EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

Efficient DNA-Based Logistic Mapping Algorithm for Color Image Encryption

Aya Goudjil[1,3,*], Aicha Benyoucef[2,3,†], M'Hamed Hamadouche[2,3,‡], Mohamed Amine Riahla[2,3,§]

[1]Computer Science Department, Faculty of Sciences, M'Hamed Bougara University of Boumerdes , Railway Station Road, Boumerdes, Algeria
[2]Electrical Engineering Department, Faculty of Technology, M'Hamed Bougara University of Boumerdes, Frantz fanon City, Boumerdes, Algeria
[3]Computer Laboratory for Optimization Modeling and Electronic Systems, M'Hamed Bougara University of Boumerdes, Railway Station Road, Boumerdes, Algeria

Abstract

INTRODUCTION: Color images hold significant information and are widely used in diverse domains. The protection of these images against unauthorized access over the internet is a necessity that relies on encryption techniques. However, traditional encryption methods face challenges with the increasing capability and efficiency of quantum computing to solve complex problems. DNA cryptography is a promising field in information security, utilizing DNA molecules with massive parallelism and vast storage capacity to encode and decode information. While chaotic systems have been widely used in encryption due to their sensitivity to initial conditions and parameter values, resulting in unpredictability and significant variation. Exploiting the characteristics of DNA cryptography and chaotic systems is a promising alternative for securing data. Nevertheless, current methodologies exhibit limitations such as a small key space and weak resistance to differential attacks.
OBJECTIVES: This paper addresses these gaps by proposing an RGB image encryption algorithm based on DNA cryptography and a 1D logistic map.
METHODS: The proposed method randomly generates a DNA encoding/decoding table to generate the row-column permutation of the image. After the permutation of the image, the logistic map is used to generate three keys for RGB channels and seven DNA encoding-decoding rules, three are used to encode the keys into DNA sequence, the second three to encode the image RGB channel, and the last to perform the DNA-XOR operation. Finally, decode the result into integers using the DNA encoding/decoding table and generate the encrypted image.
RESULTS: The analysis of the proposed technique demonstrates significant robustness against various attacks, as evidenced by metrics such as a key space exceeding 2^{100}, an average NPCR of 99.613667%, an average UACI of 50.273742%, an entropy value approaching 8, and a strong key sensitivity.
CONCLUSION: These results validate its capacity to effectively resist differential, brute-force, and statistical attacks.

Keywords: Cryptography, DNA cryptography, Encoding, Decoding, Image security, Image encryption, Color images, Digital chaotic map, Logistic map.

Received on 29 07 2024; accepted on 05 02 2025; published on 13 02 2025

doi:10.4108/eetsis.6800

*Corresponding author. Email: e-mail: a.goudjil@univ-boumerdes.dz

1. Introduction

Color images are crucial in various domains, including healthcare, digital forensics, remote sensing, commercial applications, and communication. Given their importance and the sensitive data they contain,

