2021 / Revised: 21 June 2023 / Accepted: 8 August 2023 /

Published online: 19 August 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Rapid evolution in digital communication increases the risk of unauthorized access to sensitive data. Digital images are one of the most popular types and are frequently used to share various information including many sensitive data. Hence, security of the digital images is one of the major concerns for reliable data communication. This work addresses this issue and proposes a novel image encryption technique in which the concept of the binary search tree is introduced. The structure of the binary search tree is optimized with the help of the electromagnetism-like optimization approach that optimizes the image entropy. The proposed approach also incorporates the concept of DNA (Deoxyribonucleic acid) encoding and based on this concept bitplane decomposition is used to get DNA bit planes. A scrambling approach is also proposed to add an additional layer of security. This approach is a symmetric image encryption scheme and is completely lossless. The performance of this approach is tested in terms of both qualitative and quantitative manner. Experimental results and comparative outcomes with the state-of-the-art approaches are encouraging and prove the efficiency of the proposed approach.

**Keywords** Image encryption · Binary search tree · Electromagnetism-like optimization · DNA encoding · Bitplane decomposition

## 1 Introduction

Recent advancements in the technology helps to improve different domains that escalate the standard of living. The domain of digital data communication is no exception and experiencing a significant upliftment by utilizing the blessings of advanced technology in different ways. Image is one of the most important type of digital data

---

✉ Shouvik Chakraborty  
shouvikchakraborty51@gmail.com

Mousomi Roy  
iammouray@gmail.com

Kalyani Mali  
kalyanimali1992@gmail.com

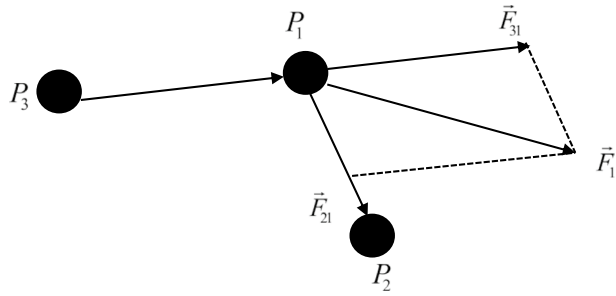
<sup>1</sup> Department of Computer Science & Engineering, University of Kalyani, Kalyani, India

that is being communicated and transferred over different types of media. The communication of images is increased significantly due to the advancements in various social media. Advent of different sophisticated technologies helps in different phases of digital image communication like image acquisition, management, interpretation, sharing, etc. From the users' perspective, these highly sophisticated tasks are equivalent to pressing or clicking some buttons. Such abstraction is achieved due to the significant technological advancements. Like other data types, digital images also contain various sensitive information that are always susceptible to leakage. So, in most of the real-life scenarios, it is essential to protect the transmitted data from the intruders and hence, the security of the digital images is one of the major challenges faced by the concerned researchers. A continuous effort can be observed from the various researchers to impose the security measures on the digital images in different phases of the digital image communication. It is essential to ensure appropriate security in different phases of data transmission like storage area, transmission channel, and so on. In some scenarios like military applications, the security of the data can never be compromised and it is the utmost priority for the parties involved in the communication process. The intensity of this threat can be easily understood when we think about losing sensitive images to the enemy countries. Various data security methods are proposed like DES [9], AES [11], IDEA [22], chaotic theory based methods [7, 31, 34, 41, 50, 56], DNA computing based methods [10, 14, 15, 45] to name a few. It is observed in different scenarios that some of the popular and widely used encryption systems that suit for the text and/or binary data do not fit well for the image, video, and other multimedia data [20, 35]. Moreover, it is also observed that many traditional data security approaches demand high computational resources that are not always possible to supply in many real-life scenarios like in unguided data communication [36, 44]. Some of the well-known traditional approaches DES, AES, etc. suit well for the independent and identically distributed (IID) data collection. Typically, various multimedia data including images do not follow such distribution. So, the direct application of some traditional approaches like DES, AES, etc. may not be possible in some circumstances [16]. Here comes the need for some specialized and dedicated approaches that suit well for various multimedia data. In general, meaningful images consist of high amount of pixel correlation and it is experimentally proven that traditional approaches may not always perform well on such kind of data and left some sections of the original image intelligible [12, 16, 25]. Motivated from this, a novel data security approach is proposed for the image data and it is always a challenging job for the researchers to develop an efficient approach to secure image data.

In this work, a symmetric image encryption approach is proposed for image encryption i.e., the same key is required for encrypting the original image as well as to decrypt the encrypted image. Typically, the symmetric image encryption systems comprise of two major steps i.e., permutation and diffusion [7, 23]. DNA computing is one of the recent approaches that is being frequently used for the image encryption purposes. Typically, DNA computing is used to encode the pixels of the plain image into the DNA bases. This article proposes a novel approach that hybridize the DNA computing and the randomness of the BST (Binary search tree) to get an efficient cryptographic framework. The EMO (Electromagnetism-like optimization) approach is used to determine the optimal structure of the BST that optimizes the overall entropy of the encrypted image.

The arrangement of the rest of the article is as follows: fundamental concepts of the electromagnetism-like optimization approach are discussed in Section 2. Section 3

**Fig. 1** Demonstration of computing resultant force using superposition principle



describes the proposed approach in detail. The performance of the proposed approach is evaluated and presented in Section 4. Section 5 concludes the article.

## 2 Fundamentals of the electromagnetism-like optimization approach

Electromagnetism-like optimization (EMO) approach is one of the metaheuristic approaches that are recently developed. In general, metaheuristic optimization methods are used in solving different real-life problems [5, 8]. The EMO approach was originally developed by Birbil and Fang [4]. This optimization approach is based on the attraction-repulsion phenomena of the charged particles [6]. A randomly-generated pool of initial solutions is created that mimics the pool of some charged particles, and it is the first phase of this approach. The second phase is dedicated to find the local optima by using a local search approach. Some of the standard methods like hill-climbing [43], gradient descent [2], etc. can be used for this purpose. The superposition principle [24] is used in the third phase to determine the exerted force by the charged particles i.e. the solutions. It is illustrated in Fig. 1.

In this figure,  $\vec{F}_{pq}$  denotes the exerted force by the particle  $P_q$  that is caused by another particle  $P_p$ . This force is used to implement the attraction-repulsion mechanism. According to Fig. 1,  $P_1$  is repelled by  $P_3$  and attracted by  $P_2$ . So, according to the superposition principle the resultant force acting on  $P_1$  that is responsible of the movement of the particles is expressed by  $\vec{F}_1 = \vec{F}_{31} + \vec{F}_{21}$ . The EMO can optimize the non-linear optimization problem along with the box constraint as shown in Eq. (1).

$$\max f(x), x = (x_1, x_2, x_3, \dots, x_n) \in \mathbb{R}^n \text{ subject to } x \in X \quad (1)$$

In this equation,  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  denotes a non-linear function, and  $X = \{x \in \mathbb{R}^n \mid l_i \leq x_i \leq h_i, i = 1, 2, \dots, n\}$  where the upper and the lower bounds are represented using  $h_i$  and  $l_i$ . The Basic EMO approach is illustrated in algorithm 1. Individual modules are illustrated in algorithms 2-4.

**Algorithm 1** The fundamental electromagnetism-like optimization (EMO) method

**Inputs:** The size of the initial population (denoted by  $I$ ), the maximum count of the global and local iterations are denoted using  $\max_{glr}$ , and  $\max_{llr}$  respectively. The controlling parameter of the local search procedure that is denoted by  $\psi \in [0,1]$ .

**Output:** Result(s) after optimization.

- 1:  $pop \leftarrow initializePop(I)$ : Initialization of the initial population of size  $I$ .
- 2:  $fit \leftarrow calcFitness(pop)$ : computing the fitness value using the problem specific fitness function.
- 3:  $localSearch(\max_{llr}, \psi, n, d)$ : perform the local search.
- 4:  $force \leftarrow calcForce(fit)$ : calculate the exerted force
- 5:  $moveParticles(force)$ : move the particles based on the computed exerted force.
- 6: Repeat the last three steps until the termination criteria gets satisfied.

**Algorithm 2**  $localSearch(ItrCount, \lambda, n, d)$ : executes the local search method

**Input:** Maximum number of iterations  $ItrCount$ , the local search parameter  $\lambda \in [0,1]$ , Size of the population  $n$  and the dimension of each point  $d$ .

**Output:** Updated population and the best fitness value of this population  $\zeta^{best}$ .

- 1: Set  $cnt \leftarrow 1$
- 2: Set  $len \leftarrow \lambda \left( \max_j \{I_{high}^j - I_{low}^j\} \right)$
- 3: Repeat the following for  $i = 1 : n$  times
- 4:     Repeat the following for  $j = 1 : d$  times
- 5:         Generate a random value  $\tau_1 = random(0,1)$
- 6:         Repeat the following steps while  $cnt < ItrCount$
- 7:              $\kappa \leftarrow \zeta^i$
- 8:             Generate a random value  $\tau_2 = random(0,1)$
- 9:             Check if  $\tau_1 > 0.5$  then
- 10:                  $\kappa_j \leftarrow \kappa_j + \tau_2(len)$
- 11:             otherwise
- 12:                  $\kappa_j \leftarrow \kappa_j - \tau_2(len)$
- 12:             end if
- 13:             Check if  $fitness(\zeta^i) > fitness(\kappa)$  then
- 14:                  $\zeta^i \leftarrow \kappa$
- 15:                  $cnt \leftarrow ItrCount - 1$
- 16:             end if
- 17:              $cnt \leftarrow cnt + 1$
- 17:         end while
- 17:     end for
- 17:     end for
- 18: Compute the fitness of each point  $fitness(i) \leftarrow calcFitness(\zeta^i)$
- 19:  $\zeta^{best} \leftarrow \arg \min \{fitness(i), \forall i\}$

**Algorithm 3** *calcForce(fitness)*: calculate force**Input:** fitness of the points *fitness*.**Output:** Computed force  $\vec{F}$ 


---

```

1: Repeat for  $i = 1 : n$  times
2:   Compute  $ch^i$  using equation 9
3:    $\vec{F}_i \leftarrow 0$ 
   end for
4: Repeat for  $i = 1 : n$  times
5:   Repeat for  $i = 1 : n$  times
6:     If  $fitness(j) < fitness(i)$  then
7:        $\vec{F}_i \leftarrow \vec{F}_i + (\zeta^j - \zeta^i) \frac{ch^i ch^j}{\|\zeta^j - \zeta^i\|^2}$  // Attraction force
8:     Otherwise
9:        $\vec{F}_i \leftarrow \vec{F}_i - (\zeta^j - \zeta^i) \frac{ch^i ch^j}{\|\zeta^j - \zeta^i\|^2}$  // Repulsion force
   end if
   end for
end for

```

---

**Algorithm 4** *moveParticles(force)*: move particles depending on their exerted force**Input:** Exerted force by each particle *force***Output:** Moved particles

---

```

1: Repeat for  $i = 1 : n$  times
2:   Check if  $i \neq best$  then
3:      $\varepsilon \leftarrow random(0,1)$ 
4:      $\vec{F}_i \leftarrow \frac{\vec{F}_i}{\|\vec{F}_i\|}$ 
5:     Repeat for  $j = 1 : d$  times
6:       Check if  $\vec{F}_i^j > 0$  then
7:          $\zeta_j^i \leftarrow \zeta_j^i + \varepsilon \cdot \vec{F}_i^j \cdot (I_{high}^j - \zeta_j^i)$ 
8:       Otherwise
9:          $\zeta_j^i \leftarrow \zeta_j^i + \varepsilon \cdot \vec{F}_i^j \cdot (\zeta_j^i - I_{low}^j)$ 
       end if
     end for
   end if
end for

