Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Toward robust image encryption based on chaos theory and DNA computing

Usman Asghar [1], Shahzad Yousaf [2], Areej Fatima [3], Muhammad Saleem [4,*], Muhammad Ahsan Raza [5], Taher M. Ghazal [6,7,8]

[1]*Department of Computer Science, National College of Business Administration and Economics, Lahore, Pakistan*
[2]*School of Integrated and Social Sciences (SISS), University of Lahore, Lahore, Pakistan*
[3]*Department of Computer Science, Lahore Garrison University, Lahore, Pakistan*
[4]*School of Computer Science, Minhaj University Lahore, Lahore, Pakistan*
[5]*Department of Information Sciences, University of Education, Lahore, Multan Campus 60000, Pakistan*
[6]*Centre for Cyber Physical Systems, Computer Science Department, Khalifa University, Abu Dhabi, United Arab Emirates*
[7]*Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia*
[8]*Applied Science Research Center, Applied Science Private University, Amman 11937, Jordan*

ARTICLE INFO

Article history:
Received 17 December 2023
Received in revised form
25 April 2024
Accepted 30 May 2024

Keywords:
Image cipher
Pixel swapping
Chaotic map
Encryption techniques
Information entropy

ABSTRACT

Due to the significant importance of image security for various users, there is an ongoing need to develop innovative algorithms to enhance this security. Image security typically involves encryption techniques. This study has tackled the challenge of creating an efficient, secure, and resilient image cipher by using pixel-swapping techniques at both DNA and decimal levels. The swapping methods include four different approaches that involve randomly selecting pixel pairs to swap with adjacent pixels—either left, right, upper, or lower—based on random numbers generated by a chaotic map. Specifically, the 2D Tinkerbell chaotic map was used to generate the necessary random numbers for diffusion and confusion processes in the encryption. Additionally, through careful arithmetic operations, two more random number streams were derived from the main streams produced by the chaotic map. Thorough performance analyses and computer simulations have shown that this image cipher is robust, secure, and resilient against various threats, making it suitable for practical applications. Notably, the cipher achieved a very high information entropy value of 7.9975, indicating its effectiveness in encryption.

1. Introduction

The global landscape has experienced a proliferation of diverse software and hardware products that are utilized across all cultures. Simultaneously, there is a persistent concern regarding security, given the inherent utility of these products. Additionally, digital images have become an integral part of various aspects of human endeavors, encompassing fields such as medicine, transportation, research, space science, art, and culture, among others (Malhotra et al., 2021). These images are routinely transmitted via the Internet from one location to another and are stored on various devices, including hard drives, cloud servers, USB drives, personal digital assistants, and tablets. However, handling these images demands a high level of caution due to their sensitive nature. For example, an image containing the blueprint of a new car designed by a multinational company holds significant strategic value. Therefore, such images must be stored and transmitted with utmost care. Cryptography is typically employed to exercise this caution. It involves converting the plaintext image into an encrypted format, rendering it unreadable to potential adversaries in the event of unauthorized access or hacking (Kolivand et al., 2024).

Traditionally, Data Encryption Standard (DES), Advanced Encryption Standard (AES), and Rivest-Shamir-Adleman (RSA) techniques have been used to achieve this purpose. However, since images have different properties of bulky volume strong inter-pixel bonding, these techniques cannot be applied to the images (Iqbal et al. 2019). Fortunately, chaotic

* Corresponding Author.
Email Address: muhammadsaleem.cs@mul.edu.pk (M. Saleem)
https://doi.org/10.21833/ijaas.2024.06.014
Corresponding author's ORCID profile:
https://orcid.org/0000-0002-5209-8375

systems have proved very useful for this purpose. By using these systems, myriads of image ciphers have been written by the researchers (Wei et al., 2012; Liu et al., 2012; Enayatifar et al., 2014; Zhen et al. 2016; Wu et al., 2018; Chai et al., 2019; Guesmi et al. 2016) to date. Many image ciphers have also been cracked up by cryptanalysts by exploiting the inherent vulnerabilities in their design principles. For instance, the image cipher (Liu et al. 2018) was broken by Ma et al. (2020) due to the unfamiliarity of the important feature of plaintext sensitivity. Besides, some image ciphers are very time-consuming (Xiong et al., 2019; Hanif et al., 2020). This era needs efficient and secured ciphers. The reason for the inefficient ciphers is two-fold. On the one hand, the hyperchaotic systems being employed are very time-consuming. On the other hand, the complicated algorithms are the major reason for the large response time. In this work, we have tried to strike a reasonable balance between these two competing requirements.

Since the DNA (Iqbal et al., 2021a; 2020a) technology inherits the excellent features of massive parallelism, surprising data density, and ultra-low energy consumption (Adleman, 1994; Babaei, 2013), it proved a plausible information processing paradigm for the niche of image encryption. A thorough perusal of the literature indicates that a plethora of image cryptosystems were written by fusion of chaotic maps and DNA encoding (Wei et al., 2012; Liu et al., 2012; Enayatifar et al., 2014; Zhen et al., 2016; Wu et al., 2018; Chai et al., 2019; Guesmi et al., 2016). DNA computing is concerned with DNA subtraction/addition and DNA XOR strategies. DNA encoding is related, including the corresponding directions of strands. DNA-based image ciphers have a few defects in their design principles. For example, the cipher (Liu et al., 2012) was broken by Özkaynak et al. (2013) by using the chosen plaintext attack strategy. Some image encryption schemes did not choose the proper chaotic map. For example, the techniques used by Liu et al. (2012) and Enayatifar et al. (2014) utilized the Logistic map with DNA encoding. However, the chaoticity and randomness of the generated random numbers do not comply with the standards of chaoticity (Zhou et al., 2014). Moreover, some studies have not used dynamic rules for DNA encoding. Rather, they used the fixed rules for the mask and plain images (Guesmi et al., 2016; Babaei, 2013; Zhang et al., 2010; Zhang et al., 2013; Batool et al., 2021). This is an inherent vulnerability that can be exploited by hackers. For instance, the work of Zhang et al. (2013) employed the fourth rule of decoding and the third rule for encoding. The fixed rules for encryption make it easier for hackers to break the ciphers. Therefore, these rules should be adapted based on the specific input image being encrypted. This approach would help prevent chosen-plaintext and known-plaintext attacks.

Inspired by the above discussion, this study proposes an alternative image cipher based on the Tinkerbell chaotic map, pixel swapping from randomly selected pairs of the input image, and DNA encoding. The swapping operations are performed in four modes: each randomly selected pixel is swapped with its right, left, upper, or lower neighbor, depending on the random values generated by the chaotic maps. The paper is organized as follows: Section 2 explains the essential concepts and building blocks, including DNA encoding and the Tinkerbell chaotic map. Section 3 describes the proposed image cipher and the method for generating random data in detail. Section 4 presents computer simulations to demonstrate the feasibility of the proposed image cryptosystem. Section 5 evaluates the security and performance using selected validation metrics. Finally, Section 6 concludes the study and suggests future research directions. Sun et al. (2019) introduced an image encryption method based on block swapping and a 5D hyperchaotic system. They used the image's SHA-256 hash value to determine key sensitivity and generate the chaotic map's pseudorandom numbers, which created a control table for pixel swapping. A cyclic shift process was applied for diffusion, demonstrating robustness against various attacks, including differential, correlation, and entropy attacks. Chai et al. (2018) proposed an image encryption method using a hyperchaotic system, elementary cellular automata, and compressive sensing. They used the SHA-512 hash value for plaintext key sensitivity and the zigzag method for scrambling, providing protection against known and selected plaintext attacks. Their simulation results showed that the method offered security, time efficiency, and resistance to various attacks.