it is essential to protect them against unauthorized access using strong encryption methods to ensure their confidentiality, integrity, and authenticity across these diverse fields. Over the decades, traditional encryption methods such as DES, AES, and RSA have been widely used and tested for securing data [1]. However, quantum computing poses a significant challenge to these traditional methods by exploiting its capability to solve complex problems efficiently. Color images, with their multiple layers of data compared to grayscale images, require more intensive computation and careful implementation of these conventional methods. To address these weaknesses, combining DNA cryptography with chaotic encryption offers a promising solution due to their inherent complexity and sensitivity to initial conditions. Several encryption strategies for color images that employ DNA cryptography and chaotic functions have been proposed. [2] Develops a new image encryption technique utilizing DNA coding, DNA operations, and a two-dimensional logistic map for securing remote-sensing images. The logistic map determines the DNA encoding rules and DNA mask. The plain image is encoded using these DNA rules, followed by a DNA addition operation with the DNA mask. The output is used to calculate the probability of the DNA bases, establishing the new initial condition of the logistic map and generating a random sequence. This sequence is sorted in ascending order to create an index matrix used for rearranging the image pixels and DNA bases, resulting in the encrypted image. While this technique demonstrates acceptable encryption speed and sensitivity to small changes in the image or key, it does not experiment with larger remote-sensing images and relies on the chaotic map. This becomes a weakness if the initial parameters are exposed or predicted. [3] The paper proposes a chaotic color image encryption scheme based on DNA-coding calculations and arithmetic over the Galois field. It introduces SLM (Sine-logistic map), CLM (Chebyshev-logistic map), and SCM (Sine-Chebyshev map), which are modified and improved versions of the classical 1D chaotic maps (Logistic, Sine, and Chebyshev). The initial conditions of SLM, CLM, and SCM are updated based on the odd pixels of the RGB components of the original image. Using predefined equations, four operations are performed. First, pixel-level scrambling of the RGB channels of the original image is applied, generating three matrices. Each matrix is divided into smaller, zero-padded square matrices and encoded using a specific DNA rule. Second, DNA calculations including addition, subtraction, and XOR operations are performed between the encoded RGB components and the three matrices. Third, the decoding rules are determined, and used to decode the result of the DNA calculations. Finally, multiplication on the GF(17) is implemented to extract the encrypted image. This scheme has excellent performance, though its the encryption time must be reduced to be suitable for real-time communication. [4] An effective encryption algorithm for both grayscale and color images introduces an improved 2D logistic sine chaotic map (2D-LSMM) to determine the DNA encoding and operation rules. The 2D-LSMM is based on logistic and sine chaotic mappings. The initial conditions of the chaotic maps are calculated from the pixel values of the original image, generating two random sequences from the logistic and sine maps. The derived logistic sequence is reshaped to match the image dimensions and used to select the DNA operation (addition, subtraction, or XOR), which is applied between the encoded image and LSMM sequence using the selected encoding rules from the sine sequence. The output is then decoded using DNA decoding rules determined by the logistic sequence and saved as the encrypted image. This scheme achieves proper encryption and resists different attacks, but it lacks the permutation process and relies only on the diffusion, limiting the effectiveness of the scheme. [5] A proposed image encryption method leverages DNA encryption, linear feedback shift registers (LFSR), and 3D chaotic maps. The encryption process begins by extracting the RGB components of the image, dividing each one into two equal parts, and converting these to 1D arrays. These arrays are then distributed using a key generated by the LFSR and encrypted with one of four selected DNA rules. The first key is normalized to values between 0 and 255 using the modulo operation, serving as a second key, and encoded using the same DNA rule as the first. An XOR operation is performed between the second key and the two extracted parts of the image using the four chosen DNA rules. The final result is achieved by applying the XOR operation between the RGB components and the three keys generated by the 3D chaotic map. The efficiency of this method is validated through statistical tests and a comparison with selected research work, though further exploration of chaotic maps and optimization is necessary to reduce computational complexity. A 3-stage image cryptosystem is developed and proposed in [6], leveraging DNA cryptography, standard and Tan logistic maps, and an S-box. The first stage converts the image to a 1D bit-stream and creates a sequence based on the Tan logistic map, then performs a DNA addition operation after converting both to DNA sequences. In the second stage, the DNA addition result is substituted using an S-box produced by the Lorenz system and then subtracted from 256. The final result is XORed with a sequence derived from the standard logistic map, and the encrypted image is reconstructed from it. This cryptosystem shows good resistance to various attacks, though its security could be enhanced by