```

---

### 3 The proposed image encryption approach

A detailed explanation of the proposed work is given in this section. The encryption system begins by taking 128 bits randomly chosen secret key  $k_c$ . After that, a pseudorandom bit sequence is generated with the help of the chaos theory. In this work, the chaotic logistic map is used for this purpose [7]. The chaotic logistic map is one of the most frequently used maps in the domain of cryptography and it is mathematically defined in Eq. (2) [27].

$$q_{\delta+1} = f(q_{\delta}) = r \cdot q_{\delta} \cdot (1 - q_{\delta}) \quad (2)$$

The chaotic logistic map is consisting of two parameters i.e.,  $q$  and  $r$  where  $q$  denotes the state variable of the chaotic logistic function and  $r$  is known as the controlling parameter. The state variable  $q$  can take any value from the range  $[0, 1]$  whereas the controlling parameter  $r$  can take any value from the range  $[1, 4]$ . The  $\delta^{th}$  value of the state variable  $q$  is denoted as  $q_{\delta}$  and the initial value of the state variable is denoted by  $q_0$ . Two chaotic logistic maps are given in Eqs. (3) and (4) respectively and these two maps are used to generate the pseudorandom bit sequence using Eq. (5) and the generation of the pseudorandom bit sequence is a deterministic process. Statistical independence and bias-free nature are some of the characteristics of the generated pseudorandom bit sequence [28].

$$s_{\delta+1} = f(s_{\delta}) = r \cdot s_{\delta} \cdot (1 - s_{\delta}) \quad (3)$$

$$t_{\delta+1} = f(t_{\delta}) = \hat{r} \cdot t_{\delta} \cdot (1 - t_{\delta}) \quad (4)$$

$$ps_{\delta} = PRBG(s_{\delta}, t_{\delta}) = \begin{cases} 0 & \text{if } s_{\delta} \leq t_{\delta} \\ 1 & \text{if } s_{\delta} > t_{\delta} \end{cases} \quad (5)$$

In Eqs. (3) and (4), initial values of the state variables  $s$  and  $t$  i.e.  $s_0$  and  $t_0$  are known as the seeds where  $s_0 \in [0, 1]$ ,  $t_0 \in [0, 1]$ , and  $s_0 \neq t_0$ . Following features [28] must be followed by the generated pseudorandom bit sequence to have the ‘perfect cryptographic features’ [40].

1. Trajectories of the two chaotic maps must be independent when  $n \rightarrow \infty$ .
2. Both involved chaotic maps must be surjective in a particular interval.
3. The density distribution for the chaotic maps must be unique and invariant.
4. Both chaotic maps must consist the ergodicity property at some particular interval.
5. According to a certain point, both chaotic maps should possess equal or symmetric density distribution.

Now, to satisfy the second point i.e., the surjectivity of the chaotic maps, it is essential to set the values of the controlling parameters for both chaotic maps same i.e.,  $r = \hat{r}$ . A large value should be considered (i.e., close to 4) to get a large interval for the initial seeds of the involved chaotic logistic maps. The length of the generated pseudorandom bit sequence  $k_p$  is 128 bits. Bitwise XOR operation is performed between the initially selected 128 bits key and the 128 bits pseudorandom key sequence as given in Eq. (6).

$$k_x \leftarrow k_c \oplus k_p \quad (6)$$

In this work, the concept of binary search tree used for the encryption purpose. The height of the binary search tree cannot exceed more than the number of rows if the binary

search tree is applied in row-wise fashion and similarly, the height of the binary search tree cannot be more than the number of columns if it is applied in the column-wise fashion. It means, for an image of size  $d_1 \times d_2$ , the height  $h$  of the binary search tree  $b$  must be less than or equal to  $d_1$  i.e.  $h \leq d_1$  if the binary search tree is applied in row-wise fashion and it must be less than or equal to  $d_2$  i.e.  $h \leq d_2$  if the binary search tree is applied in column-wise fashion. So, the number of nodes  $N$  in the binary search tree should be carefully chosen. The value of  $N$  can be calculated using Eq. (7) or Eq. (8) depending on the row-wise or column-wise implementations respectively.

$$N = \lfloor \varphi \cdot d_1 \rfloor \quad (7)$$

$$N = \lfloor \varphi \cdot d_2 \rfloor \quad (8)$$

Now, to determine the value of  $\varphi$ ,  $k_x$  is divided into 16 bytes i.e.,  $[k_{x0}, k_{x2}, \dots, k_{x15}]$ . The value of  $\varphi$  can be calculated using Eq. (9).

$$\varphi = \left[ \frac{\frac{k_{x0} \oplus k_{x2} \oplus \dots \oplus k_{x14}}{2^8} + \frac{k_{x1} \oplus k_{x3} \oplus \dots \oplus k_{x15}}{2^8} + \frac{\sum_{i=0}^{15} k_{xi}}{2^{12}}}{3} \right] \quad (9)$$

The main motivation behind designing Eqs. (7) and (8) is the fact that a fully skewed binary search tree of height  $\hat{h}$  can have at most  $\hat{N}$  nodes where  $\hat{N} = \hat{h}$ . A pseudorandom bit sequence  $ps$  is generated using Eq. (5) of length  $\tau$  where the length is defined in Eq. (10). The pseudorandom bit sequence  $ps$  is divided into  $N$  groups each consisting of 8 bits as shown in Eq. (11). Now each group is converted into decimal form and assigned in the nodes of the BST. In this way, the binary search tree is prepared.

$$\tau = N \cdot 8 \quad (10)$$

$$ps = [(b_1, \dots, b_8)_1, (b_9, \dots, b_{16})_2, \dots, (b_{\tau-7}, \dots, b_{\tau})_N] \quad (11)$$

The DNA encoding strategy is adopted in this work to encode the actual image as well as the binary search tree. There are four major bases adenine (A), cytosine (C), guanine (G), and thymine (T) that contribute to the fundamental DNA encoding schemes. Therefore, two bits are sufficient to represent four combinations. Now, base 'A' can only be replaced with base 'T' and vice-versa. Similarly, base 'G' can only be replaced with base 'C' and vice-versa. Depending on this concept, 8 rules for DNA encoding rules can be constructed and it is given in Table 1.

These rules are used to convert the binary search tree as well as the plain image into their DNA equivalents. For each node of the binary search tree, one of the 8 rules are selected dynamically using Eq. (12) and for every pixel, one rule is selected using Eq. (13).

**Table 1** Eight possible rules for DNA encoding and decoding rules

Binary combinations	R #1	R #2	R #3	R #4	R #5	R #6	R #7	R #8
00	A	A	T	T	C	C	G	G
01	C	G	C	G	A	T	A	T
10	G	C	G	C	T	A	T	A
11	T	T	A	A	G	G	C	C

**Table 2** Biological XOR operation

$\oplus$	A	T	G	C
A	A	T	G	C
T	T	A	C	G
G	G	C	A	T
C	C	G	T	A

$$\hat{r}_i = \lfloor v_i \cdot 7 \rfloor + 1, i = 1, 2, \dots, N \quad (12)$$

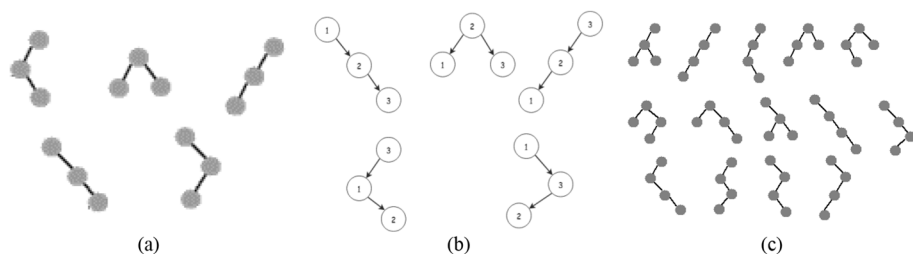
$$\hat{r}_k = \lfloor \omega_k \cdot 7 \rfloor + 1, k = 1, 2, 3, \dots, (d_1 \times d_2) \quad (13)$$

$$C_N = \frac{(2 \cdot N)!}{(N+1)! \cdot N!} \quad (14)$$

In Eq. (12),  $v_i$  denotes the  $i^{\text{th}}$  value of the chaotic sequence generated using Eq. (2). Similarly, in Eq. (13),  $\omega_k$  denotes the  $k^{\text{th}}$  value of the chaotic sequence generated using Eq. (2).  $d_1$  and  $d_2$  denotes the number of rows and columns in the original image, therefore,  $(d_1 \times d_2)$  denotes the total number of pixels present in the original image. After encoding both, the original image and the BST into the DNA bases, the biological XOR operation is performed between the BST and the original image using Table 2. Before performing the XOR operation, it is essential to decide the structure of the BST because, more than one BSTs are possible. In general, Eq. (22) can be used to find the number of BSTs possible with N nodes. For example, if  $N=3$ , then  $C_N=5$  and if  $N=4$ , then  $C_N=14$ . This fact is depicted in Fig. 2.

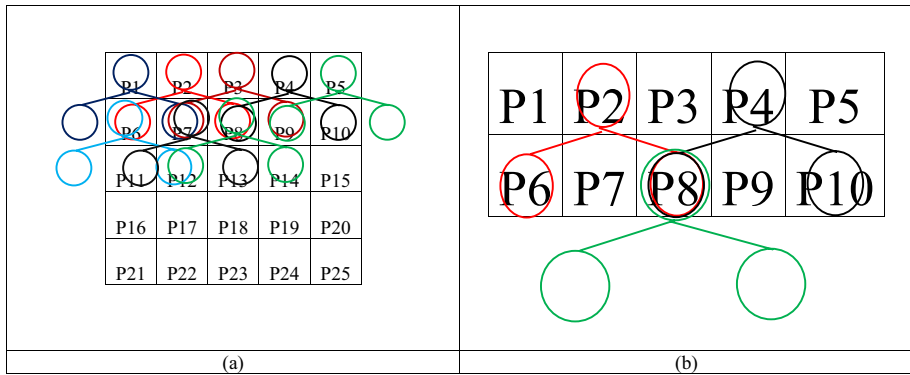
The root of the binary search tree is placed at the beginning of the actual image to perform the biological XOR operation. Other nodes are also positioned accordingly. The process can be repeated either column wise or row wise. In this work, row wise execution is considered. It can be understood that except the first row, some pixels will be processed more than one times depending on the structure of the BST. It is explained in Fig. 3. From Fig. 3, it can be observed that a single pixel is processed for multiple times except the first row. For example, pixel P6 is processed twice whereas pixel P7 is processed three times and so on. A better visualization can be obtained from Fig. 3b. Here, the pixel P8 is processed thrice with three different BSTs.