In the context of robust image encryption, integrating Deep Extreme Learning Machine (DELM), Khan et al. (2021), Siddiqui et al. (2022), and Khan et al. (2022) could enhance feature extraction for improved security. Hyperchaotic systems, DNA encoding, and Swarm Optimization offer potential strategies for creating robust encryption algorithms, leveraging chaos-based theories and machine learning techniques to strengthen image protection against unauthorized access (Iqbal et al., 2021a; Khan et al., 2019; Iqbal et al., 2021a).

2. Concepts and building blocks

In this section, the required fostering blocks of the current study will be discussed.

2.1. Chaotic system/maps

Chaos theory studies the behavior of dynamic systems that are highly sensitive to initial conditions and parameters. This multidisciplinary field shows that despite the apparent randomness of chaotic systems, there are deterministic patterns such as self-organization, fractals, and continuous feedback loops. A small change in the initial conditions can significantly affect future states, illustrating the concept of sensitive dependence. The "butterfly effect," proposed by Edward Lorenz, metaphorically describes this idea: "if a butterfly flaps its wings in

Brazil, it can cause a hurricane in Texas." Mathematicians have developed various chaotic systems and maps based on chaos theory. These maps, essentially mathematical functions, generate random numbers and have been successfully used in many image cryptosystems due to their unique properties. There are three types of maps: low-dimensional maps (one or two dimensions), high-dimensional maps (more than two dimensions), and hyperchaotic maps. Low-dimensional maps are simple and generate less chaotic data, while high-dimensional maps are complex but produce a large amount of chaotic data useful for encryption. In image cryptography, chaotic systems are widely used to create encryption algorithms. This study selects the Tinkerbell chaotic map (Iqbal et al., 2020b) because it is a low-dimensional map, making it faster to generate the required random numbers, and it has highly chaotic characteristics. Here is its mathematical equation (Iqbal et al., 2020b).

$$\begin{cases} u_{n+1} = u_n^2 - v_n^2 + au_n + bv_n \\ v_{n+1} = 2u_n v_n + cu_n + dv_n \end{cases} \quad (1)$$

where, n = 0, 1, 2, ... Moreover, u and v are the initial states, and a, b, c, and d are the system parameters of the map. Its random and chaotic behavior can be observed through the attractors drawn in Fig. 1. These attractors have been drawn by setting the values (Table 1).

2.2. DNA-based computing

In this model, four bases (strands) are used: Adenine (A), Thymine (T), Cytosine (C), and Guanine (G). These bases pair complementarily (Popli, 2019). For example, if Adenine (A) is represented by '10', then Thymine (T) will be represented by '01'. Similarly, if Guanine (G) is represented by '11', then Cytosine (C) will be represented by '00'. Table 2 shows that out of 24 possible encodings, 8 comply with the Watson-Crick complementary rules (Mohamad Zulkufli et al. 2019). Additionally, XOR operations can be applied to these strands as suggested by the study of King and Gaborit (2007). Table 3 demonstrates this operation. Typically, translation functions are defined to convert pixel data to its corresponding DNA strands and vice versa. This study defines two functions, To_DNAEncode and To_DNADecode.

The To_DNAEncode function converts a pixel intensity value (ranging from 0 to 255 and using 8 bits) from the plain image to its DNA strands using one of the 8 rules. Conversely, the To_DNADecode function converts the DNA strands back to their decimal equivalents. For example, To_DNAEncode (156, 1) = GCTA, To_DNAEncode (156, 3) = TAGC, and To_DNAEncode (156, 7) = GCAT. Similarly, To_DNADecode (GCTA, 1) = 156, To_DNADecode (TAGC, 3) = 156, and To_DNADecode (GCAT, 7) = 156. Furthermore, for the DNA XOR operation, $\oplus$ (ATCG, GTAC) = GACT.

Table 1: System parameters in addition to primary values of Tinkerbell map

System parameter/initial value	Value
u_0	-0.8
v_0	-0.5
A	-0.9
B	-0.6013
C	2
D	0.5

Table 2: List of encoding rules for DNA

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

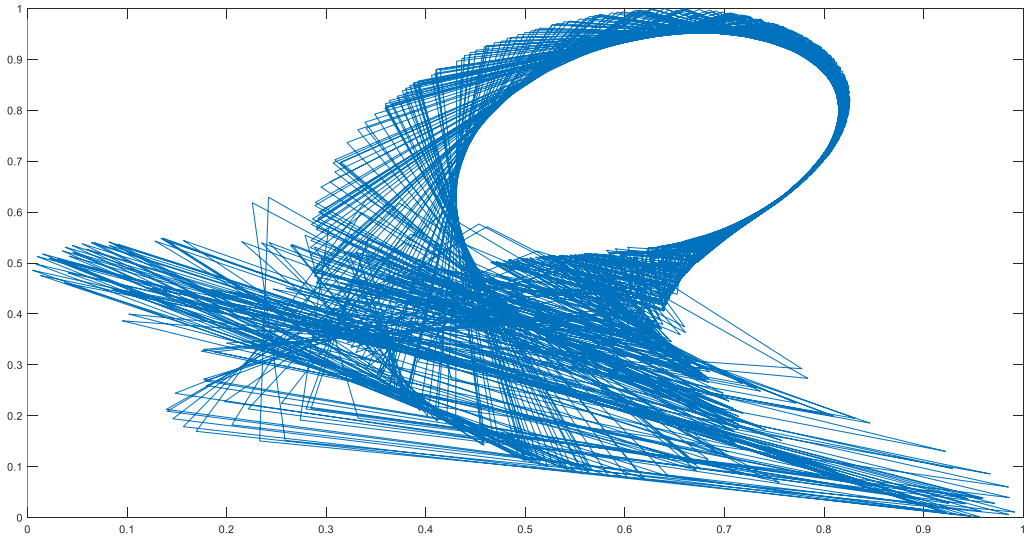


Fig. 1: Attractors of Tinkerbell's chaotic map

3. Proposed image encryption scheme

This work has developed a novel image cryptosystem employing swapping operations of pixels of the randomly selected pairs at both the DNA and decimal levels. Its corresponding flowchart can be seen in Fig. 2. Proposed algorithm contains a few stages. In the 1st stage, a grayscale image I is