optimizing the S-box design, utilizing discrete chaotic maps, and exploring higher-dimensional ones. [7] A novel color image encryption algorithm adopts a 3D chaotic system, a bidirectional spiral transformation, and dynamic DNA operations. The researchers define a new 3D chaotic system by incorporating the cosine function into the Spoot B system and introduce a look-up table method to accelerate DNA operations. For encryption, the RGB components of the image are extracted and permuted using the bidirectional spiral transformation. The hash value of the original image is computed using the SHA-256 algorithm, combined with external keys to generate control parameters and initial values for the chaotic map. The resulting chaotic sequence is employed for dynamic DNA encoding using a set of encoding rules and DNA operations, including XOR, XNOR, left cyclic shift, and right cyclic shift. Analysis of various security aspects demonstrates that the proposed algorithm is suitable for real-time secure communication systems. On the other hand, its performance may vary depending on the hardware. [8] An encryption technique merges SHA-256/384, Lorenz and Rossler hyper-chaotic maps, and DNA cryptography. The SHA-256/384 hash of the image is calculated to generate the initial parameters of the chaotic maps. The Lorenz hyper-chaotic system scrambles the image channels to produce confusion, which are then converted to DNA using DNA encoding and various DNA rules. A DNA addition operation is performed on these encoded channels, generating new RGB channels that are then XORed with Rossler hyper-chaotic sequences. Finally, DNA decoding is applied to the XOR results to reconstruct the encrypted image. Despite the strengths of the proposed method, its reliance on the hash function poses a potential weakness. If the same image is repeatedly encrypted, along with the use of static DNA encoding/decoding rules could compromise the security of the encryption. A color image encryption method based on DNA encryption and the 2D logistic chaotic map is proposed in [9]. The encryption process begins by permuting the image using a key-based scrambling operation and then applying DNA encryption to the permuted image with a selected DNA encoding rule. Three DNA rules are then used to encrypt each RGB component. Subsequently, an XOR operation is performed between the even and odd rows of each encrypted channel and the first and second keys generated by the chaotic 2D logistic map. Experimental analysis shows that the proposed method provides robust encryption and withstands several common attacks. However, it could be further enhanced by using higher-dimensional 5D logistic chaotic map instead of the 2D logistic chaotic map and by expanding the scope of the encryption method to encompass other types of data, like audio and video. To ensure image security, [10] designs a 2D hyper-chaos system (2D-CICM) and a 3D L-shaped transformation. The SHA-256 hash of the plain image is used to initiate the conditions of 2D-CICM and generate five chaotic sequences. These sequences are utilized for Dynamic DNA encoding, permutation, diffusion, DNA crossover, and dynamic decoding after applying the 3D L-shaped transformation to the image cube of the divided plain image. While this technique is suitable for real-wold applications it depends on the SHA-256 hash to initialize the chaotic system, which may weaken the security of the encryption. A safe and reliable algorithm is presented in [11], combining Logistic and Sine chaotic maps to enhance a two-dimensional chaotic mapping and integrating a new four-dimensional chaotic system. The encryption process starts by using SHA-256 algorithm to calculate the hash value of the sum of the plain image and initial keys to generate the initial parameters for the chaotic mapping. Using the four-dimensional chaotic system, the RGB channels of the original image are scrambled in ascending order. The scrambled image is then encrypted using DNA rules and DNA operations, including addition, subtraction, and XOR, based on the chaotic matrices generated from the two-dimensional chaotic mapping. While the algorithm is safe and reliable, it needs to explore complex chaotic maps and optimization to enhance the performance.

Even though existing chaotic-DNA encryption methods resist differential and brute-force attacks with some achieving key spaces as large as 2^{884}, their effectiveness may be limited by the rapid advancements in technology. To address these challenges, this paper proposes a DNA-based logistic map algorithm to secure RGB images, offering superior resistance to differential and brute-force attacks. This makes it suitable for medical image encryption, secure cloud storage, biometric image protection, and watermarking.

This paper is organized as follows. Section 2 introduces the fundamentals of DNA cryptography and the chaotic logistic map. Section 3 provides a detailed description of the proposed color image encryption scheme. Section 4 presents an experimental evaluation and security analysis of the proposed method. Finally, Section 5 offers concluding remarks.