From the above discussion, it can be understood that different structure of BST is can led to different results. Exhaustive search to determine the optimal structure of the BST can



**Fig. 2** Different structures of binary search tree (a) with three nodes (5 structures), (b) all possible BSTs with three nodes and key values 1,2, and 3, (c) with four nodes (14 structures)

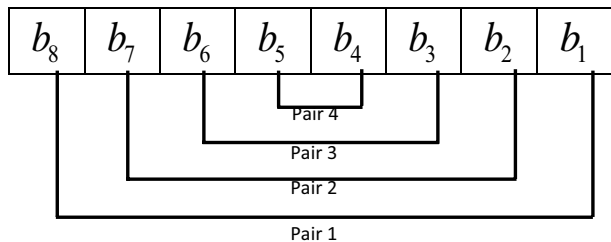




**Fig. 3** **a** Superimposing BST on an image (row wise and left to right manner). Each color represent the same BST applied different times. **b** Illustrating the same concept for a single pixel (P8) for better visualization

be computationally expensive for large images. So, in this work, the structure of the BST is determined with the help of a recently developed metaheuristic approach called Electromagnetism-like optimization (EMO) [4]. The initial population is created by randomly choosing some of the BST structures from all possible set of BSTs. The EMO algorithm tries to optimize the entropy by determining the best possible BST structure. A brief overview of the EMO approach is already given in the previous section. The EMO approach treats every solution as a charged particle. Depending on the exerted forces the particles are moved. In this case, the charge of the particle is computed by computing the value of the objective function i.e., the value of the entropy. After this step, the biologically XORed image will be obtained which is decoded (i.e., DNA bases to pixels) and further supplied to another layer. This layer is used to scramble the input image that is used to add an extra layer of security (Fig. 4).

**Fig. 4** Initial pairing of bits for the scrambling algorithm



**Algorithm 5** The proposed scrambling algorithm

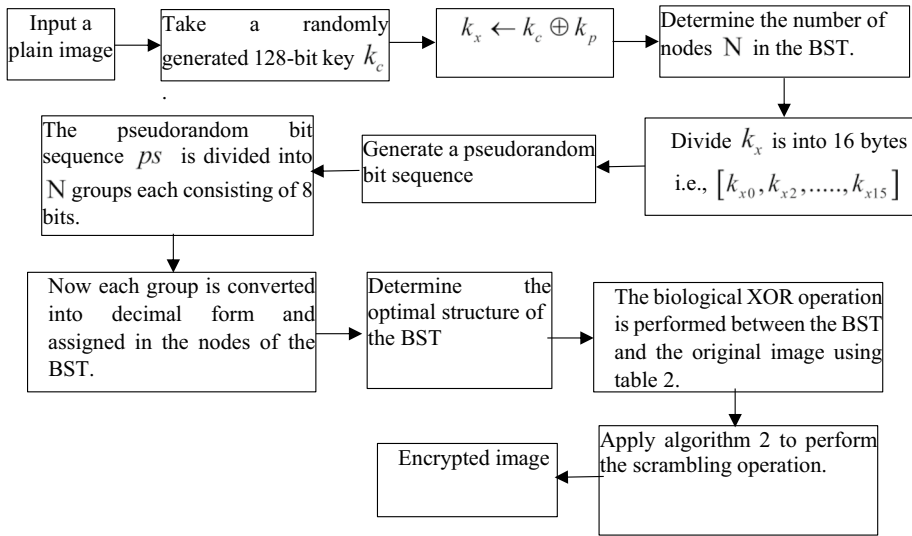
**Input:** Decoded image  $IMG$  obtained from the previous step.

**Output:** The scrambled image  $IMG'$ .

1. Convert each pixel into 8-bit binary form and pick one bit from beginning and one bit from end to form a pair. In this way four pairs can be formed as depicted in Figure 4. Therefore, the four pairs are  $(b_8, b_1)$ ,  $(b_7, b_2)$ ,  $(b_6, b_3)$ , and  $(b_5, b_4)$ .
2. For a certain pixel, use equation (15) to decide one of the DNA encoding rules and encode the above pairs of pixels using the selected rule. Here,  $(i, j)$  denotes the position of a pixel in two-dimensional grid.
3. So, for every pixel, four DNA bases are created. Now using these bases four DNA bitplanes are constructed. So, an actual image is converted into 4 bitplanes and represented using a three-dimensional array  $Y$ .
4. A one-dimensional vector  $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$  is constructed from the 4 bitplanes where  $\lambda_{(i-1)*d_2+j+(k-1)*m/8} = Y(i, j, k)$ ,  $m = d_1 \times d_2 \times b$ ,  $d_1 \times d_2$  is the dimension of the actual image, and  $b$  is the number of bitplanes.
5. DNA XOR operation is performed on the successive bases of the vector  $\lambda$  using table 2, and store the result in the lower-order position between the two successive base positions i.e.,  $\lambda_k \leftarrow \lambda_k \oplus \lambda_{k+1}, k \geq 1$ . This operation leaves the last base of the vector  $\lambda$  unaltered.
6. A pseudorandom bit sequence  $prbs$  is generated by following the method described in [33]. This approach involves the Chen chaotic environment. The length of the  $prbs$  is  $2 \cdot m$ .
7. Consecutive two bits of  $prbs$  are paired and encoded into DNA bases using equation (15) and store it in  $prds$ . Here,  $j = i + 1$ .
8. Perform biological XOR operation between  $prds$  and  $\lambda$  using table 2, and store it in  $\lambda$  i.e.,  $\lambda_j = \lambda_j \oplus prds_j, j \geq 1$ .
9.  $cnt_A \leftarrow \text{no. of 'A's present in } \lambda$ .
10. Iterate chaotic logistic map  $(cnt_A - 1)$  number of times with the same value of the controlling parameter as used in equations (3) and (4) but with different seed to get the  $cnt_A^{th}$  element of the chaotic sequence.
11. Iterate the chaotic logistic map and generate a chaotic sequence  $c = \{c_1, c_2, \dots, c_m\}$  with slightly different value of the controlling parameter compared to the previous step and consider the seed value as the  $cnt_A^{th}$  element of the chaotic sequence that is obtained in the previous step.
12. Apply insertion sort to sort the chaotic sequence  $c = \{c_1, c_2, \dots, c_m\}$  in descending order and generate the sorted sequence  $c' = \{c'_1, c'_2, \dots, c'_m\}$ .
13. The vector  $\lambda$  is modified and rearranged to generate another vector  $\lambda' = \{\lambda'_1, \lambda'_2, \dots, \lambda'_m\}$  where  $\lambda'_\alpha = \lambda_\alpha$  when  $c'_\alpha = c_\alpha$ . So,  $\lambda'$  is one of the permutations of  $\lambda$ .
14. Rebuild the DNA bitplanes  $Y'$  using  $\lambda'$  where  $Y'(i, j, k) = \lambda'_{(i-1)*d_2+j+(k-1)*m/4}$ .
15. Combine the bitplanes and decode the DNA bases into its equivalent binary for to get generate the scrambled image  $IMG'$ .

$$ruleNo = [(i + j) \bmod 8] + 1 \quad (15)$$

The overall proposed image encryption algorithm is illustrated in algorithm 6 and the flow diagram of the proposed approach is illustrated in Fig. 5.



**Fig. 5** Flow diagram of the proposed image encryption approach

**Algorithm 6** The proposed image encryption algorithm

**Input:** Input image.

**Output:** Encrypted image.

1. Take a randomly generated 128-bit key  $k_c$ .
2. Use equation 13 to generate a 128-bit pseudorandom bit sequence  $k_p$ .
3. Perform bitwise XOR operation between  $k_c$  and  $k_p$  and store it in  $k_x$  i.e.,  $k_x \leftarrow k_c \oplus k_p$ .
4. Apply equation (7) or equation (8) to determine the number of nodes  $N$  in the BST.
5. To determine the value of  $\varphi$ ,  $k_x$  is divided into 16 bytes i.e.,  $[k_{x0}, k_{x2}, \dots, k_{x15}]$ .
6. The value of  $\varphi$  can be calculated using equation (9).
7. A pseudorandom bit sequence  $ps$  is generated using equation (5) of length  $\tau$  where the length is defined in equation (10).
8. The pseudorandom bit sequence  $ps$  is divided into  $N$  groups each consisting of 8 bits as shown in equation (11).
9. Now each group is converted into decimal form and assigned in the nodes of the BST.
10. For each node of the binary search tree, one of the 8 rules is selected dynamically using equation (12) and for every pixel, one rule is selected using equation (13).
11. Determine the optimal structure of the BST by optimizing entropy using the EMO approach.
12. The biological XOR operation is performed between the BST and the original image using table 2. The root of the binary search tree is placed at the beginning of the actual image to perform the biological XOR operation. Other nodes are also positioned accordingly. The process can be repeated in either column wise or row wise fashion.
13. Apply algorithm 5 to perform scrambling.
14. Return the encrypted image.

The decryption procedure is simple and briefly presented in algorithm 7.

**Algorithm 7** The proposed image decryption algorithm

**Input:** Encrypted image, structural information of the optimized BST.

**Output:** Decrypted/Original image.

1. Perform the reverse scrambling
2. Take a randomly generated 128-bit key  $k_c$  (same key for the encryption process).
3. Use equation 13 to generate a 128-bit pseudorandom bit sequence  $k_p$ .
4. Perform bitwise XOR operation between  $k_c$  and  $k_p$  and store it in  $k_x$  i.e.,  $k_x \leftarrow k_c \oplus k_p$ .
5. Apply equation (7) or equation (8) to determine the number of nodes  $N$  in the BST.
6. To determine the value of  $\varphi$ ,  $k_x$  is divided into 16 bytes i.e.,  $[k_{x0}, k_{x2}, \dots, k_{x15}]$ .
7. The value of  $\varphi$  can be calculated using equation (9).
8. A pseudorandom bit sequence  $ps$  is generated using equation (5) of length  $\tau$  where the length is defined in equation (10).
9. The pseudorandom bit sequence  $ps$  is divided into  $N$  groups each consisting of 8 bits as shown in equation (11).
10. Now each group is converted into decimal form and assigned in the nodes of the BST (whose structural information is already available).
11. For each node of the binary search tree, one of the 8 rules is selected dynamically using equation (12) and for every pixel, one rule is selected using equation (13).
12. The biological XOR operation is performed between the BST and the original image using table 2. The root of the binary search tree is placed at the beginning of the actual image to perform the biological XOR operation. Other nodes are also positioned accordingly. The process can be repeated in either column wise or row wise fashion.
13. Return the decrypted image.

For any encryption scheme, including DNA encryption, can potentially be vulnerable to cryptanalysis if there are flaws in the design, implementation, or key management. Cryptanalysis involves analyzing the encryption algorithm and attempting to find weaknesses that could be exploited to recover the original data without knowledge of the decryption key. DNA encryption poses unique challenges due to the nature of DNA and the complexity of encoding, sequencing, and decoding processes. It is crucial to develop encryption algorithms specifically designed for DNA data and evaluate their resistance against known cryptographic attacks. It is also worth mentioning that advancements in technology and computing power can impact the security of any encryption scheme. As computing power increases, the feasibility of certain cryptanalytic attacks may change. There are several advantages of DNA-based encryption such as high data storage capacity, parallel processing, potential for data longevity, etc. DNA-based encryption can provide opportunities for data hiding within the DNA sequences themselves. This could allow for covert transmission or storage of sensitive image data within larger DNA samples. DNA-based encryption draws inspiration from biological systems, which have evolved to be highly secure and resilient. By mimicking biological processes, DNA-based encryption schemes may offer innovative and robust security features. In this work, the DNA encoding approach is used along with the chaos theory and BST. Chaos theory is well-known in the context of image encryption. The overall design of this cryptographic framework makes the system resilient against various attacks and the experimental results prove the effectiveness of the proposed approach.