provided to the encryption algorithm. $m \times n$ is assumed to be its size. Plaintext sensitivity was extracted by taking the average value average of the pixels. In the second stage, this value, along with the secret key, is fed to the Tinkerbell chaotic map (1) in order to spawn the streams u and v of random numbers. These two streams have been, in turn, operated to acquire the four random stream numbers, namely x, y, *choice*, and *mask*. In the 3rd stage, the decimal-level scrambling function has been invoked to acquire the scrambling effects among the pixels of the given image. Let the scrambled image obtained is I_1. In the fourth stage, decimal-level diffusion was conducted by XORing the scrambled image I_1 beside the mask image *mask* and getting the image I_2. In the fifth stage, DNA encoding has been carried out by recycling the stream y for the usage of the rules. In particular, the image I_2 was converted to its DNA strands, and we got the new image I_3. In the sixth stage, the DNA level scrambling was realized by calling the same function that was called at the decimal level scrambling. Let the image obtained is I_4. In the seventh stage, the key stream *mask* has been converted into the new stream *DNA mask* by using the key stream y for the rules of DNA encoding. In the eighth stage, diffusion influences are realized by carrying out the XOR operation, linking image I_4 and the DNA mask image. Let us get the image I_5. In the ninth stage, DNA decoding was carried out for image I_5 by using the streams x and y to calculate the rules dynamically. Let the final cipher image obtained is I_6.

3.1. Spawning of random data

This subsection illustrates the way the chaotic data has been produced. The following steps explain it.

Step 1: Restructure the given input image I to the size of $1 \times mn$ and calculate the average (avg) of its pixels' intensity values. Temper the initial value u0 of the system as follows.

$$u_o \leftarrow u'_o + \frac{Avg}{2^{20}} \tag{2}$$

where, u'_o and u_o correspond to the initial value formerly and afterward, combining the plaintext sensitivity.

Step 2: Spark the system given in Eq. 1 by giving it the set of primary values and the system parameters (Table 1). This map rendered us these streams of random numbers $u = [u_1, u_2, \ldots u_{mn+\epsilon}]$, $v = [v_1, v_2, \ldots v_{mn+\epsilon}]$, where the input image's size is (m, n). Here $\epsilon \geq 500$. These values are normally ignored in order to avoid several types of brief influences of the chaotic map being used.

Step 3: We have further customized the obtained random numbers' streams to seek compliance with the logic of the encryption algorithm we have conceived in the following equations.

$$\begin{cases} x\,(i) \leftarrow \lfloor mod(abs(u(i)) \times 10^{14}, m) \rfloor + 1 \\ y\,(i) \leftarrow \lfloor mod(abs(v(i)) \times 10^{14}, n) \rfloor + 1 \end{cases} \tag{3}$$

where, $\lfloor\rfloor$ refers to the floor function. Besides, mod () denotes the remainder operator. $i = 1, 2, \ldots, mn$.

Step 4: Two more streams *choice* and *mask* have been obtained through the following arithmetic manipulations over the streams y and x.

$$\begin{cases} choice\,(i) \leftarrow mod(x(i) + y(i),\ 4) + 1 \\ mask\,(i) \leftarrow mod(x(i) + y(i),\ 256) \end{cases} \tag{4}$$

where, i varies from 1 to mn inclusive

3.2. Image encryption algorithm

One of the novel features of this study is that encryption has been sought at the two levels, i.e., decimal level and DNA strands level. So, we will present it one by one in the dedicated subsections.

Table 3: DNA strands Exclusive-OR (XOR) operation

-	A	T	C	G
A	A	T	C	G
T	T	A	G	C
C	C	C	A	T
G	G	G	T	A

3.2.1. Decimal level encryption

Call the Algorithm 1 plus the parameters I, x, y and *choice*.

*1. **Algorithm 1:** Scrambling*
2. Input: I, x, y, choice
3. Output: I1
1. [m, n] = size (I)
2. for k ← 1 to mn do
3. switch (choice(k))
4. case 1:
5. if x(k) < m then
6. temp ← I(x(k), y(k))
7. 1(x(k), y(k)) ← I (x(k) + 1, y(k))
8. I(x(k) +1, y(k)) ← temp
9. case 2:
10. if x(k) > 1 then
11. temp ← I(x(k), y(k))
12. 1(x(k), y(k)) ← I (x(k) - 1, y(k))
13. I(x(k) - 1, y(k)) ← temp
14. case 3:
15. if y(k) < n then
16. temp ← I(x(k), y(k))
17. 1(x(k), y(k)) ← I (x(k), y(k) + 1)
18. I(x(k), y(k) + 1) ← temp
19. case 4:
20. if y(k) > 1 then
temp ← I(x(k), y(k))
21. 1(x(k), y(k)) ← I (x(k), y(k) - 1)
22. I(x(k), y(k) - 1) ← temp
23. I1 ← I