2. Preliminaries

2.1. Logistic map

In cryptography, chaotic maps are widely used for generating pseudo-random sequences as encryption keys. The inherent chaotic nature exhibits sensitivity to initial conditions and parameter values, resulting in unpredictability and significant variations in the generated sequences. The logistic map, a mathematical

system demonstrating chaotic behavior, is defined by the equation (1), where $r \in [0, 4]$ and $x_i \in [0, 1]$. To prevent the system from converging to stable fixed points or periodic limit cycles, the parameter r of the logistic map is typically constrained to $r \in [3.57, 4]$ [12].

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

2.2. DNA cryptography

DNA cryptography is one of the rapidly growing technologies that use Deoxyribonucleic acid (DNA), the genetic material responsible for building living organisms. [13] DNA consists of four nitrogen bases: Adenine(A), Thymine(T), Guanine(G), and Cytosine(C), which pair up following the complementary theory of Watson-Crick. Adenine(A) pairs with Thymine(T), and Cytosine(C) pairs with Guanine (G). [14] Various DNA operations can be used in DNA cryptography including DNA complement, DNA replication, DNA transcription, and Translation, among others.

2.3. DNA Encoding and Decoding

In DNA Cryptography, the digital data is converted into a DNA sequence known as *DNA encoding operation* or *DNA encryption*, which converts each two bits of digital data to a unique DNA base using a specific encoding scheme. The resulting DNA sequence can be stored or transmitted. To recover the original data, the DNA sequence is converted into binary value using the same encoding scheme. This process is called *DNA decoding operation* or *DNA decryption*. Out of the 24 possible ways to encode four DNA bases, only eight satisfy the complementary rule (see Table 2).

2.4. DNA-XOR Operation

The XOR (exclusive-OR) operation between two DNA sequences is known as the DNA-XOR operation, following the same concept as the binary XOR operation. Nevertheless, the result of the DNA-XOR operation varies depending on the chosen DNA encoding/decoding rule as shown in Table 1.

3. Proposed image encryption scheme

The proposed method is an enhanced algorithm for image encryption based on a logistic map and DNA cryptography. It decomposes into two essential parts: key generation and image encryption. The key generation process involves randomly generating a DNA encoding/decoding table. Based on this table, two permutation keys are generated—one for row permutation and the other for column permutation. Additionally, keys for each channel of the image are generated using the logistic map. Seven DNA encoding/decoding rules are selected: the first three rules are for encoding the RGB channels of the image, the second three rules are for encoding the generated keys, and the last rule is used to perform the DNA XOR operation between the encoded channels and their keys. The encryption of the image starts by permuting its rows and columns. Then, the RGB channels of the permuted image are extracted and encoded using the second set of three selected DNA rules. The DNA XOR operation is performed with the keys, and finally, the results are decoded using the DNA decoding table and merged to construct the encrypted image. The decryption process is the reverse of the encryption algorithm where Figure 1 shows a framework of the proposed encryption method. The key generation and image encryption processes are detailed in the two next sections.

3.1. Key generation process

Generating DNA encoding and decoding tables. Generating a DNA encoding/decoding table requires randomly associating DNA sequences with specific integers based on the desired length of the DNA sequence. This can be done as follows:

Step 1: Define B the number of DNA bases used to encode the data, with $B \geqslant 4$.

Step 2: Generate *dna_values* all possible DNA sequences of size B.

Step 3: Calculate the number of bits N, where: $N = 2 \times B$.

Step 4: Generate all possible integer values *integer_values* of size N bits.

Step 5: Initialize the dictionaries *encoding_table* and *decoding_table* to empty.

Step 6: Select a random DNA value *rand_dna* and remove it from *dna_values*.

Step 7: Select a random integer value *rand_int* and remove it from *integer_values*.