In Electromagnetism-like Optimization (EMO), the controlling parameters determine how the algorithm explores and exploits the search space. EMO uses a population of particles or agents to explore the search space. This parameter determines the number of particles involved in the optimization process. Generally, a larger population size allows for better exploration but may also increase computational complexity. In EMO, particles are assigned charges, which represent their fitness or objective value. Charges influence the attractiveness or repulsiveness of particles towards each other during the optimization process. Particles with higher charges are considered more attractive and tend to be more influential. EMO particles move through the search space by adjusting their positions and velocities. The position update rule determines how particles explore the space, while the velocity update rule affects the particles' movement dynamics. These rules are usually based on electromagnetic principles, such as Coulomb's law and the Lorentz force equation. The scaling factor determines the impact of the charges on the forces acting between particles. It helps control the strength of attraction or repulsion between particles based on their charges. EMO often incorporates an interaction radius that limits the range within which particles interact with each other. Particles within this radius influence each other's movement based on their charges and distances. The termination criteria define when the optimization process should stop. They can include reaching a maximum number of iterations, achieving a satisfactory solution, or when the improvement rate falls below a certain threshold.

The controlling parameters are tuned empirically. The values of some controlling parameters are given in Table 3.

## 4 Results of the experiments

The obtained results after applying the proposed image encryption system are reported in this section. All experiments are carried out in desktop computer with Intel i5 Processor, 8GB RAM, 1 TB HDD, 240 GB SSD. A detailed analysis of the obtained results is presented in a comprehensive manner that is helpful to understand the efficiency and the practical applicability of the proposed image encryption system. Different types of images are investigated to test and establish the effectiveness of the proposed approach. The experimental outcomes of the experiments that are performed on some of the images are depicted in Fig. 6. Apart from these images, the proposed method is applied on several other images and the obtained results are promising. It can be observed that the proposed approach is quite effective in encrypting both grayscale and color images. The color images contain three separate channels i.e., Red 'R', Green 'G', and Blue 'B'. These channels are encrypted separately and combined to obtain the encrypted outcomes. Fig. 6 consists of three different columns where the first column i.e., Fig. 6a shows the original images, Fig. 6b shows

**Table 3** Controlling parameters and their values

Parameter	Value
Dimension of the Problem (n)	2
Size of the initial population (I)	50
maximum number of iterations ( $max_{gIter}$ )	75
maximum number of local search iterations ( $max_{lIter}$ )	20
local search parameter ( $\psi$ )	0.005

**Fig. 6** Result of the encryption and decryption process after applying the proposed approach (a) Original Images (from top to bottom: cameraman, barbara, boat, mountain, goldhill, airplane, baboon, cat, frymire, and girl), (b) encrypted images, and (c) decrypted images

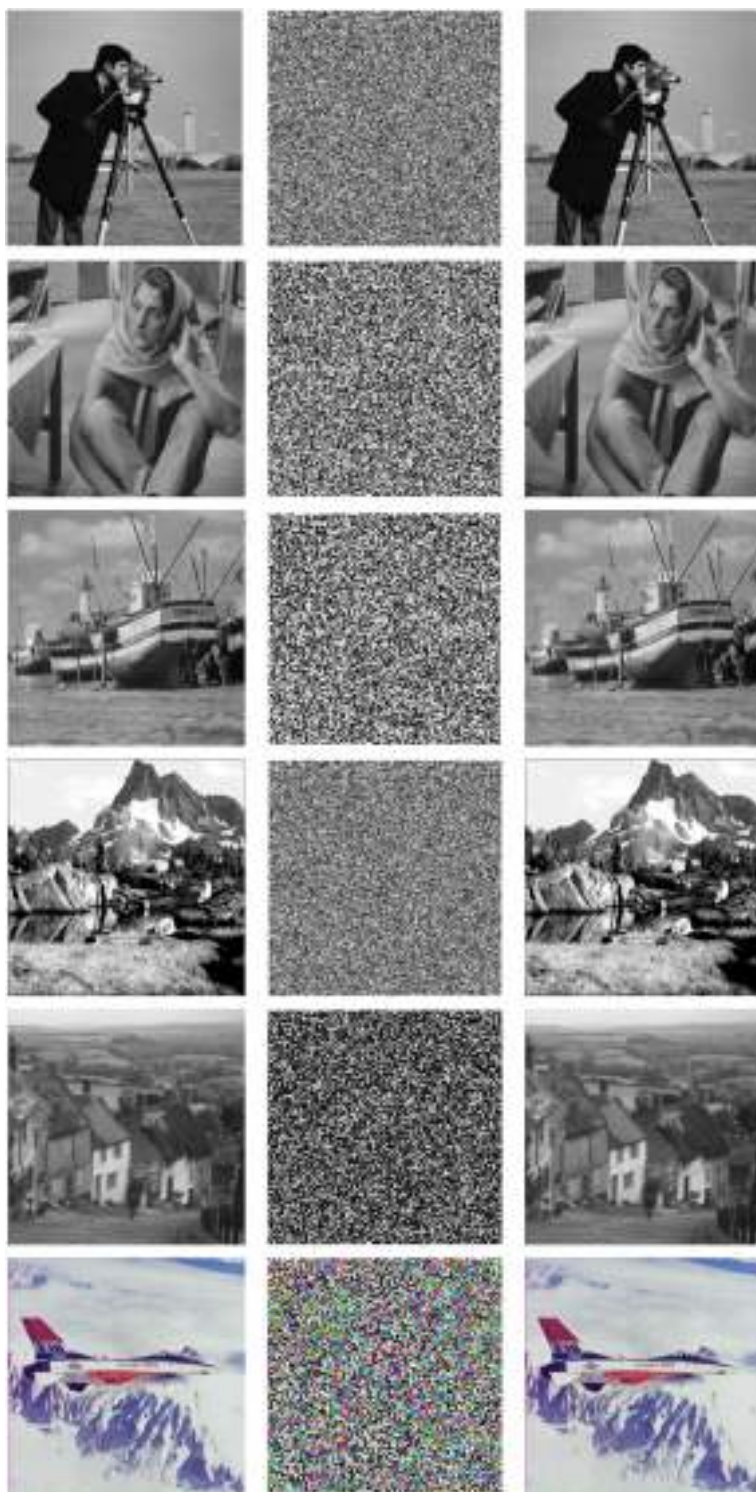
the encrypted images that corresponds to the original images Fig. 6a. It can be observed that the obtained encrypted images possess noisy patterns and hard to understand. No part of the actual images is getting revealed to the human eye. The decrypted images are given in the third column i.e., Fig. 6c and it can be observed that the decrypted images are looking similar to the actual images that are given in Fig. 6a. In fact, the proposed approach is completely lossless and it is possible to recover the actual image completely from the encrypted image.

From Fig. 6, a generic idea about the performance of the proposed image encryption system can be obtained. The visual investigation is necessary but not sufficient always to understand the performance of the encryption algorithm. To understand and interpret the performance of the proposed system, some standard quantitative metrics are used. In this section, a detailed and comprehensive analysis is presented using both qualitative and quantitative procedures. This analysis is helpful to understand the performance of the proposed system in real-life scenarios and against different types of attack.

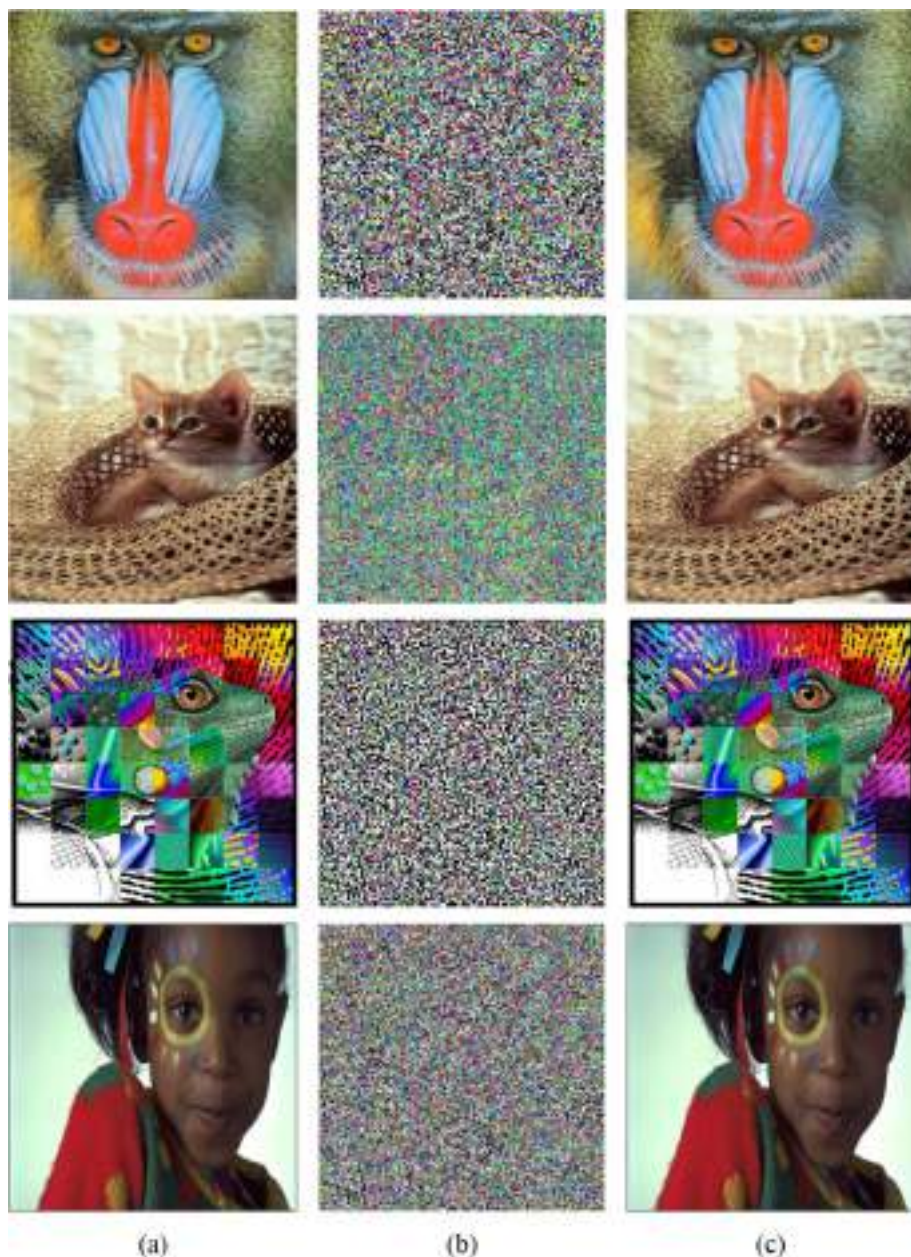
#### 4.1 Keyspace analysis

The keyspace is nothing but the collection of all possible combinations of the key that can be used to perform encryption and decryption. This set is very sensitive in the context of security. If some attacker can access the key or the set of keys then the total security system will get compromised. So, keys are to be protected from the attackers, and to do so, it is necessary to analyze the keyspace of a security system. One of the frequently used and fundamental approaches that are used by the attacker to get the actual combination of the keys is to check all possible combinations of keys. Attackers use all generated combinations to decrypt the encrypted message and to generate possible combinations and to test these combinations one by one, it is necessary to use some automated systems [1, 29, 39]. These types of attacks are known as brute-force attacks. From this discussion, it can be observed that if the keyspace is small then the overall encryption system is highly vulnerable to brute-force attacks because it will be easier to search the overall keyspace. It is essential to have a large keyspace to resist brute-force attacks. In this subsection, the keyspace of the proposed approach is investigated in detail. The overall keyspace of the proposed approach has a dependency on the following factors:

1. Possible combinations of the randomly generated 128-bit key  $k_c$ .
2. The initial parameters  $(s_0, t_0, r)$  of the chaotic logistic map to generate a 128-bit pseudorandom bit sequence  $k_p$ .
3. The initial parameters  $(s_0, t_0, r)$  of the chaotic logistic map to generate a  $\tau$ -bit pseudorandom bit sequence  $p_s$ .
4. The initial parameter  $(s_0)$  of the chaotic logistic map to get the  $cn_A^{th}$  element of the chaotic sequence for the scrambling purposes.
5. The initial parameter  $(r)$  of the chaotic logistic map to generate a chaotic sequence  $c = \{c_1, c_2, \dots, c_m\}$  for the scrambling purposes.



