Using Magic Square Chaotic Algorithm and DNA for Evolutionary-based Image Encryption Operators

Mahdi Tahbaz[1],Hossein Shirgahi[2*], Mohammad Reza Yamaghani[1]

Abstract – The development of digital technologies has improved the transfer of data over the Inte rnet in recent years. Image encryption is a technique to ensure security in information transfers. The current paper presents an evolutionary model on the basis of a hybridization of DNA biomolecule o perators and the LS2 Map chaos function for encryption of image. The model proposed here include s three stages. In the initial stage, the MSC (Magic Square Chaotic) algorithm and a secret key are utilized with the SHA-256 algorithm to determine the initiating the LS2 Map function value, which i s then employed to manipulate the pixels of the image. Then, DNA biomolecule operators and the c haos function are used for propagation. Additionally, the previous stages process is iterated with the starting population of the genetic algorithm in the third stage. Afterward, the optimization is carried out through genetic algorithm operators. The results indicate that the introduced model is superior to other rivals. Furthermore, as for the high level of entropy obtained, the model exhibits strong resista nce to common attacks.

Keywords: LS2 map, DNA operators, Magic square algorithm, Image encryption, Genetic algorithm, hash function.

1. Introduction

Every day, individuals use audio, images, and video to communicated through the media Specifically, information contained in images can be easily accessed via wireless networks and the Internet. Thus, ensuring the security of digital image content is essential to prevent illegal viewing, copying, or editing [23, 27]. There are several methods and algorithms proposed for image encryption [3]. Cryptography can be non-textual or textual [21]. Given the data volume (to many pixels), encryption of image is different significantly from texts. As a result, text and image protection are completely different and the methods for encrypting texts may not be used for encryption of images [4,18]. The systems designed for image encryption mostly used chaotic and turbulent functions because of the distinctive characteristics of chaotic maps, like random chaotic sequence values, excessive sensitivity of initial values, and simplicity of the maps. Such methods essentially aim to divide pixels of image into smaller units (bits) and encryption is carried out based on operators like XOR. Therefore, the encryption cannot be recognized unless the encryption key is given. In addition, the proposed algorithms feature unique specifications. That is, some of them have a higher sensitivity to the starting condition, while others have sensitivity to other parameters [5, 10-13]. The systems for cryptography rely on diffusion and permutation of pixels and creating chaotic patterns. Furthermore, because of the encryption keys with large spaces, chaotic systems can stand exhaustive search attacks. In general, researchers primarily utilize complex chaotic systems or integrate novel methods of encryption and available chaotic systems to improve the security of cryptographic algorithms for image encryption [24-26]. There are two different methods to encrypt images including diffusion and permutation. In the latter, pixels position is changed using matrix transformation or chaotic sequences like checkerboard transformations. In these algorithms keep the value of pixels unchanged while the pixel position is altered. Therefore, these algorithms are vulnerable to statistical analysis. The value of pixels is changed by diffusion phase based on a chaotic sequence. In comparison to permutation technique, diffusion provides a higher level of security; however, its efficiency is lower in terms of cryptographic impact. Therefore, these two techniques are combined to enhance effectiveness and security of cryptography [2,16].

Recently, the properties and structural complexity of biological molecules such as RNA and DNA have been extensively discussed in the literature. There is a higher

1 Department of Computer Engineering, Lahijan Branch, Islamic Azad University, Lahijan, Iran.

Email: mahdi_tahbaz@liau.ac.ir, o_yamaghani@liau.ac.ir

2* Corresponding Author : Department of Computer Engineering, Jouybar Branch, Islamic Azad University, Jouybar, Iran.

Email: h.shirgahi@jouybariau.ac.ir

Received: 08.08.2023 ; Accepted:06.03.2024

efficiency in biomolecule encryption techniques for emerging security applications with several levels as for security and performance. As a result, a mixture of DNA sequences and chaotic systems is used for encryption of images with a higher level of security. Thus, cryptography based on DNA is a desirable supplement to the standard mathematical cryptography [11, 14, 17]. There have been many studies on utilizing evolutionary algorithms and chaotic functions in encryption of images. The chaotic functions are mostly utilized to generation starting encryption images; while evolutionary algorithm and general solutions are used to achieve a higher level of quality of solutions [6, 19]. Abbasi et al. [1, 7] encrypted images using a novel evolutionary chaotic model using the specifications of biological molecules. The introduced method demonstrated strong reliability against normal attacks thanks to 256-bit concealed key and incorporating the chaotic function's dependency on the key to produce sequences that are quasi-random, along with the integration of evolutionary algorithms and biomolecule operators.

A new model was introduced that utilizes genetic algorithms to increase entropy values and attenuates correlation coefficients in image pixels by employing DNA operators. The goal is to enhance the diffusion criterion through mixing magical square algorithm and the LS2 Map chaos function, ultimately increasing the level of turbulence to address drawbacks of image encryption. The remaining of the article is arranged as follows: the fundamental concepts of chaotic functions, the magic square algorithm, biological molecules, hash functions, and genetic algorithms are introduced in Section two; Section 3 provides a more detailed focus on the model. The results of simulations for the proposed model are given in Section 4 along with comparing them with those of previous models. Eventually, section five concludes the paper with conclusion remarks.

2. Basic Concepts

The basics of chaotic functions, magic square algorithms, biological molecules, hash functions, and genetic algorithms are discussed as follows:

2.1 Chaos Function

The term "chaos" explains the complicated function of dynamic systems. It indeed is a behavior exhibited by these systems, offering rapid and highly secure methods for steganography and encryption. It boasts unique benefits, including sensitivity to initial parameters and conditions, unpredictability, and random behavior [16,26].

2.1.1. Logistic-Sinusoidal Mapping

As one of the main mappings in chaos theory, the logistic mapping (Eq.1) is characterized by complex chaotic behavior and simple equations in its output. In addition, the sine mapping (Eq. 2) has a highly similar behavior to the logistic mapping. Failure to generation consistently chaotic output in specific intervals is one issue with these maps, and they also exhibit inefficient chaotic behavior within their chaotic interval. A novel approach known as LS Map was introduced to deal with this drawback (Eq. 3). By incorporating an additional operation, altering the output using the mod operation, and giving it back to the interval, the authors enhanced the chaotic performance of the two maps [0, 1]. Still, according to the results, we do not have a uniform mapping. Hence, the mapping is not a suitable option for encryption, given that the maximum intervals of distribution are easily determined through a simple statistical analysis. As illustrated Figure 1, it is possible to have a uniform output utilizing XOR operator in series (Eq. 4). The output of this result is further combined with two sinusoidal and logistic maps and then constrained to the interval [0,1] (LS2 Map) [8, 28].