Step 8: Add the selected values *rand_dna* and *rand_int* to *encoding_table* and *decoding_table*, where:

$$encoding\_table[rand\_int] = rand\_dna \tag{2}$$

$$decoding\_table[rand\_dna] = rand\_int \tag{3}$$

Step 9: Repeat *Steps 6, 7, and 8* until the *integer_values* and *dna_values* are empty.

Generating permutation keys. Generating permutation keys involves creating two sets of keys, one for the permutation of rows and another for column permutation based on the dimensions of the image and the generated DNA encoding/decoding table. The permutation keys are generated as follows:

Step 1: Determine the dimensions of the image where:

- H is the height (number of rows) of the image.
- W is the width (number of columns) of the image.

Table 1. DNA-XOR Operation

	Rule 1				Rule 2				Rule 3				Rule 4			
⊕	A	T	C	G	A	T	C	G	A	T	C	G	A	T	C	G
A	A	T	C	G	A	T	C	G	T	A	G	C	T	A	G	C
T	T	A	G	C	T	A	G	C	A	T	C	G	A	T	C	G
C	C	G	A	T	C	G	A	T	G	C	T	A	G	C	T	A
G	G	C	T	A	G	C	T	A	C	G	A	T	C	G	A	T
	Rule 5				Rule 6				Rule 7				Rule 8			
⊕	A	T	C	G	A	T	C	G	A	T	C	G	A	T	C	G
A	C	G	A	T	C	G	A	T	G	C	T	A	G	C	T	A
T	G	C	T	A	G	C	T	A	C	G	A	T	C	G	A	T
C	A	T	C	G	A	T	C	G	T	A	G	C	T	A	G	C
G	T	A	G	C	T	A	G	C	A	T	C	G	A	T	C	G