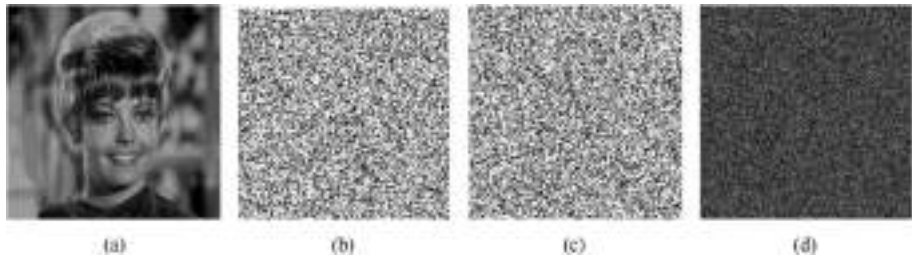




**Fig. 6** (continued)

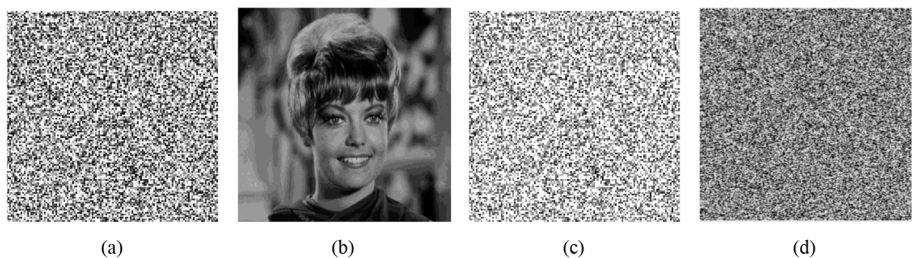
The keyspace of the proposed encryption system is dependent on the aforementioned 5 factors. A detailed analysis of these five factors is essential to enlighten the overall key-space also to test the resistance of the proposed encryption framework against brute-force attacks. Before proceeding further, our basic assumption is that the proposed approach is processing an image of size  $d_1 \times d_2$ .





**Fig. 7** Illustration of the key sensitivity on the encryption process. **a** Original Zelda image, **b** Encrypted image with  $s_0=0.55$ ,  $t_0=0.8$ ,  $r=3.98$ , **c** Encrypted image with  $s_0=0.5$ ,  $t_0=0.7$ ,  $r=3.98$ ; **d** difference between (b) and (c)

The first major factor that is affecting the keyspace is the possible combinations of the randomly generated 128-bit key  $k_c$ . If some intruder wants to determine this key by checking all possible combinations, then, certainly  $2^{128}$  number of combinations are required to be tested. The second important factor is the combinations of the 128-bit pseudorandom bit sequence  $k_p$ . Now, this sequence is generated with the help of the chaotic logistic map and dependent on the initial parameters  $(s_0, t_0, r)$ . These three controlling parameters are assigned floating point values hence, theoretically infinite number of values are possible. So, let us just focus on the objective of using these parameters. In this case, the main objective to use the chaotic system is to generate the 128-bit pseudorandom bit sequence  $k_p$ . If some intruder wants to determine this pseudorandom bit sequence by exhaustive searching, then certainly  $2^{128}$  number of combinations are required to be evaluated. Similarly, to generate a  $\tau$ -bit pseudorandom bit sequence, there is no need to depend on the initial parameters of the pseudorandom bit sequence. Instead of that, intruder will test  $2^r$  number of combinations. According to the proposed scrambling process, the  $cnt_A^{th}$  element of the chaotic sequence will be required to generate a chaotic sequence  $c = \{c_1, c_2, \dots, c_m\}$  for the scrambling purposes which is again used to rearrange the one-dimensional vector  $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$  where,  $m = d_1 \times d_2 \times b$ ,  $d_1 \times d_2$  is the dimension of the actual image, and  $b$  is the number of bitplanes. So, from this discussion, it can be observed that the ultimately a permutation of the vector  $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$  will be obtained. Hence, an attacker may try to generate all the permutations and then check each of them. To do so, the attacker may need to perform  $m!$  number of tests in worst case.



**Fig. 8** Illustration of the key sensitivity on the encryption process. **a** Encrypted image with  $s_0=0.55$ ,  $t_0=0.8$ ,  $r=3.98$ , **b** Decrypted image with  $s_0=0.55$ ,  $t_0=0.8$ ,  $r=3.98$ , **c** Decrypted image with  $s_0=0.5$ ,  $t_0=0.7$ ,  $r=3.98$ ; **d** difference between (b) and (c)

Now summarizing all the situations, the overall complexity  $Z$  of the proposed image encryption is  $Z = 2^{128} \times 2^{128} \times 2^r \times m!$ . For the sake of example, let us consider the proposed image encryption is applied on an image of size  $128 \times 128$  i.e.,  $d_1 = 128$ , and  $d_2 = 128$ . So, the overall keyspace turned out to be  $Z = 2^{128} \times 2^{128} \times 2^8 \times (128 \times 128 \times 4)! \approx 1.53 \times 10^{287273}$  that is significantly large to resist the brute-force attacks.

## 4.2 Analysis of key sensitivity

Key sensitivity refers to the effect of a tiny change in the key on the encrypted outcome. A small change in the key must produce a significantly different outcome and it is an essential property for the secured image encryption systems. In absence of this property, there is always a chance that a tiny change in the keys reveals some portion of the actual information that can be helpful for the cryptanalysis [17, 32]. Hence, it is essential to test the sensitivity of the key or the set of keys to evaluate an image encryption approach [30, 33]. In this subsection, the sensitivity of the key is tested on both encryption and decryption processes and the experimental outcomes are depicted in Figs. 7 and 8 respectively.

To test the key sensitivity, the initial parameters  $s_0, t_0, r$  of the chaotic logistic map that is used to generate a 128-bit pseudorandom bit sequence  $k_p$  are modified slightly and applied for both encryption and decryption purposes. The Zelda image is used for this experiment purposes and depicted in Fig. 7a. Figure 7b and c depicts the encrypted outcome corresponding to the keys (initial parameters)  $s_0=0.55, t_0=0.8, r=3.98, s_0=0.5, t_0=0.7, r=3.98$  respectively. It can be observed that this experiment is carried out with a small change in the key values. The differences between Fig. 7b and c are depicted in Fig. 7d. From Fig. 7d, it can be observed that the difference is significant. Therefore, from this experiment, it can be concluded that a small change in the key brings a significant difference in the encrypted outcome.

The same experiment is carried out on the decryption process. The original Zelda image (Fig. 7a) is encrypted with  $s_0=0.55, t_0=0.8, r=3.98$  and the encrypted outcome is depicted in Fig. 8a. Now, this encrypted image is decrypted with the key values  $s_0=0.55, t_0=0.8, r=3.98$  (i.e., the same key values using which the original image was encrypted) and the result is depicted in Fig. 8b. It can be observed that the original image is returned by the decryption process without any loss because the proposed approach is completely lossless. The same encrypted image is decrypted with the key values  $s_0=0.5, t_0=0.7, r=3.98$  and the result is depicted in Fig. 8c. The difference between Fig. 8b and c can be visualized in Fig. 8d. From Fig. 8d, the significant difference between Fig. 8b and c can be observed.

The above experiments prove the key sensitivity of the proposed approach on both encryption and decryption approaches. It can be observed that a small change in the key brings a significant difference in the produced outcome. Therefore, cryptanalysis becomes a difficult task.

## 4.3 Histogram analysis

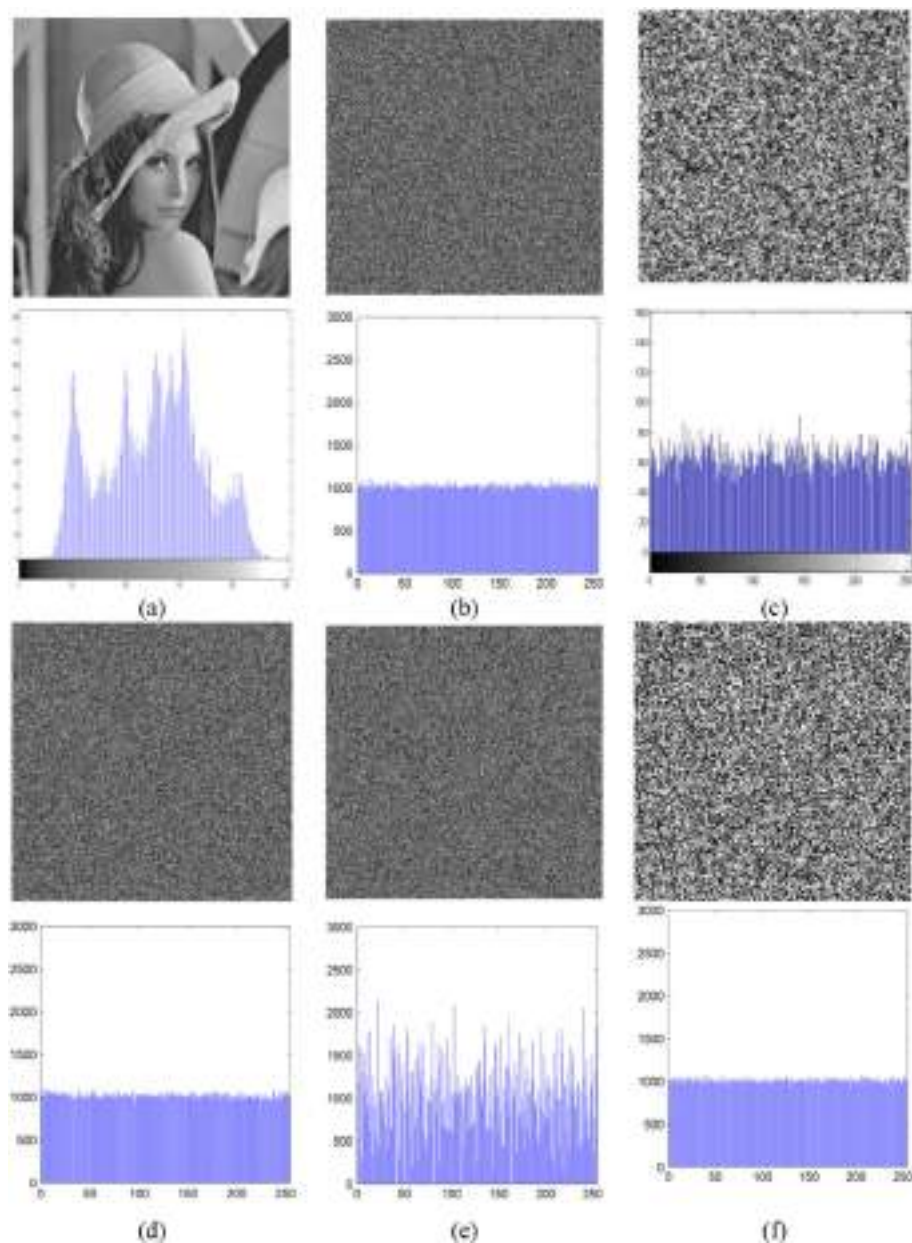
The histogram is the most frequently used tool to study and understand the intensity distribution of an image. It is very popular across the domains and histogram analysis proves to be a successful tool in solving various problems. Typically, most of the visually meaningful images possess non-uniform distribution of the image intensity [21, 37]. In the case of encrypted images, it is necessary to get a uniform or near-uniform distribution of the image

intensity. This property is an essential one to withstand various statistical attacks. Therefore, visual inspection of an encrypted image may provide any hints about the intensity distribution and noise-like images may not always possess uniform or near-uniform intensity distribution. Hence, histogram analysis plays a crucial role to understand the performance of an image encryption system [20, 38].