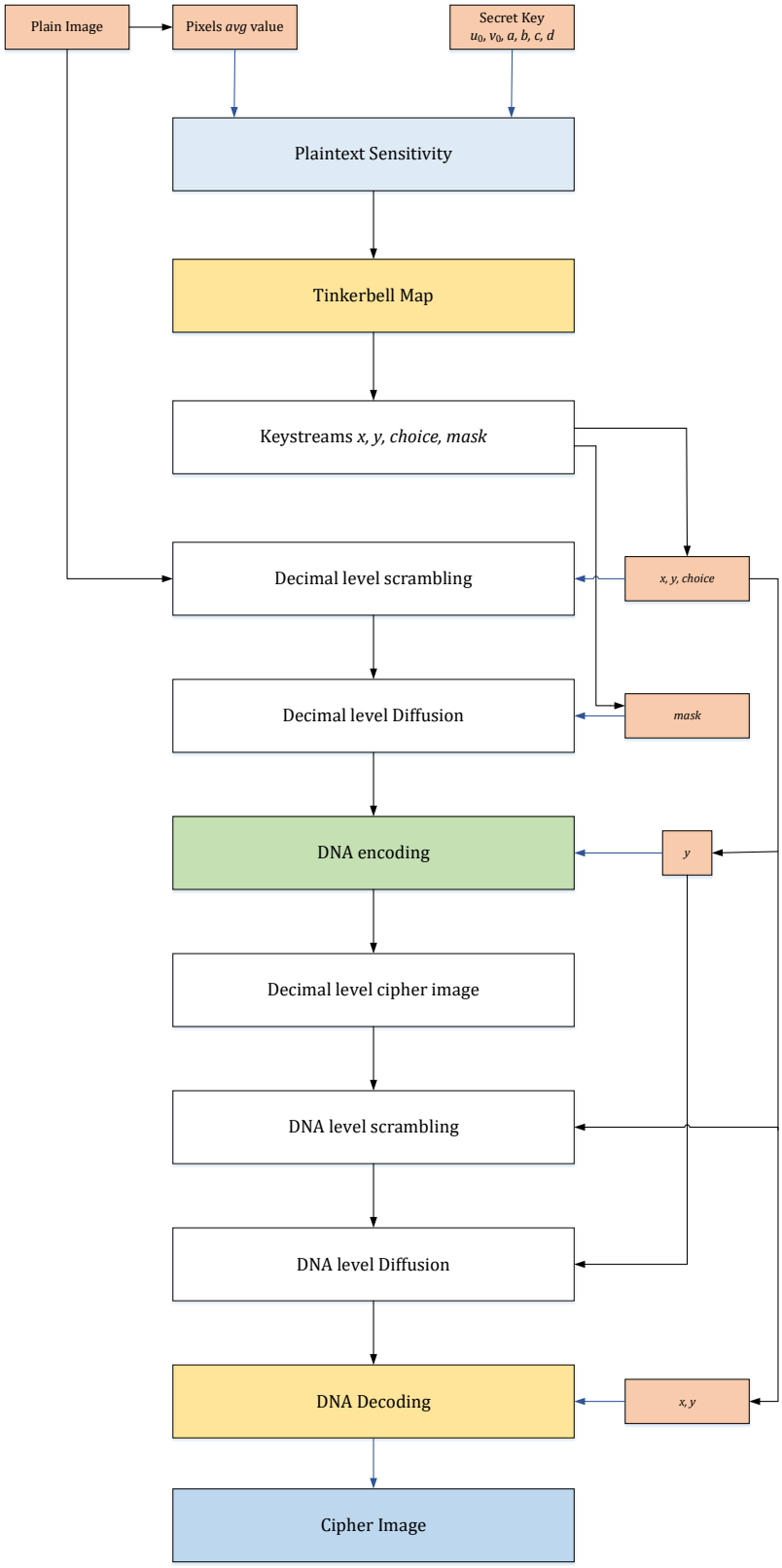


Fig. 2: Proposed image encryption scheme

Here, we will present the explanation of this algorithm in a step-by-step fashion. Line 1 calculates the size (m, n) of the given image I. The scrambling task has been carried out in the lines (2-23). The header of the for loop on line 2 is iterating mn times. In line 3, the switch statement is written with the parameter choice for the kth iteration of the for loop. Depending upon the value of *choice(k)* in each iteration of the for loop, one set of statements out of the four sets of statements written against the cases gets executed.

In each case, the pixel at the location ($x(k)$, $y(k)$) gets swapped with one of the pixels at the left, right, upper, and lower positions. For instance, in the case 1, the pixel at the location ($x(k)$, $y(k)$) has been swapped with its right pixel at the location ($x(k)$+1, $y(k)$). Similarly, the other three cases have been coped with. It is to be further noted that before the

swapping operation, it has been considered that the current location of the pixel $(x(k), y(k))$ should not be at either of the four edges of the image since in that case, the neighboring pixel will not be found. If such a case arises, no swapping operation has been carried out. In line 24, the scrambled image I have been assigned to the variable I_1. For the diffusion influences at the decimal level, remake the scrambled image I_1 to $1 \times mn$. Now image I_1 likewise, the key image mask are XORed and each to get the image I_2.

$$I_2(i) = I_1(i) \oplus mask(i) \quad (5)$$

where, $1 \le i \le mn$. I_2, having the size of $1 \times mn$, is an encrypted image at the decimal level.

3.2.2. DNA level encryption

Here, we have resized the decimal-level encrypted image I_2 and the stream of random numbers y into the 2D matrices with the dimensions of $m \times n$. Moreover, by exploiting the set of rules (Table 2), a translation of I_1 and y has been made. In this way, we obtained the corresponding DNA strands, A, T, C, G. These strands have been stored in the arrays I_3 and *DNA_mask*.

$$\begin{cases} I_3(i,j) = To\_DNAEncode(I_2(i,j), mod(y(i),8) + 1, \\ DNA\_mask(i,j) = To\_DNAEncode(x(i,j), mod(y(i),8) + 1) \end{cases} \quad (6)$$

For $1 \le i \le m$ and $1 \le j \le n$. Here stream y has been used to get the DNA encoding rule dynamically. Arrays I_3 and DNA mask comprise of mn DNA sequences. Each of these sequences has a length of 4. Further, *mod* (8) + 1 serves as the rule for decimal to DNA conversion. Now call Algorithm 1 again, including the parameters I_3, x, y and choice. Now, this algorithm will work based on the DNA strands, and we will get the image I_4. The next step is the diffusion at the DNA level. Here, the DNA XOR operation has been carried out, linking DNA mask and I_4 (Table 3).

$$I_5(i,j) = I_4(i,j) \oplus DNA\_mask(i,j) \quad (7)$$

Here, array I_5 is filled with the DNA strands after the XOR operation. Moreover, the symbol $\oplus$ refers to the XOR operation. In the following equation, these DNA strands are reversed by invoking the DNA decoding operations.

$$I_6(i,j) = To\_DNADecode(I_5(i,j), mod(x(i) \times y(i),8) + 1) \quad (8)$$

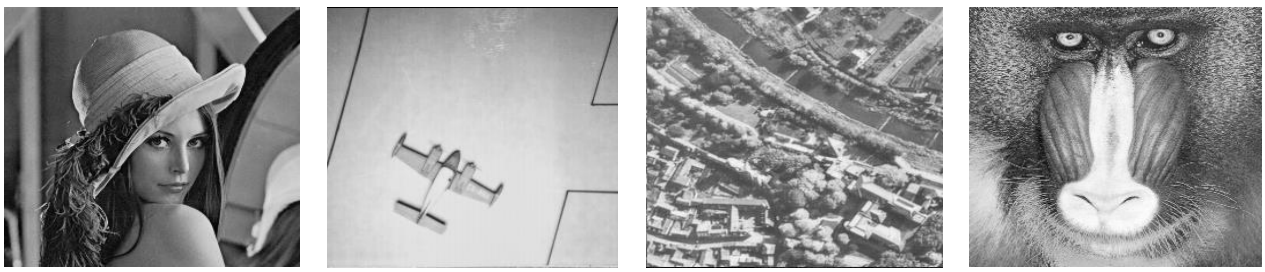
where, I_6 is the final output cipher image, including the size of $m \times n$. Since this security work has been deliberated along the private key cryptography principles, its decryption algorithm would be very slight in nature. Just taking the reverse of the steps of the encryption algorithm would render its corresponding decryption algorithm. So, there is no need to entertain it in depth.

4. Experimentation through computer simulation

In this section, we will demonstrate by taking some sample images for image encryption, as described in the previous section. The images chosen for this purpose are Lena, Airplane, Aerial, and Baboon, all with a size of 256 × 256. USC-SIPI Image Database is a great repository for test images. These images have been taken from it. Along with 64-bit double-precision, MATLAB 2016 version, permitting the IEEE standard 754, has been employed for the purpose of simulation. Figs. 3, 4, and 5 show the chosen input grayscale images, encrypted/cipher images, and restored/decrypted images correspondingly. The conversion of plain images into the cloud, in addition to the distorted format, indicates the success of the proof of concept and the successful implementation of the algorithm as well.