Table 2. DNA encoding/decoding rules

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

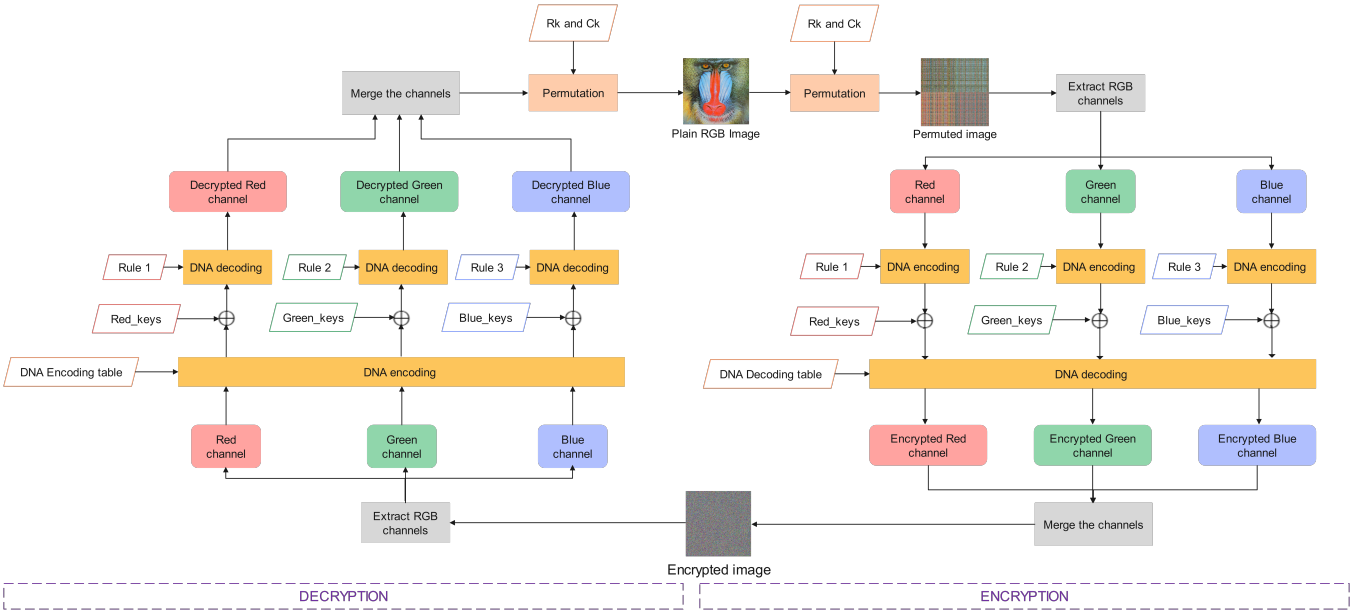

Figure 1. Framework of the proposed encryption method

Step 2: Extract *Pk* the primary key of permutation from the DNA encoding/decoding table, where *L* is the length of *Pk*:

- In the case of taking the DNA encoding table as a reference, the *Pk* is equal to the list of the values of the dictionary *encoding_table*.
- In the case of taking the DNA Decoding table as a reference, the *Pk* is equal to the list of the keys of the dictionary *decoding_table*.

Step 3: Generate *Rk*, the row permutation key, based on *Pk* by implementing Algorithm 1 and using *H* as the number of key permutations (*key_number*).
Step 4: Generate *Ck*, the column permutation key, based on *Pk* by implementing Algorithm 1 and using *W* as the number of key permutations (*key_number*).

Algorithm 1 Permutation keys

Require: $Pk, key\_number, L, i, j, n$
Ensure: $keys$
$L \Leftarrow length(Pk)$
if $L = key\_number$ **then**
  $keys \Leftarrow Pk$
else if $L > key\_number$ **then**
  $keys \Leftarrow []$
  $i \Leftarrow 0$
  while $i < key\_number$ **do**
    $keys.append(Pk[i])$
    $i \Leftarrow i + 1$
  end while
else
  $keys \Leftarrow Pk$
  for $i \in range(2, key\_number + 2)$ **do**
    for $j \in Pk$ **do**
      $n \Leftarrow (i \times j) \mod key\_number$
      if $n \notin keys$ **then**
        $keys.append(n)$
      end if
    end for
  end for
end if

DNA Encoding/Decoding rules selection. Generating *dna_rules* seven random DNA rules for encoding/decoding the data using the logistic map function via the application of Algorithm 2, where $num = 7$ and $max\_val = 7$:

- r and x_0, are the initial parameters of the logistic map.
- num, represents the number of keys/rules needed to generate.
- max_val, is the maximum value that the generated random number can take.
- normalize, is a process that transforms numerical values x of $numbers$ into another values y that are less or equal to a maximum value max_val. There are many techniques to normalize data. Here, we use Min-Max normalization with the following formula [15]:

$$y = \frac{x - min(numbers)}{max(numbers) - min(numbers)} \times max\_val \tag{4}$$

- round_int involves rounding the numerical values x of $numbers$ to the nearest integer.

Algorithm 2 Random numbers

Require: $r, x_0, num, max\_val, i, x$
Ensure: $numbers, x$
$x \Leftarrow x_0$
$i \Leftarrow 0$
$numbers \Leftarrow []$
while $i < num$ **do**
  $x \Leftarrow Logistic\_map(r, x)$
  $numbers.append(a)$
end while
$numbers \Leftarrow normalize(numbers, max\_val)$
$numbers \Leftarrow round\_int(numbers)$

Generating keys for image encryption. Generating keys for image encryption requires creating three sets of keys: *Red_keys*, *Blue_keys*, and *Green_keys*, to encrypt each channel of an RGB image:
Step 1: Define num the number of keys for *Red_keys*, *Blue_keys*, and *Green_keys*. Where: $num = H \times W$.
Step 2: Define max_val the maximum value that the key can take, in the case of RGB image: $max\_val = 255$.
Step 3: Generate *Red_keys* using Algorithm 2, with r as the initial value of the logistic map, and x_0 as the resulting value x obtained from the application of Algorithm 2 in the section 3.1.
Step 4: Generate *Blue_keys* using Algorithm 2, with r as the initial value of the logistic map, and x_0 as the resulting value x obtained from the application of Algorithm 2 in step 3.
Step 5: Generate *Green_keys* using Algorithm 2, with r as the initial value of the logistic map, and x_0 as the resulting value x obtained from the application of Algorithm 2 in step 4.
Step 6: Transform *Red_keys*, *Blue_keys*, and *Green_keys* into DNA keys by encoding them using the DNA rules(*dna_rules*) 4, 5, and 6, respectively.