To establish the efficiency of the proposed approach, four standard and state-of-the-art approaches Zhu's approach [56], Chakraborty's approach [7], Zhou's approach [55], Zhou's approach [54] are considered for the comparison purposes. Figure 9 illustrates the performance comparison for all five approaches (including the proposed one). The well-known Lena image is used for demonstration purposes. The actual Lena image and its histogram are depicted in Fig. 9a. Five encrypted images and their histograms are depicted in Figs. 9b to 9f. The encrypted image that is obtained after applying the proposed approach and its corresponding histogram is depicted in Fig. 9f. From this figure, it can be observed that the proposed approach generates a noise like an encrypted image. Moreover, the histogram of the encrypted image shows that the intensity distribution in the encrypted image is near uniform. This histogram is comparable with the histograms encrypted images that are produced by applying Zhu's approach [56], and the Zhou's approach [55] and depicted in Fig. 9b and d respectively because, the two approaches also produce encrypted images with near-uniform intensity distribution. Figure 9c and e depict the encrypted image and their corresponding histograms that are obtained after applying Chakraborty's algorithm [7], and Zhou's algorithm [54] respectively. From Fig. 9c and e, it can be observed that the encrypted outcome does not contain the uniform intensity distribution. This experiment proves that the proposed approach generates a good quality encrypted image in terms of intensity distribution.

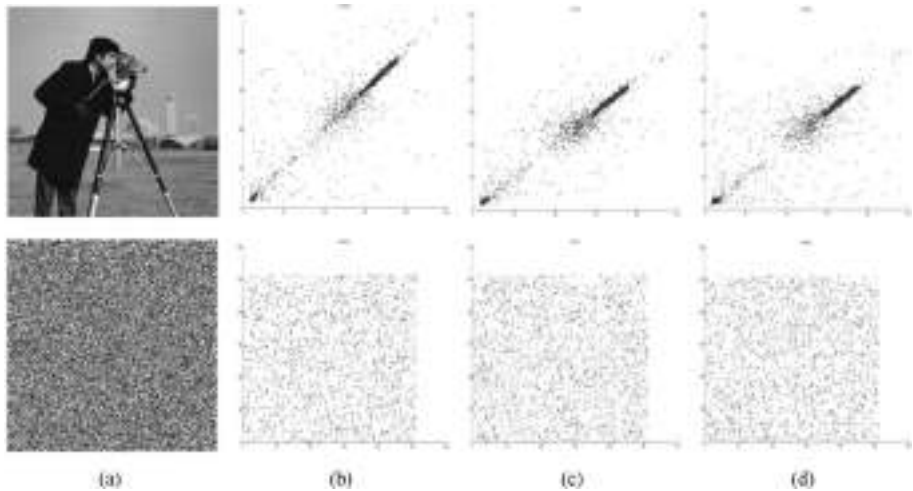
#### 4.4 Pixel correlation analysis

Typically, most meaningful images possess high pixel correlation and it is essential to remove the pixel correlation to produce secured encrypted images. Therefore, a good image encryption scheme must break the high correlation of the actual images. It is essential to prevent unwanted data leakage. Cryptanalysts can attempt to recover actual information or some part of the actual information by analyzing the pixel correlation. Moreover, pixel correlation is also helpful to predict the key or the set of keys. To assess the strength of an image encryption framework, analysis of the correlation coefficient plays a crucial role. To test the correlation coefficient, an experiment is conducted with 2400 randomly selected points and the pair-wise correlations are investigated in both qualitative and quantitative manner in three different directions viz. horizontal, vertical, and diagonal. To visualize the correlation coefficients, the pairwise pixel intensity values are plotted on the x-axis and y-axis. So, if both pixels in the pair of pixels contain the same value, then it will be plotted on the imaginary diagonal line of the plot. The graphical demonstration is given in Fig. 10 and the cameraman standard image is considered for this experiment. From the above discussion, it can be understood that a meaningful image will produce a significantly dense diagonal plot [19]. This fact can also be verified from the top row of Fig. 10. The correlation of the original cameraman image in different dimensions crates highly dense plot around the diagonal region and most of the meaningful images behaves similarly [3]. High correlation among pixels is considered a significant threat to information security and therefore, a good encryption approach must break this and incorporate some randomness in the pixel correlation. An ideal image encryption



**Fig. 9** **a** The original Lena test image and its histogram, **(b)–(f)** encrypted images and their histograms that are obtained after applying the **b** Zhu's approach [56], **c** Chakraborty's approach [7] **d** Zhou's approach [55] **e** Zhou's approach [54], and **f** Proposed approach

procedure should generate an encrypted image with correlation plots that cover the entire domain of the gray-level intensity. Although the result of this experiment is graphically demonstrated in Fig. 10, the graphical explanation is not always sufficient to understand



**Fig. 10** Graphical demonstration of the pair-wise pixel correlation (a) the Cameraman test image and the corresponding encrypted image, plot of the correlation coefficient in the (b) horizontal, (c) vertical, and (d) diagonal directions, respectively

the result of an experiment, and therefore the quantitative analysis is also presented that will be helpful to understand the correlation coefficient numerically. Equation (16) helps to determine the value of the correlation coefficient.

$$CC(x, y) = COV(x, y) / \sqrt{S(x)} \sqrt{S(y)} \quad (16)$$

Here, the pair of intensity values for a pair of adjacent pixels is denoted by  $(x, y)$ . Equations 17 and 18 mathematically express the  $COV(x, y)$  and  $S(z)$  respectively where *PairCnt* denotes total number of pixel pairs under consideration.

$$COV(x, y) = \frac{1}{PairCnt} \sum_{i=1}^{PairCnt} ((x_i - M(x))(y_i - M(y))) \quad (17)$$









$$S(z) = \frac{1}{PairCnt} \sum_{i=1}^{PairCnt} (z_i - M(z_i))^2 \quad (18)$$

Equation 19 gives the mathematical definition of  $M(t)$ .

$$M(t) = \frac{1}{PairCnt} \sum_{i=1}^{PairCnt} t_i \quad (19)$$

The value of the correlation coefficient lies in the range of  $[-1, +1]$ . An absolute value of the correlation coefficient 1 indicates a high correlation [52]. Hence, for a good encryption algorithm, the absolute value of the correlation coefficient should be near 0. The quantitative outcome of the correlation coefficients is reported in Table 4. From this table, it can be observed that the proposed approach generates encrypted images with significantly low pixel correlation that proves the efficiency of the proposed approach. This fact can be

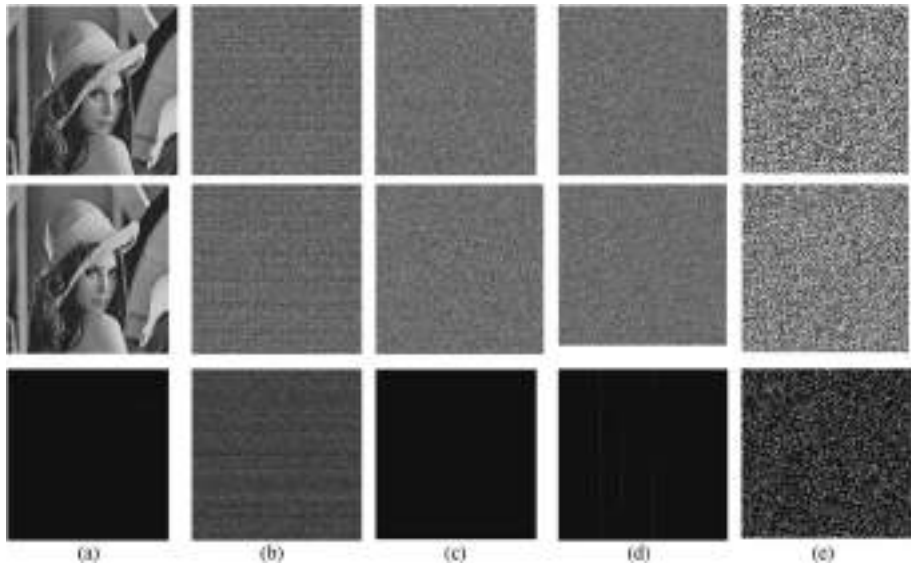
**Table 4** Numerical results of the correlation coefficients for different test images

Image		Original Image			Encrypted Image		
		Vertical	Horizontal	Diagonal	Vertical	Horizontal	Diagonal
Camerman		0.952029213	0.948568075	0.871644035	0.00083567	0.000152979	0.00055243
Cat		0.975442112	0.975010093	0.886554001	0.001222081	-0.000829422	-0.000093550
Owl		0.805903538	0.772681198	0.821597194	0.0020297	-0.000871008	0.001385017
Bridge		0.881123459	0.889081319	0.765126515	-0.00049337	-0.000165289	-0.000360686
Butterfly		0.9478914	0.901313667	0.941885606	0.000631111	0.000268456	0.006789365
Pepper		0.979014786	0.966086679	0.920172381	0.000401151	-0.000637931	-0.001296633
Lena		0.977361884	0.889050899	0.861804563	-0.00686245	-0.000236823	0.001194614
Trees		0.933302803	0.955888887	0.93562286	-0.00053926	0.001765848	0.001460726

crosschecked from Fig. 10. From Fig. 10, it can be observed that the correlation plot almost covers the entire range of the gray level intensity.

#### 4.5 Analysis of the robustness against differential attacks

The attackers may try to get some hint about the key using a method called differential attack. Attackers change some small amount of information in the actual image and then produce the corresponding encrypted image by applying the same algorithm. Now comparing the encrypted images (i.e., one generated from the original image and another one that is generated from the slightly modified original image) attacker tries to analyze and explore some patterns that may help to reveal the actual key or the set of keys. Therefore, it is considered a severe security threat and an efficient image encryption approach must resist such kind of attacks. To prevent such kinds of attacks, it is essential to generate a significantly different encrypted image with a small change in the input image [26, 42]. This subsection is dedicated to analyzing the performance of the proposed approach against such differential attacks. Both qualitative and quantitative measures are considered for this experiment. The graphical illustration is provided in Fig. 11 where the experiment is carried out by applying the proposed approach on the Lena standard image. To conduct this experiment and to compare the performance of the proposed approach with some other state-of-the-art approaches, the original Lena image is used twice with just a single bit difference. In this experiment, the proposed approach is compared with three other state-of-the-art approaches viz. Decom-Crypt [55], Zhu's approach [56], and Zhou's approach [54]. From the graphical results that are presented in Fig. 11, it can be observed that Zhu's approach and Zhou's approach do not respond well for a single bit difference but, the proposed approach and DecomCrypt perform well and shows a significant change in the encrypted image.