$$X\_(n+1) = r.X\_n\,(1 - X\_n) \quad (1)$$

$$(X\_(n+1) = (4 - r)\ .sin\,(\pi.X\_n)) / 4 \quad (2)$$

$$X\_(n+1) = (r.X\_n\,(1 - X\_n) + ((4 - r)\ .\sin\,(\pi.X\_n)) / 4) mod\ 1 \quad (3)$$

Where $r \in (0..4]$

$$X\_(n+1) = (((r.X_n(1 - X_n) \oplus ((4 - r)\ .\sin\,(\pi.X\_n)))) / 4) + (r.X\_n\,(1 - X\_n) + ((4 - r)\ .\sin\,(\pi.X\_n)) / 4)) mod\ 1 \quad (4)$$

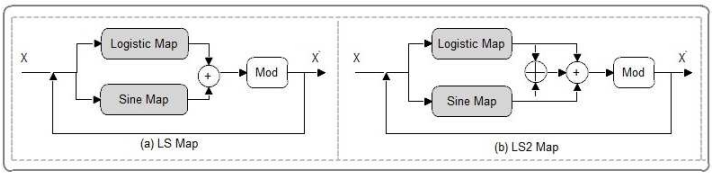


Fig. 1. (a) LS Map and (b) LS2 Map

2.2 Biological Molecules

Three categories of biological molecules are used by all forms of complex life on Earth to carry out essential functions. Proteins have various function in cells like providing structure, acting as chemical transporters, and promoting catalytic functions. Meanwhile, DNA and RNA contain genetic information, which is transferred to next generations [17,31].

2.2.1. DNA

DNA (Deoxyribonucleic Acid) is the chemical compound that contains all the genetic information and hereditary traits of living organisms. This molecule contains two very long strands that coil and form a double-helix structure. DNA is found in all cells of living organisms and is passed from parent cells to offspring. Based on Figure 2, nitrogenous organic bases have a circular structure and exist in four forms within the DNA molecule: guanine (G), cytosine (C), adenine (A), and thymine (T).

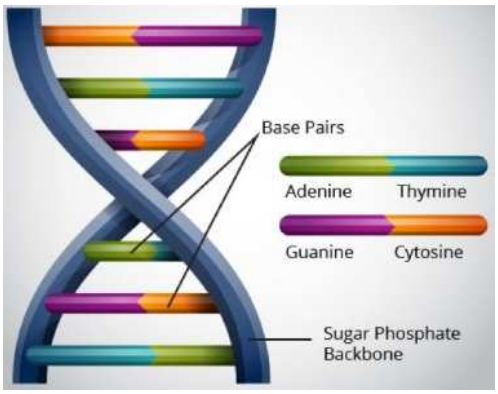


Fig. 2. Structure of DNA molecules

2.2.2. Image encryption and decoding by DNA

Adelman conducted the first DNA computational experiment in 1967. Furthermore, he developed a novel method in the field of molecular computing to address combinatorial problems related to information principles. The emergence of DNA computing marked the formation of a new field that utilized DNA sequences as carriers of information and as implementation tools.

Bases A and T on one hand and G and C on the other hand are complementary to each other. As 0 and 1 are complementary, 00 and 11 will also be complementary. Similarly, 01 and 10 are complementary and have the same relationship. There are 24 different encryption modes achievable utilizing the four combinations. Nevertheless, considering the complementary nature of the A-T and G-C base pair rules, only the four modes presented in Table 1 will be valid [17, 20].

As shown in Table 1, each element in a row or column is repeated only once. Various operators, like subtraction, addition, XNOR, and XOR, are utilized in image encryption according to Table 2.

As illustrated in Table 1, each element in a row or column is repeated only once. Different operators including addition and subtraction, XOR and XNOR are used in image encryption based on Table 2.

Table 1. Rules of DNA sequences operators for decoding and encryption

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

Table 2. XOR, Subtraction, addition, and XNOR operations for DNA operators.

	+				-				XOR				XNOR			
	A	G	C	T	A	G	C	T	A	G	C	T	A	G	C	T
A	A	G	C	T	A	T	C	G	A	G	C	T	T	C	G	A
G	G	C	T	A	G	A	T	C	G	A	T	C	C	T	A	G
C	C	T	A	G	C	G	A	T	C	T	A	G	G	A	T	C
T	T	A	G	C	T	C	G	A	T	C	G	A	A	G	C	T

2.3 Magic Square Algorithm

There are n rows and n columns in a magic square, containing n^2 cells filled with unique natural numbers (see Eqs. 5 and 6).

$$1 + 2 + 3 + \cdots + n^2 = n.M \qquad (5)$$

$$\sum_{i=1}^{n^2} i = n.M. \qquad (6)$$

M refers to the total numbers on columns, rows, or principal diagonal of the magic square of order n, so that the total value of figures in the magic square is equal n*M (Eq. 7).

$$n.M = \frac{n^2(n^2+1)}{2} \qquad (7)$$

The total value of all columns, rows, and diagonals of a magic square is fixed. The figures in each cell of a magic square of order n includes all figures between 1 to n^2. The magic square is known as n^2 order magic square. Here, the total of the columns, rows, and diagonals of this square is calculated based on Equation 8.

$$M = \frac{n(n^2+1)}{2} \qquad (8)$$

The 8x8 magic square are used in the model proposed here. Thus the total value of each column, row, and

diagonal of the magic square is equal to eight, which equals (8^2 + 1)/2 = 260. The magic square 8x8 arrangement is: first, counting is started from the cell in the top left corner, and the numbers given to the non-diagonal cells are added to them (black numbers in Figure 3). Afterward, the count in the next row at the end of each row is restarted from the cells on the left. Then, after going through all non-diagonal cells, counting begins at the right side of the row below and the number dedicated to the diagonal cells are inserted (figures in blue and red color in Figure 3). Then, we start counting at the right to left, and with each row completed, it restarts in the top row from the right side cell.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

→

64	2	3	61	60	6	7	57
9	55	54	12	13	51	50	16
17	47	46	20	21	43	42	24
40	26	27	37	36	30	31	33
32	34	35	29	28	38	39	25
41	23	22	44	45	19	18	48
49	15	14	52	53	11	10	56
8	58	59	5	4	62	63	1

Fig. 3. The 8*8 magic square left (before moving) and right (after moving)

2.4. Hash function

A hash is a mathematical function that converts any input value to another compressed value. The input of the hash function is a value of unknown length; however, the output is always of a constant length. Hash functions are widely-used and are used in almost all information security applications [8].

2.4.1. Encryption hash function

Cryptographic hash function with its own unique features is used for security and authentication applications. A cryptographic hash function is:

1- Definite and Constant

Each specified input is converted to a fixed output by the hash function. In this process, it does not matter how many times or when the hash function affects the input, in any case the output is fixed. If this is not the case and the value of the hash function is different in each effect, data retrieval will be impossible.