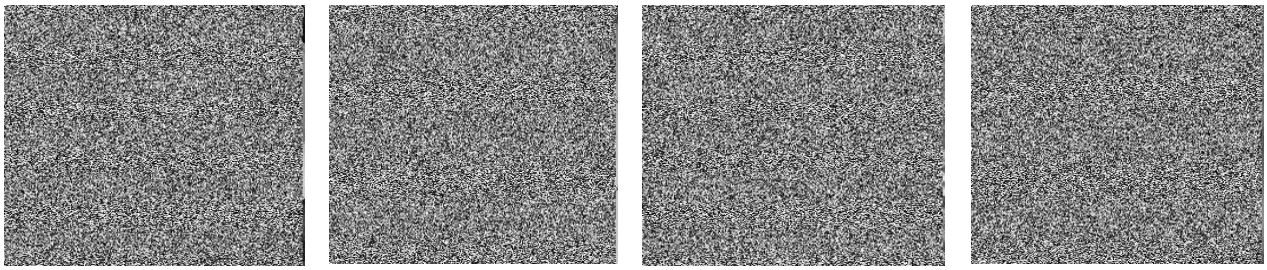
5. Validation

In this section, the proposed image cipher will be validated based on some chosen state-of-the-art metrics. Besides, we have chosen these image ciphers (Guo and Sun, 2022; Chai et al., 2019; Iqbal et al., 2021b; Wang et al., 2020) for the sake of comparison with the current work. It is to be noted that these works have used the concepts of DNA encoding coupled with chaotic systems in writing the image ciphers.



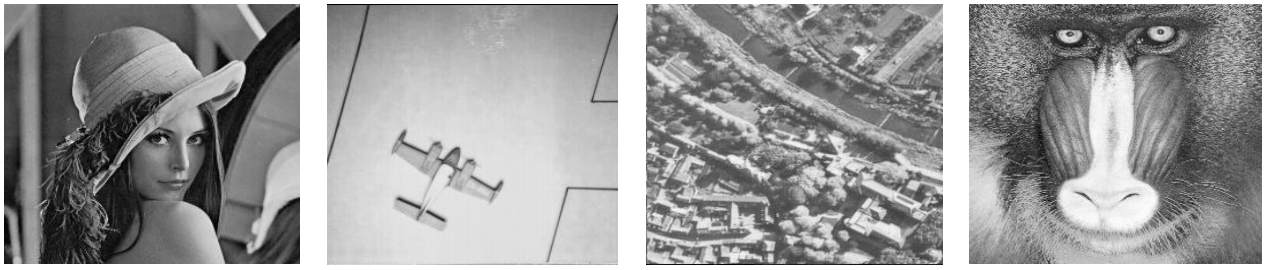
(a) (b) (c) (d)

Fig. 3: The plain input grayscale images: (a) Lena; (b) Airplane; (c) Aerial; (d) Baboon



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

Fig. 4: The cipher/encrypted images:(a) Lena; (b) Airplane; (c) Aerial; (d) Baboon



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

Fig. 5: The decrypted/restored images: (a) Lena; (b) Airplane; (c) Aerial; (d) Baboon

5.1. Analysis of keyspace

u_0, v_0, a, b, c, and d constitute the key of the suggested image cryptosystem. The precision of the computer system on which the current project has been carried out is 10−15. The key space comes out to be $(10^{15})^6 = 10^{90} \approx 2^{298}$. This value is $2^{298} >> 2^{100}$, which is the minimum threshold set by the cryptographers to avert the potential threats of brute force attack (Iqbal et al. 2022). Table 4 shows the resemblance of our algorithm to other algorithms based on this metric.

Although we could not beat any of the selected research, even then, we contend that we have easily crossed the minimum threshold to foil the brute force attacks. Actually, we selected the low-dimensional chaotic map to get the competence of encryption and decryption algorithms. This is the cause that the significant space of suggested work is relatively smaller. The selected studies, on the other hand, have employed higher dimensional maps. This is the reason that the key spaces of their algorithms are relatively higher

5.2. Statistical analysis

This analysis is frequently employed by image cryptographers. Under this heading, histogram and correlation analysis of the plain and encrypted images are carried out.

5.2.1. Histogram

The distribution of pixel intensity values is shown in a histogram. A typical plain image has a distinctive curved bar pattern in its histogram. A skilled hacker can easily extract information from such patterns. Therefore, a good encryption scheme should transform the plain image so that its histogram becomes uniform after encryption. Fig. 6 shows the histograms of both the encrypted and plain images of Lena. The histogram of the plain image, before applying the encryption algorithm, has a variable pattern. In contrast, the histogram of the encrypted image is relatively uniform. This uniformity makes it much harder for hackers to access the valuable and confidential information in the images.

5.2.2. Correlation coefficient analysis

Plain images have a strong inter-pixel linking among neighboring pixels. These neighboring pixels can be in any dimension, i.e., vertical, horizontal, or diagonal. A nice image cipher must be furnished with the intrinsic capability to break this correlation of inter-pixel intensities. Following is the mathematical equation of this concept (Iqbal et al. 2022).

$$CC = \frac{N\sum_{j=1}^{N}(x_j \times y_j) - \sum_{j=1}^{N} x_j \times \sum_{j=1}^{N} y_j}{\sqrt{(N\sum_{j=1}^{N} x_j^2 - (\sum_{j=1}^{N} x_j)^2)(N\sum_{j=1}^{N} y_j^2 - (\sum_{j=1}^{N} y_j)^2)}} \quad (9)$$

Here, we describe the variables of this equation. N indicates the total number of pixels in the image under consideration. Apart from that, x and y refer to the concentration values of the pixels. For analyzing the correlation coefficient of our proposed system, 3000 pairs of consecutive pixels were chosen in an arbitrary fashion from both plain likewise encrypted images of Lena.

Table 5 shows the obtained results of our proposed encryption scheme. Moreover, Table 6 proves the comparison of correlation coefficient of our algorithm with some existing algorithms (Guo and Sun, 2022; Chai et al., 2019; Iqbal et al., 2021b; Wang et al., 2020). This assessment is not straightforward, but the results of our study are very competitive. Besides, Fig. 7 appears the obtained correlation distribution of the neighboring pixels i.e., vertically, horizontally, and diagonally for both plain besides encrypted images of Lena.