3.2. Image encryption process

Permutation. **Step 1:** Begin by reading the image and extracting the image matrix (*mat*).
Step 2: Apply row reordering to *mat* using the key *Rk*, and then save the result as *row_mat*:

$$row\_mat = mat[Rk] \tag{5}$$

Step 3: Reorder the columns of *row_mat* using the key *Ck*, and save the output as the *permuted_image*:

$$permuted\_image = row\_mat[:, Ck] \tag{6}$$

DNA encoding. **Step 1**: Extract the *red*, *green*, and *blue* Channels of *permuted_image*.
Step 2: Encode each channel into DNA bases using DNA rules(*dna_rules*): 3 for blue, 2 for green, and 1 for the red channel.

XOR operation. Using the seventh DNA encoding rule from *dna_rules*, perform the DNA XOR operation between the encoded channels and the generated DNA keys *Red_keys*, *Blue_keys*, and *Green_keys*. The results are matrices of DNA sequences:

$$green\_xor = green \oplus Green\_keys \quad (7)$$
$$blue\_xor = blue \oplus Blue\_keys \quad (8)$$
$$red\_xor = red \oplus Red\_keys \quad (9)$$

DNA decoding. Employing *decoding_table*, convert the DNA matrices obtained from DNA XOR operation: *green_xor*, *blue_xor*, and *red_xor* into integer matrices, and save the results as *encrypted_green*, *encrypted_blue*, and *encrypted_red*.

Generating the encrypted image. Construct the encrypted image by merging *encrypted_blue*, *encrypted_green*, and *encrypted_red* matrices.

4. Experimental evaluation and security analysis

To elucidate the strengths and weaknesses of the proposed cryptosystem, we conducted an experimental evaluation using specific metrics to validate its effectiveness, efficiency, and robustness against various cryptographic attacks. For testing, we utilized 4 color images, including the standard test images Lenna and Baboon, as well as medical images obtained from the *Brain MRI Segmentation* dataset (downloaded from *kaggle* site). The key generation parameters utilized Table 3 as the DNA encoding/decoding table, with logistic map parameters set to r=3.6501 and x_0= 0.1. The encryption scheme was implemented in Python 3.11.7 on a 64-bit Dell laptop featuring an Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz, 8GB of memory, and running Kali Linux 2024.1.

4.1. Statistical analysis

Histogram analysis. Histogram plots of the original and encrypted versions of four colored images—two 512×512 standard images (Baboon and Lena) and two 256×256 medical images (MRI scans of the brain) for each RGB channel are depicted in Figure 2. The pixel distribution of the original images exhibits non-uniformity, whereas the histogram of the encrypted images displays a rectangular shape with a flat distribution of pixel values. Therefore, the encryption process preserves the essential statistical properties of the original image and protects against statistical attacks relying on pixel value frequencies.

Correlation analysis. Correlation analysis in image encryption involves examining the relationship between the adjacent pixels of the image. This relationship can be exploited by attackers to extract patterns and derive the original image from its encrypted version or deduce the encryption key. Therefore, the correlation coefficient should be significantly reduced or eliminated after the encryption process.The correlation coefficient is determined by Equation (10), as follows:

$$r_{x,y} = \frac{\mathrm{Cov}[x,y]}{\sqrt{D(x) \cdot D(y)}}, \quad (10)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2, \quad (11)$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad (12)$$

$$\mathrm{Cov}[x,y] = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)). \quad (13)$$

Where x and y are the adjacent pixels, and N is the number of the pairs (x, y).