2 - Calculation speed

Calculation speed of the input hash is a key factor for the efficiency of the hash function.

3- One-way function

The cryptographic hash function is a one-way relationship, it means that the output can be easily calculated by hashing. However, an output corresponding to an input is a barrier to obtaining the input.

2.4.2. Secure Hash Algorithm (SHA-256)

The SHA-256 algorithm is characterized by one of the most secure hash functions and is one of the branches of SHA-2 created by the National Security Agency in 2001 as a replacement for SHA-1. SHA-256 is a 256-bit encrypted hash function, with three features which make the algorithm highly secure:

a) Restructuring raw data from a hash value is almost not possible. An attack to create raw data requires 2^{256} attempts!

b) It is very unlikely that two messages with the same amount of hashes (so-called collisions) are possible. With 2^{256} possible hash values (which is more than the total number of known atoms in the universe), the probability that two numbers are the same is infinitely unimaginable.

c) A trivial change in the original data alters the hash value so that it is unclear whether the new hash value was taken from the same data – i.e. avalanche effect.

2.5. Genetic Algorithm

As a specialized type of evolutionary algorithm, a genetic algorithm employs biological techniques like mutation and inheritance. To achieve the best formula to predict or match the patterns, genetic algorithms (GA) actually utilize the idea of natural selection as introduced by Darwin. They are frequently a viable choice for regression-based prediction methods. In artificial intelligence, a GA is a method for programming that employs genetic evolution for solving problems. The problem involves inputs that are transformed into solutions through a simulated process of genetic evolution. Subsequently, the Fitness Function assesses the solutions as candidates. Furthermore, with an exit condition, the algorithm will terminate. The GA is typically an iteration-based algorithm in which most components are selected through random processes [19].

3. Proposed Model

The introduced model contains four phases. At the beginning, a 256-bit secret key is processed based on the SHA-256 algorithm to obtain the starting value for the LS2map function. Then, in the next phase (permutation), the encryption process uses chaos sequence and magic algorithm to enhance security. Afterward, in the diffusion phase, the chaotic sequence and the DNA sequence operators are utilized to alter the gray surface on the image pixels. In the fourth phase, based on a genetic algorithm, the optimization process is performed following the encryption.

Phase 1: Secret key generation

To improve the security against attacks, A 256-bit key

was utilized. To create pseudo-random figures using the LS2map function, we needed a starting value (x0), obtained using a 256-bit key. Using the input data (image) the SHA-256 algorithm produces a unique 256-bit hash key, through this, a direct link is created that connects the input data and the generated key (image). In other word, a unique identifier is assigned to each input data.

As depicted in Figure 4, to generate secrete key, the proposed algorithm involves the initial conversion of the input image to a 64-character hexadecimal figure based on the SHA-256. Subsequently, to improve the level of security, using SHA-256, the same 64-character string is further changed into a new 64-character hexadecimal number. Afterward, the new secret key is turned into binary form. For instance, with an image of Rayan, it is first converted to "7d23f64f7759bfe39acc056b0bc7c85aa4d46bf5fefee15b54c416526b9d02cc" using the SHA-256 algorithm.

Then, it is further converted to "6445a9943240fe983d058158a2b07e4bf3aad8b842914c18ba9f9b48cd9930dc" before being transformed into binary form. The obtained key id demonstrated as "bi," where "bi" indicates the ith bit of the key. The 256 bits are divided into 32 8-bit blocks (Eq. 9).

Key= {K_1,K_2,…,K_32 } (9)

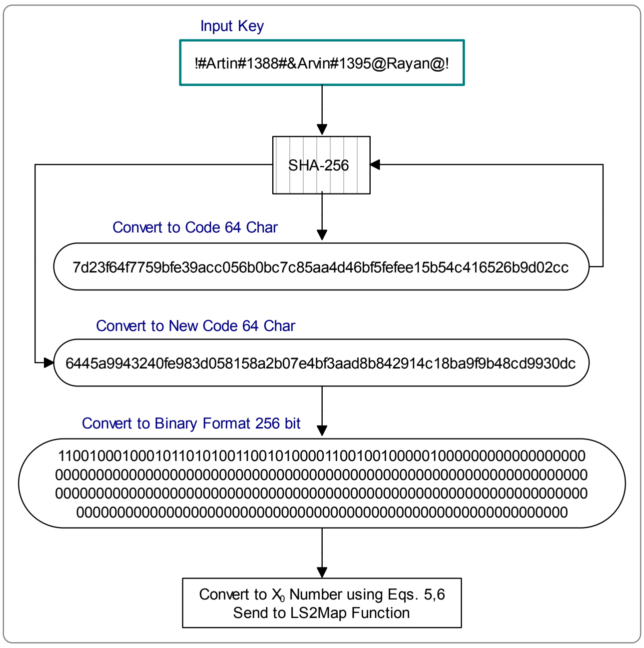


Fig.4. Generating the secret key

Formula 10 is employed to obtain x0. Which ⊕ represents the XOR operator.

$$X_0 = \left. \begin{array}{c} \left(\sum_{i=1}^{8} K_i + (K_{17} \oplus \ldots \oplus K_{24})\right) + \\ \left(\sum_{i=9}^{16} K_i + (K_{25} \oplus \ldots \oplus K_{32})\right) \end{array} \right/ 2^{12} \qquad (10)$$

Phase 2: permutation

Step 1. Level one blocking:

This step involves dividing the main image into blocks (64 to 64×64 pixel blocks for 512×512 images and 64 to 32×32 pixel blocks for 256×256 images).

Step 2: Move the level one blocks based on the magic square algorithm.

Section 3-2. Figure 6 displays the changes in the blocks of the original image.

Step 3. Level 2 Blocking:

The blocks dispositioned from step 2 are chosen through the LS2map chaos (Eq. 3) to achieve a random number (X_i) in the [1..64] based Eq. 11. The blocks are then is divided 8x8 blocks (64 to 8×8 pixel blocks 512x512 images and 16 to 8x8 pixel blocks 256x256 images).

Select_Block=Round(X_i×64)+1 (11)

Step 4. Blocks selected based on the rate of permutation:

A random number (X_i) is generated within the range of 4 to 8 for 256 × 256 images, and within the range of 16 to 32 for the same image size. The LS2 map function is utilized for selecting blocks for the rate of image pixel permutation (Permute_Rate).

Select_B_256=Round(X_{i+1} ×8)+4 (12)

Select_B_512=Round(X_{i+1} ×32)+16 (13)

Step 5. Moving the second level blocks using the magic square algorithm:

To rearrange the pixels inside the selected blocks, we used the magic square algorithm (see section 3-2).