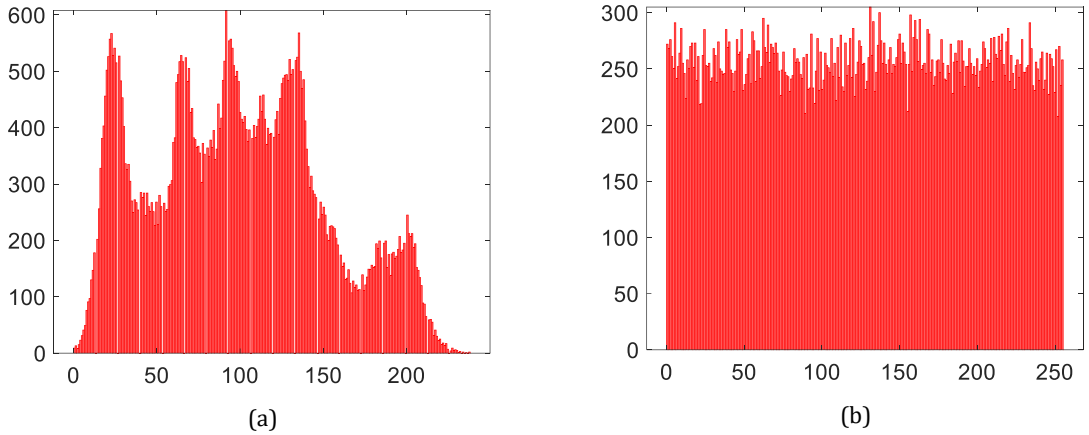


Fig. 6: Histogram analysis (a) Plain Lena image; (b) Encrypted Lena image

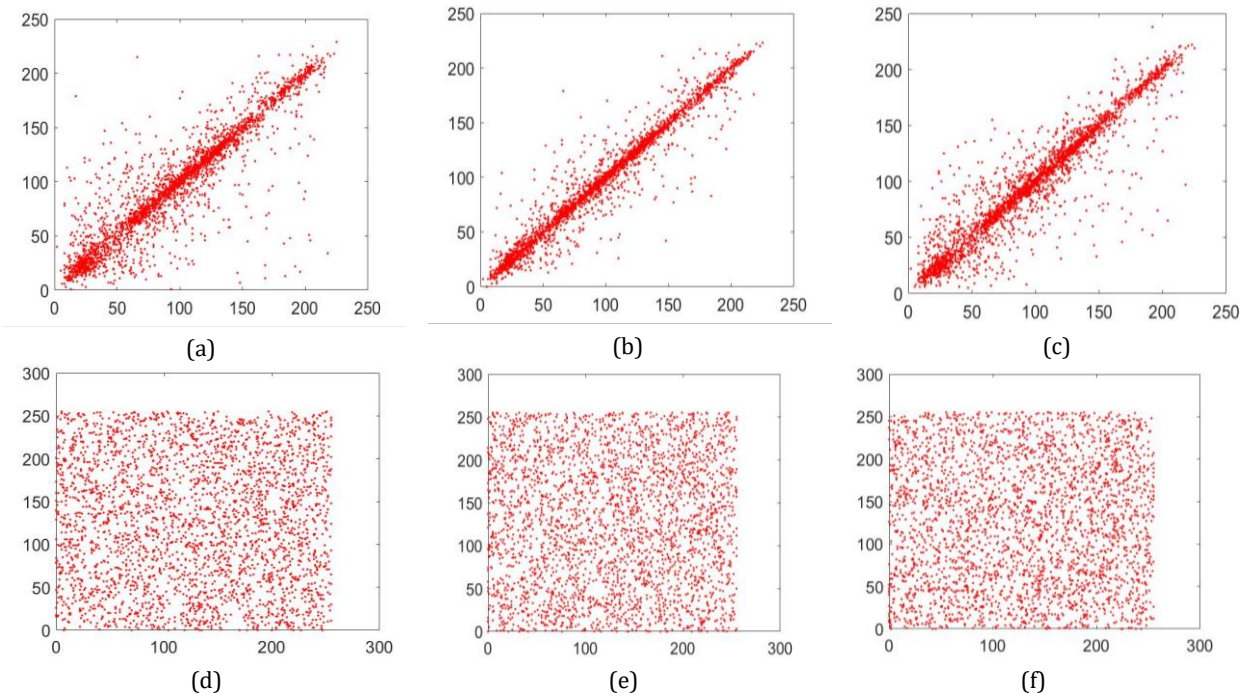


Fig. 7: Correlation distribution for image of Lena: (a) Plain Lena image in the horizontal orientation; (b) Plain Lena image in the vertical orientation; (c)Plain Lena image in the diagonal orientation; (d) Cipher Lena image in the horizontal orientation;(e) Cipher Lena image in the vertical orientation; (f) Cipher Lena image in the diagonal orientation

5.3. Information entropy analysis

The level of arbitrariness, unpredictability, and uncertainty in the images of the cipher is judged through the notion of information entropy by the image cryptographers. Information theorist Shannon (1949) worked on this concept in 1949. Below is its mathematical equation (Iqbal et al., 2022). In this equation, $Z(r)$ is the data entropy of image r. p relates to the possibility, and n indicates the total number of pixels in the specified image r. An image with 256 grayscale standards will have an extreme value of 8 for this metric. Table 7 shows the results of this important metric. Apart from that, a comparison with some other selected schemes has also been made. Our study beats all the other studies based on the Lena image. Besides, the average value of this metric for the four chosen images is also plausible.

$$Z(r) = \sum_{c=0}^{2^n-1} p(r_c) log \frac{1}{p(r_c)} \quad (10)$$

Table 4: Proposed algorithm key distance and resemblance with other works

Algorithms	Key Space
Proposed	$10^{90} \approx 2^{298}$
Guo and Sun (2022)	2892
Chai et al. (2019)	3.9402×10^{185}
Iqbal et al. (2021b)	2^{647}
Wang et al. (2020)	2.9647×10^{149}

Table 5: Results of correlation coefficient for Lena (cipher and plain images)

Image	Orientation of Correlation		
	Horizontal	Vertical	Diagonal
Lena image (plain)	0.9412	0.9727	0.9115
Lena image (cipher)	-0.0056	0.0009	-0.0067

5.4. Differential attack

Attackers can obtain the secret key through various methods, one of which is the variance attack. In this attack, the attacker uses two copies of the same image with a very small difference between them, often just a one-bit change.

Table 6: Correlation coefficients of the suggested work and its comparison

Image	Encryption algorithm	Direction		
		Horizontal	Vertical	Diagonal
Lena image (plain)		0.9412	0.9727	0.9115
Lena image (cipher)	Our algorithm	-0.0056	0.0009	0.0067
	Guo and Sun (2022)	-0.00257	0.003835	-0.00409
	Chai et al. (2019)	-0.0029	0.0013	-0.0026
	Iqbal et al. (2021b)	-0.0052	0.0086	-0.0020
	Wang et al. (2020)	-0.0021	0.0009	0.0003

Table 7: Results (information entropy)

Encryption algorithm	Images	Original	Encrypted
		Horizontal	Vertical
Our algorithm	Lena	7.5954	7.9975
	Airplane	6.3012	7.9972
	Aerial	7.0184	7.9972
	Baboon	6.2387	7.9973
Guo and Sun (2022)	Lena	-	7.9974
Chai et al. (2019)	Lena	7.3003	7.9971
Iqbal et al. (2021b)	Lena	7.2699	7.9974
Wang et al. (2020)	Lena	7.5788	7.9972

These two images are then encrypted using the same algorithm to produce their corresponding cipher images. By analyzing the relationship between these two cipher images, the attacker can potentially uncover the secret key. The effectiveness of a cipher against a variance attack is measured using two metrics: Unified Average Changing Intensity (UACI) and Number of Pixels Change Rate (NPCR). These metrics help determine the robustness of the encryption. Their mathematical formulas are provided below (Iqbal et al., 2022).

$$NPCR = \frac{\sum_{e,k} D(e,k)}{G \times T} \times 100\% \quad (11)$$

where, G and T denote the width plus height of the image separately. D (e, k) can be specified by:

$$D(e,k) = \begin{cases} 1, if\ \ S(e,k) \neq S'(e,k) \\ 0, if\ S(e,k) = S'(e,k) \end{cases} \quad (12)$$

$$UACI = \frac{1}{G \times T}\left[\sum_{e,k} \frac{|S(e,k) - S'(e,k)|}{255}\right] \times 100 \quad (13)$$

where, S and S' are individually the ciphered images formerly, and afterward, one pixel of the plain image is replaced. Table 8 shows the acquired values of NPCR plus UACI for the proposed encryption outline.