**Fig. 11** Qualitative demonstration of the differential attack (a) from top to bottom: the original Lena test image and the same Lena image with a single bit difference, and the difference between these two images, (b)–(e) encrypted images and their difference that are generated by applying (b) DecomCrypt [55], (c) Zhou's approach [54], (d) Zhu's approach [56], (e) proposed approach

Not only the qualitative analysis but the quantitative analysis is also provided to test the resilience of the proposed approach against differential attacks. Two well-known and frequently used parameters i.e., the Number of Pixel Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) are used for this purpose. Equations 20 and 21 mathematically express these two parameters respectively.

$$NPCR = \frac{\sum_{x=1}^{d_1} \sum_{y=1}^{d_2} M(x, y)}{d_1 \times d_2} \times 100\% \quad (20)$$

$$UACI = \frac{1}{d_1 \times d_2} \sum_{x=1}^{d_1} \sum_{y=1}^{d_2} \frac{|I_1(x, y) - I_2(x, y)|}{255} \quad (21)$$

In the above equations,  $M(\cdot)$  denotes a 2-D matrix that contains some binary members indicating the pixel variation between two images  $I_1$  and  $I_2$  of dimension  $d_1 \times d_2$  and it is mathematically expressed using Eq. 30.

$$M(x, y) = \begin{cases} 1 & \text{if } I_1(x, y) \neq I_2(x, y) \\ 0 & \text{Otherwise} \end{cases} \quad (22)$$

All four approaches are applied on the ten different images that are selected from a web repository and the experimental outcomes along with the image IDs are given in Table 5. These comparative quantitative outcomes show that the proposed approach works well for most of the images and on average proposed approach outperforms almost all other approaches.



Table 5 Numerical outcomes: NPCR and UACI values

Image Id	NPCR			UACI				
	DecomCrypt [55]	Zhou's approach [54]	Zhu's approach [56]	Proposed approach	DecomCrypt [55]	Zhou's approach [54]	Zhu's approach [56]	Proposed approach
1	0.9959	0.00000382	0.0048	0.999451689	0.3454	0.000000314	0.0016	0.316745214
2	0.9959	0.00000372	0.0012	0.995201963	0.3342	0.000002244	0.0004	0.321135618
3	0.9962	0.00000379	0.0024	0.992678365	0.3363	0.000000254	0.0008	0.335448981
7	0.9962	0.00000374	0.0004	0.991125243	0.3341	0.000001586	0.0001	0.331051104
12	0.7531	0.00000369	0.0059	0.829171026	0.2661	0.000000793	0.0020	0.297186362
14	0.9374	0.00000382	0.0053	0.906210528	0.2864	0.000002872	0.0018	0.311268019
15	0.9963	0.00000380	0.0010	0.950376641	0.3456	0.000005595	0.0004	0.325320851
20	0.9961	0.00000377	0.0000	0.997467039	0.3699	0.000003471	0.0000	0.33657154
25	0.9963	0.00000382	0.0070	0.988904282	0.3348	0.000005984	0.0023	0.331275317
35	0.9960	0.00000376	0.0042	0.981905969	0.3358	0.000003560	0.0014	0.305584516
Average	0.96594	0.00000377	0.00322	0.963249274	0.32886	0.00000267	0.00108	0.321158752



From Table 5 it can be observed that, on average, the proposed approach achieves 96.32% of NPCR value and 32.11% of UACI value. Moreover, the proposed approach outperforms the other three approaches in terms of NPCR percentage but the DecomCrypt approach outperforms the proposed approach in terms of UACI percentage. However, the proposed approach outperforms the other two approaches in terms of UACI percentage. This experiment proves the strength of the proposed approach against differential attacks.

#### 4.6 Analysis of the time complexity

Time complexity is one of the most important parameters from the algorithmic perspective that is analyzed in this sub-section. The process begins by generating a 128-bit key  $k_c$  and a 128-bit pseudorandom bit sequence  $k_p$ . This can be done in constant amount of time because the length of both  $k_c$  and  $k_p$  are fixed. The time complexity of pseudorandom bit generation depends on the specific algorithm or method used for generating the random bits. Here, it is dependent on the length of the key. The length of the key is considered as a fixed value (128 bits) irrespective of the input size. Therefore, it can be considered that this operation can be performed in constant amount of time. Hence the bitwise XOR operation  $k_x \leftarrow k_c \oplus k_p$  can also be performed in constant time. Because, the time complexity of the bitwise XOR operation depends on the size of the operands, specifically the number of bits in the binary representation of the numbers being XORed. Since the XOR operation for a pair of bits can be computed in constant time, the overall time complexity of the bitwise XOR operation is typically considered to be  $O(n)$ , where  $n$  is the number of bits in the operands. This is because the XOR operation needs to perform  $n$  individual bit comparisons. Therefore, as the size of the operands increases, the time required to perform the XOR operation also increases linearly. However, it's important to note that for most practical applications, the actual time taken to perform a bitwise XOR operation on typical modern hardware is very fast and can be considered constant time in practice, regardless of the number of bits. Moreover, here the number of bits is constant (i.e., 128 bits) and not dependent on the input size. So, we can consider it as a constant time. The value of  $\varphi$  can be calculated and the number of nodes  $N$  can also be calculated in constant amount of time. A pseudorandom bit sequence  $ps$  is generated using Eq. (5) of length  $\tau$ . So, this step can be computed with  $O(\tau)$  time. The EMO algorithm operates in an iterative manner, where each iteration consists of several steps, such as the evaluation of the objective function, updating the particle positions, and adjusting the particle charges. Now the EMO approach involves the local search process that can be implemented with a time complexity of  $O(n^2 \cdot k)$ . Next, the force computation phase can be implemented in a  $O(n^2 \cdot d)$  time. Now, the movement of the particles can be implemented in  $O(n)$  time i.e., in linear time. Hence, the overall time complexity involved in the EMO process is  $O(n^2 \cdot k + n^2 \cdot d + n)$  i.e.,  $O(n^2 \cdot (k + d) + n)$  where  $n$  denotes the number of particles i.e., member of the population,  $k$  represents the iteration count, and  $d$  denotes the dimension of each particle [13]. The convergence behavior of EMO varies depending on the problem and the implementation. Some problems may require more iterations to converge, while others may converge faster. The number of iterations is typically determined by a termination condition, such as reaching a maximum number of iterations or achieving a desired level of convergence. The biological XOR operation can be performed in  $O(d_1 \times d_2)$  time because every pixel needs to be visited. The proposed scrambling approach incurs a time complexity  $O(d_1 \times d_2 \times b)$  because a three-dimensional array needs to be converted into a one-dimensional array. Here, different operations are performed on the converted one-dimensional array but they are performed separately. So,

in different steps, maximum  $d_1 \times d_2 \times b$  number of operations are involved and therefore, it will dominate. The overall complexity can be computed as follows  $O(\tau + n^2 \cdot (k + d) + n + (d_1 \times d_2 \times b))$  and from this analysis, it can be observed that the proposed approach incurs a polynomial time complexity.

#### 4.7 Analysis of the space complexity

Let us assume that the size of the input image is  $d_1 \times d_2$ . To store and process this image we need  $d_1 \times d_2$  amount of memory. Now to store,  $k_c$ ,  $k_p$ , and  $k_x$ , needs constant amount of space (because all are 128 bits) and it is not dependent on the size of the input image. The space complexity of Electromagnetism-like Optimization (EMO) depends on the number of variables, the population size, and the data structures used to represent the particles and their attributes. It's important to note that the space complexity can vary depending on the specific implementation details and the problem being solved. So, if there are  $n$  number of populations and the population is comprising of solutions of length  $l$  then, we need extra  $(n \times l)$  amount of space. So, the overall dominating factors that contribute to the space complexity is  $d_1 \times d_2$  and  $(n \times l)$ . Therefore, it can be declared that the proposed approach can be executed with a space complexity of  $O(d_1 \times d_2 + n \times l)$ .

#### 4.8 Analysis of the information entropy

Information entropy is another important parameter from the perspective of image encryption and data security that needs to be analyzed to understand the strength and performance of an image encryption approach. The value of the information entropy for an encrypted image is expected to be near 8 [49]. Hence, an encryption framework is expected to produce encrypted images with information entropy values close to 8. The mathematical expression for information entropy is given in Eq. 23.

$$IEntropy(s) = \sum_{i=0}^{2^l-1} P(s_i) \log \frac{1}{P(s_i)} \quad (23)$$

In this equation,  $l$  denotes the number of bits present in a symbol  $s_i$  where,  $s_i \in s$ .  $P(s_i)$  represents the probability of occurrence of the  $i^{th}$  symbol  $s_i$ . The numerical results that are obtained after applying the proposed approach on some standard images are reported in Table 6 and a comparative study is presented in Table 6. The proposed approach is compared with 7 standard approaches and it can be observed that the proposed approach performs well in comparison with the other approaches. The Lena standard image is considered to perform this experiment and once again, these experiments help to establish the efficiency and superiority of the proposed approach.

**Table 6** Numerical results (i.e., the values of the information entropy), that are obtained after applying the proposed approach on seven standard test images (Table 7)

Image	Camerman	Cat	Owl	Bridge	Butterfly	Pepper	Trees
Information Entropy	7.9723563	7.9955026	7.99100659	7.98012236	7.99003260	7.98550673	7.98001973

**Table 7** Comparative study of the information entropy (Lena standard test image is considered for this experiment)

Method	[49]	[18]	[48]	[47] (upto 8 rounds)	[46]	[53]	[51]	Proposed approach
Information Entropy	7.9993	7.9965	7.9909	7.9972	7.9951	7.9992	7.9991	7.9991

## 5 Conclusion

In this work, a novel method for image encryption is proposed that is based on the BST, DNA encoding, and EMO approach. The proposed approach is applied to different types of images and investigated in both qualitative and quantitative manner. A detailed and rigorous investigation helps to explore the potentials of the proposed image encryption system to become suitable for various real-life applications. The experimental outcomes show that the proposed approach can effectively resist different types of attacks and produce secured encrypted images. Significantly large keyspace makes this approach reliable and resilient against brute-force attacks. Besides proposing a novel image encryption scheme, a scrambling approach is also proposed. However, the proposed image encryption approach is flexible enough to any other scrambling approach that serves the desired purpose. Although the proposed approach is highly sensitive to the key but, the scrambling algorithm provides an additional layer of security. The proposed approach is symmetric and lossless in nature hence, the proposed approach does not incur any data loss. The comparative study shows some encouraging results and also enlightens the prospect of the proposed approach as an effective tool to secure real-life digital image communications.

**Data availability** Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

### Declarations

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.