Step 6: Repeat steps 4 and 5 for all selected blocks, based on the number of hash rates.

The difference lies in using the values of X_i and X_{i+1} for every image, starting from the last value created by the image earlier.

Phase 3: Diffusion

Step 1. Convert the image into a one-dimensional (1D) array:

As shown in Figure 4, at this stage, in order to convert the image obtained from the previous stage (permutation) into a 1D array, first, Row i+1 of the image is placed at the end of Row i, and this action is repeated for all the rows of the image. The 2D N × M array is converted into a 1D array with N × M elements in the following form:

$\{P_1, P_2, \ldots P_{N\times M}\}$. In addition, Equations 14 and 15 are used to calculate the number of the row and column of Pixel (P_i) in the 2D array.

$$\text{Row} = \left\lceil \frac{P_i}{N} \right\rceil \qquad (14)$$

$$Column = \begin{cases} (P_i \ mod \ M) & if \ (P_i \ mod \ M) \neq 0 \\ M & if \ (P_i \ mod \ M) = 0 \end{cases} \qquad (15)$$

Step 2. Conversion into a DNA sequence:

At this stage, the five sequential numbers [X_i, ..., X_{i+4}] of the chaotic function "LS2 map" are used to encrypt both pixels, in the following way: X_i is used to generate a random number in the interval [1..8] based on Equation 16 in order to select DNA rules from Table 1, X_{i+1} and X_{i+2} to generate a random number in the interval [1..256] based on Equation 17 in order to select two bits from the key to select the biomolecules operators (XOR, ADD), and X_{i+3} and X_{i+4} to generate two random numbers in the interval [0..255] based on Equation 18 in order to select two numbers from the chaotic function in order to apply the selected operator.

$$\text{Select}_{\text{Rule}} = \text{Round}(X_i \times 7) + 1 \qquad (16)$$

$$\text{Select}_{\text{Operator}} = \text{Round}(X_{i+1} \text{ and } X_{i+2} \times 255) + 1 \qquad (17)$$

$$\text{Select}_{\text{Number}} = \text{Round}(X_{i+3} \text{ and } X_{i+4} \times 255) \qquad (18)$$

Step 3. Replacement of pixels:

At each stage, two pixels are selected from the 1D array and converted into a DNA sequence using Equation 12 and Table 1.

As shown in Figure 6, two low-value nucleotides from the selected pixels are replaced by one another.

The two numbers, selected from the chaotic function based on Equation 14, are converted into a DNA sequence using Equation 12 and Table 1.

At this stage, the XOR operator is used to the two bits selected from the secret key. If the result of the XOR operator on the two bits selected from the key is zero, the XOR operator will be applied to the two pixels selected from the 1D array and the two numbers selected from the chaotic function. Otherwise, (if the result of the XOR operator on the two bits selected from the secret key is 1,) the XNOR operator will be applied to the two pixels selected from the 1D array and the two numbers selected from the chaotic function. The process is repeated for all the pixels of the image.

Step 4. The step three output will again be turned into a binary form based on Table 1, upon completion of the encryption process and then into a decimal form, and then into a 2D array using Equations 14 and 15. Figure 5 shows an example of the encryption process of two pixels selected from the image in the diffusion phase.

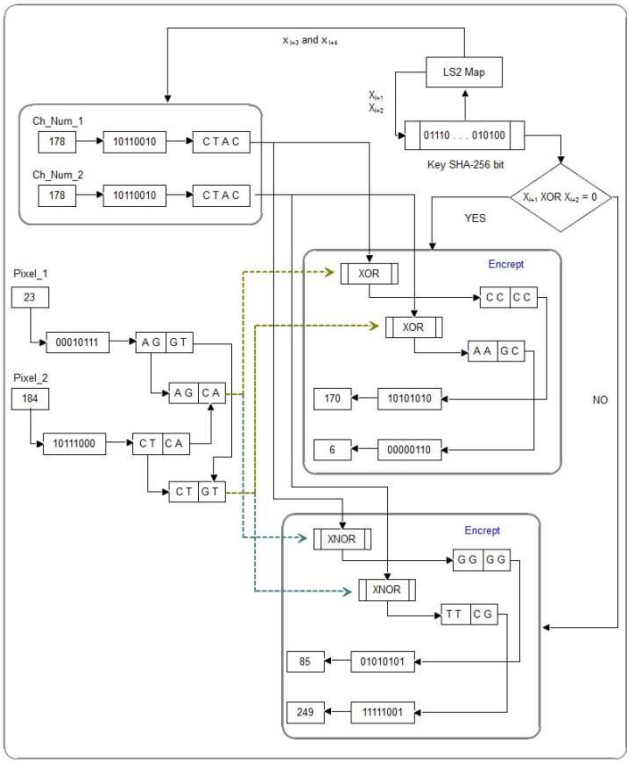


Fig. 5. The process of encryption two selected pixels of an image at the diffusion step

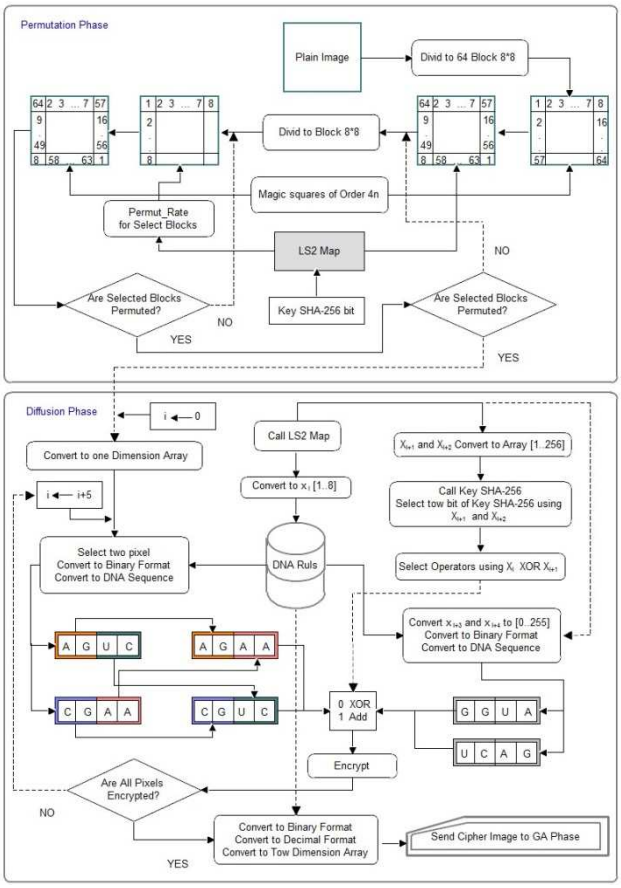


Fig. 6. Proposed Model Encryption Process

Phase 4: GA

Following diffusion and permutation process, and based on the genetic algorithm rules outlined earlier, the optimization takes place for achieved an image encrypted with greater resistance in comparison to the original one, taking into account the criterion of entropy criterion for the evaluation function:

a) Creation of the initial population:

Using 100 images, the starting population is formed and used in the diffusion. The distinction lies in using the chaotic function values, which are obtained from the chaotic function for each image. This process starts with the final value given by the image in the last step. In addition, each image is assigned with three integers depending on the duration of encryption.

b) Selection operator:

The operator indicates the chromosome that can survival in the next generation. That is, individuals with a superior chromosome have a higher survival chance. This actually happens in the process of evolution. Still, in some cases, using a mutation operator or crossover operator to a wrong chromosome may lead to the correct chromosome. Here, the top-quality image with the maximum entropy are adopted based on a roulette wheel to undergo crossover operator and mutations operator.

c) Crossover operator:

After encryption (as depicted in Figure 7), images are categorized into four equal sections following selection and arranged in a line side by side.

Here, a figure between 1 and 3 is selected through a random function.

When one is the obtained random number, the one-point crossover method is utilized. When two is the number randomly generated, the two-point crossover method is selected; and when three is the number randomly generated, the multi-point crossover method is employed. Moreover, with crossover operator employed on the image, three numbers are inserted in the three-byte encryption table. For the decryption and determine the images and operators utilized for image encryption this is essential.

d) Mutation operator:

Following the crossover operator, the mutation happens prior to the final evaluation in the search space. Actually, the mutation randomly finds the proper chromosome for change. Here, a mutation happens by random selection of a new encryption image to replace the original image using the operator.

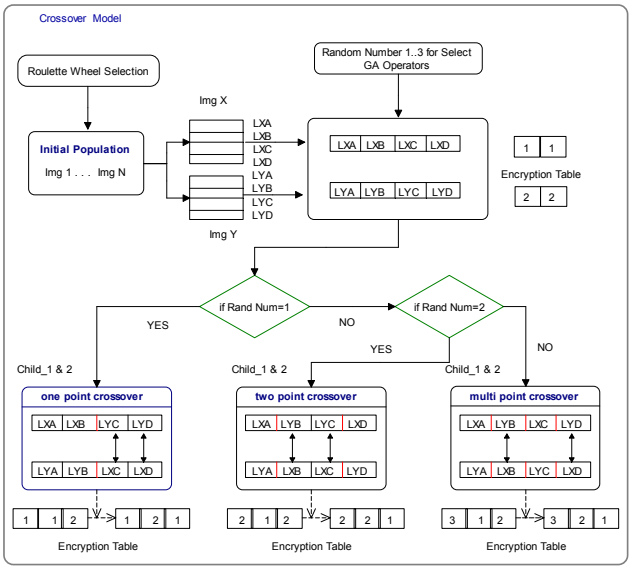


Fig. 7. Crossover of the proposed model

4. Simulation Results

Here, the performance and security of the introduced model are analyzed and evaluated through brute-force analysis, statistical analysis, information entropy analysis, differential attacks, and convergence analysis. The evaluations were carried out in MATLAB software (2.50 GHz Dual Core, 4GB memory, and Windows 7). The analysis was performed on different image with of 256 × 256 and 512 × 512 dimensions, which were developed from the USC-SIPI database based on Figure 8.

Fig. 8. Test image

4.1. Statistical Analysis

To be an efficient algorithm for encryption, the output must be resistive to statistical attacks. We generated the histogram of image to carry out statistical analysis. Additionally, using images retrieved the USC-SIPI database we analyzed adjacent vertical, horizontal, and diagonal pixels correlation.

4.1.1. Histogram Analysis

The pixel elements distribution in an image is illustrated

by image histogram. The algorithm of encryption must resist statistical attacks. Actually, with an increase in uniformity of the histography of the image of the encryption algorithm, the rate of statistical attacks decreases. The proposed model conducted analyses on the results of various experiments based on histogram analysis to check the statistical distributions. Figure 9 depicts the histograms of the images before employed and after employing the proposed model. Clearly, the histogram of the encrypted image appears uniform and is not identical to the histogram of the original image. This prevents hackers from accessing valuable information, reducing the likelihood of successful statistical attacks. This illustrates the effectiveness of the model proposed here.

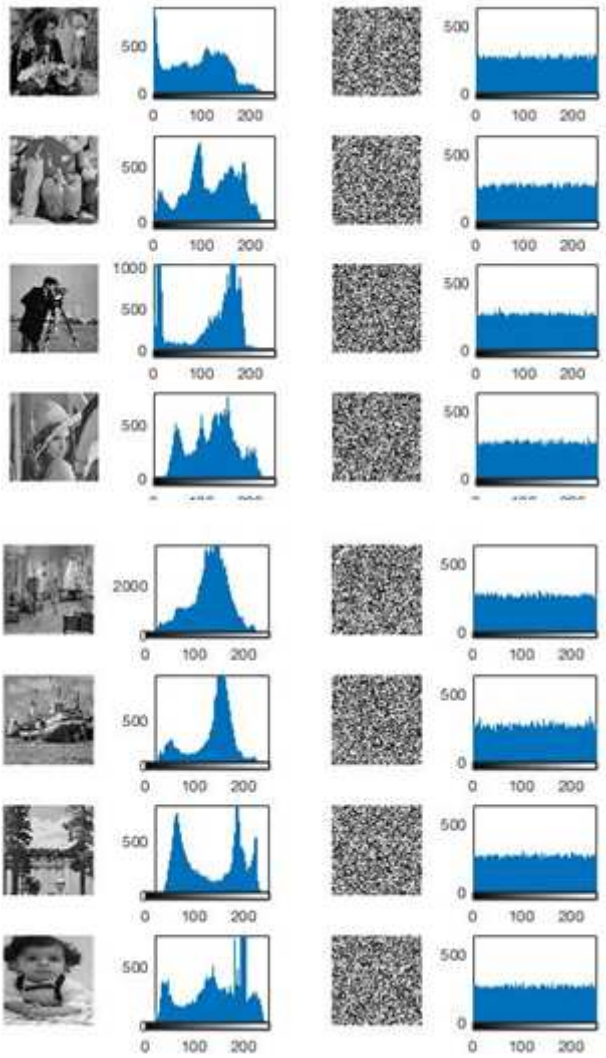


Fig. 9. From left to right in each row, simple image, simple image histogram, encrypted image and encrypted image histogram