Table 9 draws a resemblance of these metrics for the proposed scheme with some other schemes. Our algorithm beats the studies (Guo and Sun, 2022; Chai et al., 2019; Wang et al., 2020) based on the metric of NPCR. Apart from that, the outcomes of the UACI of the proposed study are better than Guo and Sun (2022) and Iqbal et al. (2021b).

Table 8: Results (NPCR furthermore UACI) for the chosen images

Images	NPCR	UACI
Lena	99.6298	33.4292
Airplane	99.6168	33.6303
Aerial	99.5912	33.4749
Baboon	99.6283	33.4212
Average	99.6165	33.4889

Table 9: Similarity of security metrics NPCR and UACI on the Lena image by various schemes

Algorithm	NPCR (%)	UACI (%)
Our algorithm	99.6298	33.4292
Guo and Sun (2022)	99.6123	33.4178
Chai et al. (2019)	99.6067	33.5000
Iqbal et al. (2021b)	99.6302	33.4277
Wang et al. (2020)	99.5956	33.4588

5.5. Noise and data loss/crop attacks

Nice image encryption systems have the inbuilt property of withstanding the potential attacks that may be aimed at them. Noise plus data loss attacks are the attacks that may be launched upon the image ciphers. To determine this feature of the proposed work, we have chosen the cipher images of Lena and Baboon. We have synthetically included Pepper and Salt noise in the cipher image of Lena (Fig. 8a). Besides, the upper half of the cipher image of Baboon has been discarded (Fig. 8c). Now the decryption algorithm has been called one by one upon them. The decrypted/restored images are depicted in the Fig. 8b and 8d. One can easily recognize these images. Hence, we are adjusted to say that the proposed image cipher has the built-in potential to thwart the potential attacks of noise and data crop.

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

Fig. 8: Noise and data crop attack analysis:(a) Pepper and Salt noise with density of 0.1 in the Lena image; (b) Decrypted Lena image from (a); (c) Baboon image with a data crop attack of 1/2 × m × n; (d) Decrypted Baboon image from (c)

5.6. Speed analysis

This project was conducted using the following hardware and software specifications: 8 GB memory, Intel(R) Core(TM) i5-4210U CPU @ 1.70 GHz to 2.40 GHz. MATLAB version R2016a was used, and the primary operating system was Windows 10. Ensuring the security of cryptographic products is a top priority for developers. However, products with faster response times are more attractive for applications in both industry and academia. Table 10 shows the speed of the proposed scheme. Additionally, a comparative analysis was conducted with other works, demonstrating that the proposed method is faster than the algorithm by Iqbal et al. (2021b).

Table 10: Speed of algorithm and its resemblance

Algorithm	Images	Speed (sec)
Our algorithm	Lena	2.9043
	Airplane	2.8703
	Aerial	2.7012
	Baboon	2.7520
	Average	2.8087
Guo and Sun (2022)	Lena	0.1647
Chai et al. (2019)	-	-
Iqbal et al. (2021b)	Lena	5.5690
Wang et al. (2020)	-	-

6. Conclusion

By using the Tinkerbell chaotic map and pixel swapping operations from randomly selected pairs, this work proposes an efficient and secure image cryptosystem. Most previous studies used higher-dimensional and hyperchaotic systems to generate random data for confusion and diffusion operations. In contrast, this work uses a 2D map to improve the cipher's speed. Additionally, two more streams were generated by recycling the existing streams. Confusion is achieved by swapping pixels from randomly selected pairs at both the DNA and decimal levels. Similarly, diffusion is applied using the XOR process at both levels. Performance and security analyses show that the proposed image cipher is both efficient and secure, with potential real-world applications. Future work will explore the use of cellular automata to enhance security and robustness. This scheme can also be adapted to work with color and binary images.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Adleman LM (1994). Molecular computation of solutions to combinatorial problems. Science, 266(5187): 1021-1024. https://doi.org/10.1126/science.7973651 **PMid:7973651**

Babaei M (2013). A novel text and image encryption method based on chaos theory and DNA computing. Natural Computing, 12(1): 101-107. https://doi.org/10.1007/s11047-012-9334-9

Batool T, Abbas S, Alhwaiti Y, Saleem M, Ahmad M, Asif M, and Elmitwal NS (2021). Intelligent model of ecosystem for smart cities using artificial neural networks. Intelligent Automation and Soft Computing, 30(2): 513-525. https://doi.org/10.32604/iasc.2021.018770

Chai X, Fu X, Gan Z, Lu Y, and Chen Y (2019). A color image cryptosystem based on dynamic DNA encryption and chaos. Signal Processing, 155: 44-62. https://doi.org/10.1016/j.sigpro.2018.09.029

Chai X, Zheng X, Gan Z, Han D, and Chen Y (2018). An image encryption algorithm based on chaotic system and compressive sensing. Signal Processing, 148: 124-144. https://doi.org/10.1016/j.sigpro.2018.02.007

Enayatifar R, Abdullah AH, and Isnin IF (2014). Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. Optics and Lasers in Engineering, 56: 83-93. https://doi.org/10.1016/j.optlaseng.2013.12.003

Guesmi R, Farah MAB, Kachouri A, and Samet M (2016). A novel chaos-based image encryption using DNA sequence operation and secure hash algorithm SHA-2. Nonlinear Dynamics, 83: 1123-1136. https://doi.org/10.1007/s11071-015-2392-7

Guo Z, and Sun P (2022). Improved reverse zigzag transform and DNA diffusion chaotic image encryption method. Multimedia Tools and Applications, 81(8): 11301-11323. https://doi.org/10.1007/s11042-022-12269-5

Hanif M, Abbas S, Khan MA, Iqbal N, Rehman ZU, Saeed MA, and Mohamed EM (2020). A novel and efficient multiple RGB images cipher based on chaotic system and circular shift operations. IEEE Access, 8: 146408-146427. https://doi.org/10.1109/ACCESS.2020.3015085

Iqbal N, Abbas S, Khan MA, Alyas T, Fatima A, and Ahmad A (2019). An RGB image cipher using chaotic systems, 15-puzzle problem and DNA computing. IEEE Access, 7: 174051-174071. https://doi.org/10.1109/ACCESS.2019.2956389