## References

1. Abdulla AA (2015) Exploiting similarities between secret and cover images for improved embedding efficiency and security in digital steganography (Doctoral dissertation, University of Buckingham)
2. Amari SI (1993) Backpropagation and stochastic gradient descent method. *Neurocomputing* 5(4–5):185–196
3. Bao L, Zhou Y (2015) Image encryption: Generating visually meaningful encrypted images. *Inf Sci* 324:197–207
4. Birbil ŞI, Fang SC (2003) An electromagnetism-like mechanism for global optimization. *J Glob Optim* 25:263–282. <https://doi.org/10.1023/A:1022452626305>
5. Chakraborty S, Mali K (2020) SuFMoFPA: a superpixel and meta-heuristic based fuzzy image segmentation approach to explicate COVID-19 radiological images. *Expert Syst Appl* 114142. <https://doi.org/10.1016/j.eswa.2020.114142>
6. Chakraborty S, Mali K (2020) Fuzzy electromagnetism optimization (FEMO) and its application in biomedical image segmentation. *Appl Soft Comput J* 97. <https://doi.org/10.1016/j.asoc.2020.106800>

7. Chakraborty S, Seal A, Roy M, Mali K (2016) A novel lossless image encryption method using DNA substitution and chaotic logistic map. *Int J Secur its Appl* 10: <https://doi.org/10.14257/ijisa.2016.10.2.19>
8. Chakraborty S, Chatterjee S, Dey N, Ashour AS, Ashour AS, Shi F, Mali K (2017) Modified cuckoo search algorithm in microscopic image segmentation of hippocampus. *Microscopy Research and Technique* 80(10):1051–1072
9. Computer security division N FIPS 46-3, data encryption standard (DES) (withdrawn May 19, 2005)
10. Cui G, Qin L, Wang Y, Zhang X (2008) An encryption scheme using DNA technology. In: 2008 3rd international conference on bio-inspired computing: theories and applications. IEEE, pp 37–42
11. Daemen J, Rijmen V (1998) The block cipher Rijndael. In *International Conference on Smart Card Research and Advanced Applications*. Berlin, Springer pp. 277–284
12. Enayatifar R, Abdullah AH, Lee M (2013) A weighted discrete imperialist competitive algorithm (WDICA) combined with chaotic map for image encryption. *Opt Lasers Eng* 51:1066–1077. <https://doi.org/10.1016/j.optlaseng.2013.03.010>
13. Gálvez J, Cuevas E, Avalos O, Oliva D, Hinojosa S (2018) Electromagnetism-like mechanism with collective animal behavior for multimodal optimization. *Appl Intell* 48:2580–2612
14. Gehani A, LaBean T, Reif J (2004) DNA-based cryptography. *Aspects of molecular computing: essays dedicated to tom head, on the occasion of his 70th birthday*, 167–188
15. Gehani A, La Bean T, Reif JH DNA-based cryptography. *DIMACS series in discrete mathematics*. *Theor Comput Sci* 54:233–249
16. Grangotto M, Magli E, Olmo G (2006) Multimedia selective encryption by means of randomized arithmetic coding. *IEEE Trans Multimed* 8:905–917. <https://doi.org/10.1109/TMM.2006.879919>
17. Hu J, Han F (2009) A pixel-based scrambling scheme for digital medical images protection. *J Netw Comput Appl* 32:788–794. <https://doi.org/10.1016/J.JNCA.2009.02.009>
18. Hua Z, Zhou Y, Pun C-M, Chen CLP (2015) 2D sine logistic modulation map for image encryption. *Inf Sci (Ny)* 297:80–94. <https://doi.org/10.1016/J.INS.2014.11.018>
19. Hua Z, Zhou Y, Huang H (2019) Cosine-transform-based chaotic system for image encryption. *Inf Sci (Ny)* 480:403–419. <https://doi.org/10.1016/J.INS.2018.12.048>
20. Kamali SH, Hedayati M, Shakerian R, Rahmani M (2010) A new modified version of advanced encryption standard based algorithm for image encryption. In: *ICEIE 2010 - 2010 International Conference on Electronics and Information Engineering*, Proceedings
21. Khan HN, Chaudhuri A, Das A, Chaudhuri A (2019) An ultra robust session key based image cryptography. *Microsyst Technol* 1–9. <https://doi.org/10.1007/s00542-019-04518-9>
22. Leong MP, Cheung OYH, Tsoi KH, Leong PHW A bit-serial implementation of the international data encryption algorithm IDEA. In: *Proceedings 2000 IEEE Symposium on Field-Programmable Custom Computing Machines (Cat. No. PR00871)*. IEEE Comput Soc, pp 122–131
23. Li L, Xie Y, Liu B et al (2019) Optical image encryption and transmission with semiconductor lasers. *Opt Laser Technol* 119:105616. <https://doi.org/10.1016/j.optlastec.2019.105616>
24. Lindholm FA, Fossum JG, Burgess EL (1979) Application of the superposition principle to solar-cell analysis. *IEEE Trans Electron Devices* 26:165–171. <https://doi.org/10.1109/T-ED.1979.19400>
25. Mali K, Chakraborty S, Seal A, Roy M (2015) An efficient image cryptographic algorithm based on frequency domain using Haar wavelet transform. *Int J Secur its Appl* 9:279–288. <https://doi.org/10.14257/ijisa.2015.9.12.26>
26. Mali K, Chakraborty S, Roy M (2015) A study on statistical analysis and security evaluation parameters in image encryption. *IJSRD-International J Sci Res Dev* 3:2321–0613
27. Pareek N, Patidar V, Sud K (2003) Discrete chaotic cryptography using external key. *Phys Lett A* 309:75–82. [https://doi.org/10.1016/S0375-9601\(03\)00122-1](https://doi.org/10.1016/S0375-9601(03)00122-1)
28. Patidar V, Sud K, Informatica NP, (2009) Undefined A pseudo random bit generator based on chaotic logistic map and its statistical testing. *informatica.si*
29. Roy M, Mali K, Chatterjee S, et al (2019) A study on the applications of the biomedical image encryption methods for secured computer aided diagnostics. In: *proceedings - 2019 Amity International conference on artificial intelligence, AICAI 2019*
30. Roy M, Chakraborty S, Mali K, et al (2019) A dual layer image encryption using polymerase chain reaction amplification and dna encryption. In *2019 international conference on optoelectronics and applied optics (Optronix)*. IEEE (pp. 1–4)
31. Roy M, Chakraborty S, Mali K (2020) A robust image encryption method using chaotic skew-tent map. In: Chakraborty S, Mali K (eds) *Applications of Advanced Machine Intelligence in Computer Vision and Object Recognition: Emerging Research and Opportunities*
32. Roy M, Chakraborty S, Mali K et al (2020) Biomedical image security using matrix manipulation and DNA encryption. In: *Advances in intelligent systems and computing*. Springer, pp 49–60

33. Roy M, Chakraborty S, Mali K, et al (2020) Data security techniques based on DNA encryption. In Proceedings of International Ethical Hacking Conference 2019: eHaCON 2019, Kolkata, India. Springer, Singapore (pp. 239–249)
34. Roy M, Chakraborty S, Mali K (2021) A chaotic framework and its application in image encryption. *Multimedia Tools Appl* 80:24069–24110
35. Roy M, Chakraborty S, Mali K (2021) The MSK: a simple and robust image encryption method. *Multimed Tools Appl* 80:21261–21291
36. Roy M, Chakraborty S, Mali K, et al (2021) A robust image encryption framework based on DNA computing and chaotic environment. *Microsyst Technol* 1–11. <https://doi.org/10.1007/s00542-020-05120-0>
37. Roy M, Chakraborty S, Mali K et al (2021) An image security method based on low dimensional chaotic environment and DNA encoding. Springer, Singapore, pp 267–277
38. Roy M, Chakraborty S, Mali K, Roy D (2021) Utilization of Hyperchaotic environment and DNA sequences for digital image security. *Advances in smart communication technology and information processing: OPTRONIX 2020*. Springer, Singapore
39. Seal A, Chakraborty S, Mali K (2017) A new and resilient image encryption technique based on pixel manipulation, value transformation and visual transformation utilizing single-level haar wavelet transform
40. Shujun L, Xuanqin M, Yuanlong C (2001) Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography. Springer, Berlin, Heidelberg, pp 316–329
41. Sun F, Lu Z, Liu S (2010) A new cryptosystem based on spatial chaotic system. *Opt Commun* 283:2066–2073. <https://doi.org/10.1016/j.optcom.2010.01.028>
42. Suri S, Vijay R (2019) A bi-objective genetic algorithm optimization of Chaos-DNA based hybrid approach. *J Intell Syst* 28:333–346. <https://doi.org/10.1515/jisys-2017-0069>
43. Tsamardinos I, Brown LE, Aliferis CF (2006) The max-min hill-climbing Bayesian network structure learning algorithm. *Mach Learn* 65:31–78. <https://doi.org/10.1007/s10994-006-6889-7>
44. Wadi SM, Zainal N (2014) High definition image encryption algorithm based on AES modification. *Wirel Pers Commun* 79:811–829. <https://doi.org/10.1007/s11277-014-1888-7>
45. Wang X, Zhang Q (2009) DNA computing-based cryptography. In: 2009 fourth international on conference on bio-inspired computing. IEEE, pp 1–3
46. Wang X, Liu L, Zhang Y (2015) A novel chaotic block image encryption algorithm based on dynamic random growth technique. *Opt Lasers Eng* 66:10–18. <https://doi.org/10.1016/J.OPTLASENG.2014.08.005>
47. Wu Y, Zhou Y, Noonan JP, Agaian S (2014) Design of image cipher using latin squares. *Inf Sci (Ny)* 264:317–339. <https://doi.org/10.1016/J.INS.2013.11.027>
48. Wu X, Wang D, Kurths J, Kan H (2016) A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system. *Inf Sci (Ny)* 349–350:137–153. <https://doi.org/10.1016/J.INS.2016.02.041>
49. Xu M, Tian Z (2019) A novel image cipher based on 3D bit matrix and latin cubes. *Inf Sci (Ny)* 478:1–14. <https://doi.org/10.1016/J.INS.2018.11.010>
50. Ye R (2011) A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism. *Opt Commun* 284:5290–5298. <https://doi.org/10.1016/j.optcom.2011.07.070>
51. Zahmoul R, Ejbal R, Zaied M (2017) Image encryption based on new Beta chaotic maps. *Opt Lasers Eng* 96:39–49. <https://doi.org/10.1016/J.OPTLASENG.2017.04.009>
52. Zhang Y (2018) The unified image encryption algorithm based on chaos and cubic S-box. *Inf Sci (Ny)* 450:361–377. <https://doi.org/10.1016/J.INS.2018.03.055>
53. Zhang W, Yu H, Zhao Y, Zhu Z (2016) Image encryption based on three-dimensional bit matrix permutation. *Signal Process* 118:36–50. <https://doi.org/10.1016/J.SIGPRO.2015.06.008>
54. Zhou Y, Panetta K, Agaian S, Chen CLP (2013) (n, k, p)-gray code for image systems. *IEEE Trans Cybern* 43:515–529. <https://doi.org/10.1109/TSMCB.2012.2210706>
55. Zhou Y, Cao W, Philip Chen CL (2014) Image encryption using binary bitplane. *Signal Process* 100:197–207. <https://doi.org/10.1016/J.SIGPRO.2014.01.020>
56. Zhu Z, Zhang W, Wong K, Yu H (2011) A chaos-based symmetric image encryption scheme using a bit-level permutation. *Inf Sci (Ny)* 181:1171–1186. <https://doi.org/10.1016/J.INS.2010.11.009>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.