4.1.2. Analysis of Correlation Coefficients

Because of the interdependence of adjacent pixels in digital images, a robust cryptographic techniques requires generating encrypted images that minimize the vertical, diagonal, and horizontal correlations among the pixels. As a statistical index, the correlation coefficient is utilized to measure the two variables relationship, returning a number within the range of -1 to 1. With a number closer to zero, the variations of the two variables become more independent, and as the figure merges to -1 or 1, the interdependence of variations increases. The coefficient of correlation is obtained as follows [7].

$$r_{xy}=\frac{|\mathrm{cov}(x,y)|}{\sqrt{D(x)}\times\sqrt{D(y)}} \qquad (19)$$

$$E(x)=\frac{1}{N}\sum_{i=1}^{N}x_i \qquad (20)$$

$$D(x)=\frac{1}{N}\sum_{i=1}^{N}(x_i-E(x))^2 \qquad (21)$$

$$\mathrm{cov}(x,y)=\frac{1}{N}\sum_{i=1}^{N}(x_i-E(x))(y_i-E(y)) \qquad (22)$$

where, Y and X represent the brightness of two pixels that are next to each other. In terms of similarity of pixel, the two adjacent pixels become more similar. When plot points are closer to the original diameter. The objective of cryptography to minimize the similarity of the encrypted images to decrease the likelihood of unauthorized image access by analyzing the resemblance among pixels.

The correlation of 8192 pairs of adjacent horizontal, vertical, and diagonal pixels extracted from the database of the image using Equation 19 is listed in Table 2.

Table 3. The coefficient of correlation of adjacent pixels (vertical, diagonal, and horizontal)

Correlation of pixels					
		Cameraman	Forest	Lena	Rayan
256 × 256	Horizontal	-0.0098	-0.0096	-0.0178	-0.0106
	Vertical	-0.0003	0.0005	0.0005	-0.0107
	Diagonal	0.0085	0.0122	-0.0101	-0.0029
512 × 512	Horizontal	0.0130	-0.0191	-0.0045	0.0062
	Vertical	0.0101	0.0123	0.0008	-0.0057
	Diagonal	-0.0211	0.0118	0.0013	0.0010
		peppers	Living room	Boat	Painter
256 × 256	Horizontal	0.0043	0.0064	0.0090	0.0021
	Vertical	-0.0048	-0.0193	0.0174	0.0131
	Diagonal	0.0074	-0.0125	0.0025	-0.0019
512 × 512	Horizontal	0.0147	0.0040	0.0023	-0.0094
	Vertical	-0.0016	0.0168	-0.0045	0.0047
	Diagonal	0.0087	-0.0062	0.0062	0.0022

The distribution of gray surface for the two diagonally, horizontally, and vertically adjacent pixels of the Lena image is shown in Figure 10. As depicted in Figure 10, the pixels are uniformly distributed and independence of the successive pixels is confirmed. Therefore, the findings

indicate that the original image differs from the encrypted image in terms of adjacent pixels.

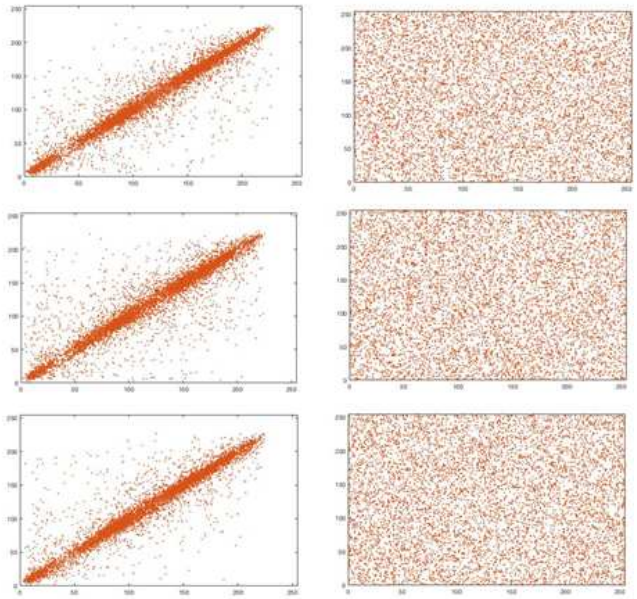


Fig. 10. Distribution of gray surface for two adjacent pixels (top to bottom) vertically, diagonally, and horizontally on the Lena image prior to (left diagram) and following encryption (right diagram)

4.2. Information Entropy

Shannon entropy, also known as information entropy, is a prominent characteristic of randomness. It is a mathematical theory for storing and transferring data, which was developed by Claude E. Shannon in 1949. Equation 23 is a well-known formulas for calculating entropy [1].

$$H(s)=\sum_{i=0}^{2^M-1} P(s_i)\log_2\frac{1}{P(s_i)} \qquad (23)$$

Where N refers to the number of gray levels in the image (256 in 8-bit images) and indicates the occurrence chance of the i-th gray level. In completely random images, this figure is eight, which is an ideal figure. The proximity of these figures 8 in the model proposed here indicates a negligible leakage of information in the image decoding. Therefore, the encryption model proposed here has a highly resistive to entropy attack. The encrypted images entropy is given in Table 4.

Table 4. Encrypted images entropy on image database

Table 4. Encrypted images entropy on image database

Image Size	Cameraman	Forest	Lena	Rayan
256 × 256	**7.9977**	**7.9976**	**7.9976**	**7.9970**
512 × 512	**7.9993**	**7.9994**	**7.9993**	**7.9993**
Image Size	Peppers	Living room	Boat	Painter
256 × 256	**7.9977**	**7.9976**	**7.9976**	**7.9970**
512 × 512	**7.9993**	**7.9994**	**7.9993**	**7.9993**

4.3. Brute-Force Attack

The sensitivity of the secret key and its state space size are examined to prevent brute-force attacks.