Iqbal N, Abbas S, Khan MA, Fatima A, Ahmed A, and Anwer N (2020a). Efficient image cipher based on the movement of king on the chessboard and chaotic system. Journal of Electronic Imaging, 29(2): 023025. https://doi.org/10.1117/1.JEI.29.2.023025

Iqbal N, Hanif M, Abbas S, Khan MA, Almotiri SH, and Al Ghamdi MA (2020b). DNA strands level scrambling based color image encryption scheme. IEEE Access, 8: 178167-178182. https://doi.org/10.1109/ACCESS.2020.3025241

Iqbal N, Hanif M, Abbas S, Khan MA, and Rehman ZU (2021a). Dynamic 3D scrambled image based RGB image encryption scheme using hyperchaotic system and DNA encoding. Journal of Information Security and Applications, 58: 102809. https://doi.org/10.1016/j.jisa.2021.102809

Iqbal N, Hanif M, Rehman ZU, and Zohaib M (2022). On the novel image encryption based on chaotic system and DNA computing. Multimedia Tools and Applications, 81(6): 8107-8137. https://doi.org/10.1007/s11042-022-11912-5

Iqbal N, Naqvi RA, Atif M, Khan MA, Hanif M, Abbas S, and Hussain D (2021b). On the image encryption algorithm based on the chaotic system, DNA encoding, and castle. IEEE Access, 9: 118253-118270. https://doi.org/10.1109/ACCESS.2021.3106028

Khan AH, Abbas S, Khan MA, Farooq U, Khan WA, Siddiqui SY, and Ahmad A (2022). Intelligent model for brain tumor identification using deep learning. Applied Computational Intelligence and Soft Computing, 2022: 8104054. https://doi.org/10.1155/2022/8104054

Khan MA, Umair M, Saleem MA, Ali MN, and Abbas S (2019). CDE using improved opposite based swarm optimization for MIMO

systems. Journal of Intelligent and Fuzzy Systems, 37(1): 687-692. https://doi.org/10.3233/JIFS-181127

Khan MF, Ghazal TM, Said RA, Fatima A, Abbas S, Khan MA, Issa GF, Ahmad M, and Khan MA (2021). An IoMT-enabled smart healthcare model to monitor elderly people using machine learning technique. Computational Intelligence and Neuroscience, 2021: 2487759. https://doi.org/10.1155/2021/2487759 **PMid:34868288 PMCid:PMC8639263**

King OD and Gaborit P (2007). Binary templates for comma-free DNA codes. Discrete Applied Mathematics, 155(6-7): 831-839. https://doi.org/10.1016/j.dam.2005.07.015

Kolivand H, Hamood SF, Asadianfam S, and Rahim MS (2024). Image encryption techniques: A comprehensive review. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-023-17896-0

Liu L, Hao S, Lin J, Wang Z, Hu X, and Miao S (2018). Image block encryption algorithm based on chaotic maps. IET Signal Processing, 12(1): 22-30. https://doi.org/10.1049/iet-spr.2016.0584

Liu L, Zhang Q, and Wei X (2012). A RGB image encryption algorithm based on DNA encoding and chaos map. Computers and Electrical Engineering, 38(5): 1240-1248. https://doi.org/10.1016/j.compeleceng.2012.02.007

Ma Y, Li C, and Ou B (2020). Cryptanalysis of an image block encryption algorithm based on chaotic maps. Journal of Information Security and Applications, 54: 102566. https://doi.org/10.1016/j.jisa.2020.102566

Malhotra P, Singh Y, Anand P, Bangotra DK, Singh PK, and Hong WC (2021). Internet of things: Evolution, concerns and security challenges. Sensors, 21(5): 1809. https://doi.org/10.3390/s21051809 **PMid:33807724 PMCid:PMC7962037**

Mohamad Zulkufli NL, Turaev S, Mohd Tamrin MI, and Messikh A (2019). Watson-Crick linear grammars. In: Abawajy J, Othman M, Ghazali R, Deris M, Mahdin H, and Herawan T (Eds.), Proceedings of the International Conference on Data Engineering 2015 (DaEng-2015). Lecture Notes in Electrical Engineering, 520: 403-412. Springer, Singapore, Singapore. https://doi.org/10.1007/978-981-13-1799-6_42

Özkaynak F, Özer AB, and Yavuz S (2013). Security analysis of an image encryption algorithm based on chaos and DNA encoding. In the 21st Signal Processing and Communications Applications Conference, IEEE. Haspolat, Turkey: 1-4. https://doi.org/10.1109/SIU.2013.6531597

Popli M (2019). DNA cryptography: A novel approach for data security using flower pollination algorithm. In the Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur, India.

Shannon CE (1949). Communication theory of secrecy systems. The Bell System Technical Journal, 28(4): 656-715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x

Siddiqui SY, Khan MA, Abbas S, and Khan F (2022). Smart occupancy detection for road traffic parking using deep extreme learning machine. Journal of King Saud University-Computer and Information Sciences, 34(3): 727-733. https://doi.org/10.1016/j.jksuci.2020.01.016

Sun S, Guo Y, and Wu R (2019). A novel image encryption scheme based on 7D hyperchaotic system and row-column simultaneous swapping. IEEE Access, 7: 28539-28547. https://doi.org/10.1109/ACCESS.2019.2901870

Wang X, Wang Y, Zhu X, and Luo C (2020). A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level and DNA level. Optics and Lasers in Engineering, 125: 105851. https://doi.org/10.1016/j.optlaseng.2019.105851

Wei X, Guo L, Zhang Q, Zhang J, and Lian S (2012). A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system. Journal of Systems and Software, 85(2): 290-299. https://doi.org/10.1016/j.jss.2011.08.017

Wu X, Wang K, Wang X, Kan H, and Kurths J (2018). Color image DNA encryption using NCA map-based CML and one-time keys. Signal Processing, 148: 272-287. https://doi.org/10.1016/j.sigpro.2018.02.028

Xiong Z, Wu Y, Ye C, Zhang X, and Xu F (2019). Color image chaos encryption algorithm combining CRC and nine palace map. Multimedia Tools and Applications, 78(22): 31035-31055. https://doi.org/10.1007/s11042-018-7081-3

Zhang Q, Guo L, and Wei X (2010). Image encryption using DNA addition combining with chaotic maps. Mathematical and Computer Modelling, 52(11-12): 2028-2035. https://doi.org/10.1016/j.mcm.2010.06.005

Zhang Q, Guo L, and Wei X (2013). A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. Optik-International Journal for Light and Electron Optics, 124(18): 3596-3600. https://doi.org/10.1016/j.ijleo.2012.11.018

Zhen P, Zhao G, Min L, and Jin X (2016). Chaos-based image encryption scheme combining DNA coding and entropy. Multimedia Tools and Applications, 75: 6303-6319. https://doi.org/10.1007/s11042-015-2573-x

Zhou Y, Bao L, and Chen CP (2014). A new 1D chaotic system for image encryption. Signal Processing, 97: 172-182. https://doi.org/10.1016/j.sigpro.2013.10.034