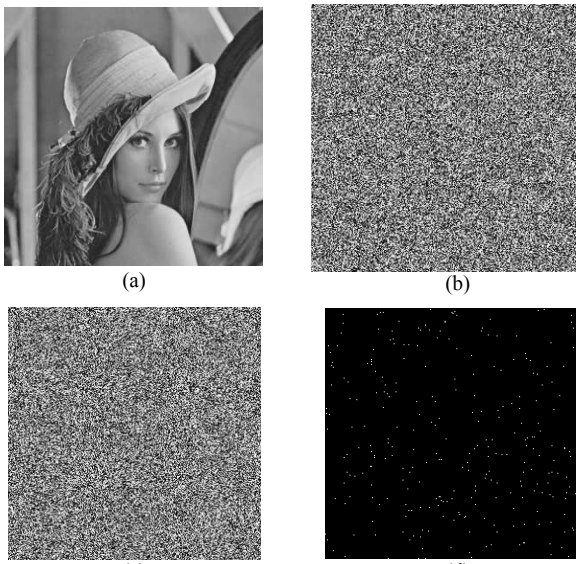
4.3.1. Secret Key Sensitivity Analysis

A main aspect of security of a robust encryption algorithm is the password sensitivity. The sensitivity level of the chaotic systems to the initial state and parameters is high. With a group of possible parameter states, selecting two closely related initial parameters can take the system into two diverse ways. Thus, chaotic systems are suitable candidates for use in cryptography. Therefore and to assess the proposed model sensitivity to the encryption key (Fig. 11), the Lena image (256 * 256) in Figure 11a was encrypted through a 256-bit hidden key "!#Artin#1388#&Arvin#1395@Rayan@!". (11b). Then, it is encrypted once more using the key "!#Artin#1388#&Brvin#1395@Rayan@!""; still, by randomly changing a bit (11 c). Figure 11d (highlighting the differences between images 11b and 11c) clearly illustrates that the two images are encoded with white pixels at the same gray level. As pictured in Figure 11d, the model has sensitivity to slightest changes in initial key, which ensures the model's resilience to a comprehensive search attack. Table 4 lists the findings of the analysis of the secret key applied to the database images.

4.3.2. Analysis of Secret Key Mode Space

The total keys usable in cryptography is given by the key state space. With a bigger key state space, the performance of encryption increases along with the resistance to cryptographic attacks like unrestrained attacks.

Here, a 256-bit key is employed as the initial value for the ls2map chaotic function, resulting in a key state space of 2^{256} states. Due to the big state space, the model proposed here demonstrated good resistance to brute force attacks.



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

Fig. 11. a. image of Lena, b. encrypted image with a 256-bit secret key, c. encrypted image with the same key and by changing one bit, d. image differences between 10b and 10c

Table 5. Sensitivity to the secret key in the model on the image database

Image Size	Cameraman	Forest	Lena	Rayan
256 × 256	**99.58%**	**99.60%**	**99.61%**	**99.77%**
512 × 512	**99.62%**	**99.60%**	**99.62%**	**99.63%**
Image Size	Peppers	Living room	Boat	Painter
256 × 256	**99.62%**	**99.59%**	**99.66%**	**99.63%**
512 × 512	**99.62%**	**99.61%**	**99.61%**	**99.61%**

4.4. Differential Attack

In image encryption, the use of a differential attack is a significant and necessary criterion for evaluating the security level. In practice, a Differential Attack aim is to indicate if altering a pixel in the original image can influence image after encryption.

4.4.1. Evaluation of NPCR and UACI Benchmark

The influence of altering a single pixel of the original image after encryption can be assessed by two criteria: NPCR and UACI. NPCR is given as the pixel change rate in the image encrypted based on the change of one pixel in the original image. Additionally, UACI is the average of such changes, determined as follows [30].

$$NPCR=\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)}{M\times N}\times 100\% \qquad (24)$$

$$UACI=\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} |C_1(i,j)-C_2(i,j)|}{255\times M\times N}\times 100\% \qquad (25)$$

Where W refers to the width of the images and H refers to length of the images; additionally, I1 and I2 are two images encrypted on the basis of two images with one pixel difference; and D is obtained as follows:

$$D(i,j)=\begin{cases}1 & \text{if } C_1(i,j)=C_2(i,j)\\ 0 & \text{otherwise}\end{cases} \qquad (26)$$

The values based on Tables 6 and 7 clearly indicate that the model also has an acceptable resistance to differential attacks, meaning that a single pixel change in the input image will result in a significant variation in the output.

Table 6. NPCR NPCR outputs of the model on the images

Image Size		Cameraman	Forest	Lena	Rayan
256×256	Average	0.995962	0.996145	0.995962	0.996104
	Best	0.996078	0.996323	0.996094	0.996338
512×512	Average	0.996174	0.996175	0.996038	0.996183
	Best	0.996197	0.996269	0.996098	0.996288
Image Size		peppers	Living room	Boat	Painter
256×256	Average	0.996084	0.996099	0.996043	0.996175
	Best	0.996292	0.996231	0.996216	0.996175
512×512	Average	0.996086	0.996062	0.996091	0.996188
	Best	0.996208	0.996294	0.996223	0.996201

Table 7. UACI results of proposed model on image database

Image Size		Cameraman	Forest	Lena	Rayan
256×256	Average	0.333867	0.334893	0.333888	0.334350
	Best	0.334233	0.335576	0.335081	0.335352
512×512	Average	0.334577	0.334666	0.334754	0.334694
	Best	0.335111	0.334830	0.335243	0.335185
Image Size		peppers	Living room	Boat	Painter
256×256	Average	0.334140	0.334459	0.335302	0.334343
	Best	0.334990	0.335960	0.336555	0.335406
512×512	Average	0.334736	0.333551	0.334434	0.334265
	Best	0.335001	0.334656	0.334794	0.335435

4.5. Comparisons

Table 8 shows the model performance in comparison to [23, 27, 30, 1, 19, 6, 7, 28]. The value of entropy of the model can be improved significantly by incorporating the LS2 Map chaos function and integrating it with DNA biomolecule operators, and utilizing genetic algorithms. As listed in Table 8, the model demonstrates superior performance for the boat image compared to other

references and is competitive with a few references in terms of alternative criteria.

Table 8. The proposed model function compared to other methods

	Entropy	Correlation Coefficient			NPCR (%)	UACI (%)
		V	H	D		
Ref. [23]	7.9965	0.0074	0.0044	0.0081	0.995890	0.335002
Ref. [27]	7.9976	0.0015	0.0007	-0.0007	0.996200	0.336900
Ref. [30]	7.9986	0.0104	0.0167	0.0074	0.971029	0.328471
Ref. [1]	7.9992	0.0103	-0.0069	-0.0011	0.996240	0.334408
Ref. [19]	7.9986	0.0167	0.0104	0.0074	0.971029	0.328471
Ref. [6]	--	0.0215	0.0154	0.0081	--	--
Ref. [7]	7.9993	0.0408	0.0123	0.0040	0.996129	0.333711
Ref. [28]	7.9965	0.0074	0.0044	0.0081	0.995890	0.335002
Proposed Model	7.9994	-0.0045	0.0023	0.0062	0.996294	0.334656

5. Conclusion

An evolutionary model based on a combination of DNA biomolecule operators and the LS2 Map chaos function was proposed for image encryption. The chaos function dependence on the 256-bit secret key and the MSC algorithm in for permutation, combined with the utilization of DNA biomolecule operators and genetic algorithms, contribute to the advantages of the proposed model. These advantages include high entropy value, encrypted image histogram uniformity, a significant decline in the adjacent pixels correlation in the image of encryption, and a large key mode space. All these factors collectively support the high efficiency of the presented model in encrypted applications. Moreover, the model has demonstrated strong resilience against different attack types such as code deciphering attacks, statistical attacks, and brute-force attacks.

References

[1] Abbasi, A. A., Mazinani, M., & Hosseini, R. (2020). Chaotic evolutionary-based image ncryption using RNA codons and amino acid truth table. Optics & Laser Technology, 132, 106465.

[2] Enayatifar, R., Abdullah, A. H., Isnin, I. F., Altameem, A., & Lee, M. (2017). Image encryption using a synchronous permutation-diffusion technique. Optics and Lasers in Engineering, 90, 146-154.

[3] Gulshan, K., Praveen, P., Rahul, S., & Kumar, R. M. (2016). Chaotic image encryption technique based on IDEA and discrete wavelet transformation. Indian J. Sci. Technol, 9(15), 71871.

[4] Hu, T., Liu, Y., Gong, L.-H., & Ouyang, C.-J. (2017). An image encryption scheme combining chaos with cycle operation for DNA sequences. Nonlinear Dynamics, 87(1), 51-66.

[5] Ibrahim, Y., Khalifa, F., Mohamed, M. A., & Samrah, A. S. (2020). A New Image Encryption Scheme Based on Hybrid Chaotic Maps. Complexity, 2020.

[6] Abdullah, A. H., Enayatifar, R., & Lee, M. (2012). A hybrid genetic algorithm and chaotic function model for image encryption. AEU-International Journal of Electronics and Communications, 66(10), 806-816.

[7] Abbasi, A. A., Mazinani, M., & Hosseini, R. (2020). Evolutionary-based image encryption using biomolecules operators and non-coupled map lattice. Optik, 219, 164949.

[8] Chenaghlu, M. A., & Khasmakhi, N. N. A fast and secure keyed hash function based on coupled chaotic maps for crypto-currencies.

[9] Abdullah, D., Rahim, R., Siahaan, A. P. U., Ulva, A. F., Fitri, Z., Malahayati, M., & Harun, H. (2018). Super-encryption cryptography with IDEA and WAKE algorithm. Paper presented at the J. Phys. Conf. Ser.

[10] Luo, Y., Yu, J., Lai, W., & Liu, L. (2019). A novel chaotic image encryption algorithm based on improved baker map and logistic map. Multimedia Tools and Applications, 78(15), 22023-

[11] Al-Mashhadi, H. M., & Abduljaleel, I. Q. (2017). Color image encryption using chaotic maps, triangular scrambling, with DNA sequences. Paper presented at the 2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT).

[12] Alabaichi, A. M. (2016). Color image encryption using 3D chaotic map with AES key dependent S-Box. International Journal of Computer Science and Network Security (IJCSNS), 16(10), 105-115.

[13] Ye, G., Pan, C., Huang, X., & Mei, Q. (2018). An efficient pixel-level chaotic image encryption algorithm. Nonlinear Dynamics, 94(1), 745-756.

[14] Yin, Q., Cao, B., Li, X., Wang, B., Zhang, Q., & Wei, X. (2020). An Intelligent Optimization Algorithm for Constructing a DNA Storage Code: NOL-HHO.International journal of molecular sciences,212191.

[15] Belazi, A., Abd El-Latif, A. A., Diaconu, A.-V., Rhouma, R., & Belghith, S. (2017). Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms. Optics and Lasers in Engineering, 88, 37-50.

[16] Li, Y., Wang, C., & Chen, H. (2017). A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Optics and Lasers in Engineering, 90, 238-246.

[17] Chai, X., Chen, Y., & Broyde, L. (2017). A novel chaos-based image encryption algorithm using DNA sequence operations. Optics and Lasers in Engineering, 88, 197-213.

[18] El-Zoghdy, S. F., Nada, Y. A., & Abdo, A. (2011). How good is the DES algorithm in image ciphering? International Journal of Advanced Networking and Applications, 2(5), 796-803.

[19] Enayatifar, R., Abdullah, A. H., & Isnin, I. F. (2014). Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. Optics and Lasers in Engineering, 56, 83-93.

[20] Wang, X., Wang, Y., Zhu, X., & 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.

[21] Joshy, A., Baby, K. A., Padma, S., & Fasila, K. (2017). Text to image encryption technique using RGB substitution and AES. Paper presented at the 2017 International Conference on Inventive Computing and Informatics (ICICI).

[22] Wang, Y., Lei, P., Yang, H., & Cao, H. (2015). Security analysis on a color image encryption based on DNA encoding and chaos map. Computers & Electrical Engineering, 46, 433-446. 22043.

[23] Tsafack, N., Sankar, S., Abd-El-Atty, B., Kengne, J., Jithin, K. C., Belazi, A., ... & Abd El-Latif, A. A. (2020). A new chaotic map with dynamic analysis and encryption application in internet of health things. IEEE Access, 8, 137731-137744.

[24] Som, S., Mitra, A., Palit, S., & Chaudhuri, B. (2019). A selective bitplane image encryption scheme using chaotic maps. Multimedia Tools and Applications, 78(8), 10373-10400.

[25] ur Rehman, A., Liao, X., Ashraf, R., Ullah, S., & Wang, H. (2018). A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2. Optik, 159, 348-367.

[26] Wang, X., Feng, L., Li, R., & Zhang, F. (2019). A fast image encryption algorithm based on non-adjacent dynamically coupled map lattice model. Nonlinear Dynamics, 95(4), 2797-2824.

[27] Nestor, T., De Dieu, N. J., Jacques, K., Yves, E. J., Iliyasu, A. M., El-Latif, A., & Ahmed, A. (2020). A multidimensional hyperjerk oscillator: Dynamics analysis, analogue and embedded systems implementation, and its application as a cryptosystem. Sensors, 20(1), 83.

[28] Maqsood Mahmuda, Atta-ur-Rahmanb, Malrey Leec, Jae-Young Choi, Evolutionary-based image encryption using RNA codons truth table, Optics and Laser Technology 121 (2020) 105818.

[29] Yadollahi, M., Enayatifar, R., Nematzadeh, H., Lee, M., & Choi, J.-Y. (2020). A novel image security technique based on nucleic acid concepts. Journal of Information Security and Applications, 53, 102505.

[30] Nematzadeh, H., Enayatifar, R., Yadollahi, M., Lee, M., & Jeong, G. (2020). Binary search tree image encryption with DNA. Optik, 202, 163505.

[31] Tahbaz, M., Shirgahi, H., Yamaghani, M.R. (2023). Evolutionary-based image encryption using Magic Square Chaotic algorithm and RNA codons truth table. Multimedia Tools and Applications, 83, 503–526. https://doi.org/10.1007/s11042-023-15677-3