Table 4 displays correlation coefficients for each RGB channel in various directions: diagonal, horizontal, and vertical. The table indicates a significant change in the correlation after the encryption, with the correlations in the original RGB image close to one, indicating a strong linear relationship between pixels across different directions. Conversely, the correlations in encrypted RGB images nearing zero or becoming negative, suggesting a high degree of randomness and no linear dependency among pixel values. This correlation transformation can be visually demonstrated in Figure 3.

Entropy analysis. Entropy information used to quantify the uncertainty and randomness in the data, serving as an indicator of the strength of encryption schemes. It can be calculated using the equation (14).

$$H = -\sum_{i=1}^{N} p(i)\log_2 p(i) \quad (14)$$

where H is the entropy of the image, N is the total number of possible pixel intensity values, and $p(i)$ is the probability of occurrence of the pixel intensity i in the image.

Based on the entropy values provided for the original, encrypted, and decrypted versions of images as presented in Table 5; The entropy values of the original and decrypted images reflect similarity, underscoring the efficacy of the decryption process in recovering the original image. Notably, the two MRI scan brain images show lower entropy in comparison to the higher entropy observed in standard images Lena and Baboon. However, following the encryption

Table 3. An example of randomly generated DNA encoding/decoding table

0:GCCC	32:GTAG	64:GTTT	96:CTCG	128:TGAT	160:GCAC	192:TCCC	224:GGCC
1:ACAC	33:AAGC	65:CTGG	97:GATA	129:ATCC	161:CGTC	193:TATC	225:GCCG
2:TGTC	34:AGGA	66:GGTA	98:GTAT	130:GCTC	162:TGCC	194:TGTG	226:CACA
3:ATTC	35:CTAT	67:TACC	99:AGAC	131:ACTA	163:CGTT	195:CCGA	227:AGGG
4:GCAT	36:CGAA	68:TCAT	100:TTTC	132:TTGT	164:GAAT	196:CCTA	228:GGGA
5:CTGC	37:CAAC	69:AGTT	101:ACTG	133:TGAC	165:CCAA	197:GTCG	229:GTTG
6:GAGA	38:GCAG	70:CTAA	102:CCCC	134:CCCT	166:CGCG	198:CAAG	230:CGTG
7:TCTG	39:GGCA	71:GACA	103:AAGG	135:TAAG	167:CCAC	199:GATT	231:TTCG
8:TTAC	40:ATAA	72:TTCC	104:ACAT	136:ACCT	168:TTTA	200:TATG	232:TAGT
9:ACGG	41:GCCT	73:CATA	105:AAGA	137:GATC	169:GGAC	201:ACAG	233:GACT
10:CGGG	42:CAGA	74:TCCG	106:GAAG	138:TGGG	170:CTCA	202:GCGA	234:CGGT
11:AATC	43:AATA	75:GGAT	107:AACC	139:ACGC	171:GACG	203:GAAA	235:TCTC
12:AGCC	44:CATG	76:GAGG	108:ATCA	140:ATTA	172:TAAT	204:CTTA	236:TGGC
13:GCAA	45:GCTG	77:CTTT	109:CCTG	141:GGCG	173:CTGT	205:TCAC	237:CGTA
14:ATTT	46:TGGT	78:GTAC	110:AGAA	142:GATG	174:ACCA	206:CAAT	238:TCAA
15:TTCT	47:AAGT	79:TGCT	111:GTGA	143:CCCG	175:GGTC	207:CACC	239:CGCA
16:AACG	48:CGCC	80:TCGA	112:GGAG	144:GGGT	176:CGGC	208:TTGC	240:GTCC
17:AACA	49:TGCA	81:CATC	113:CTTC	145:GGGG	177:TAGC	209:ATAG	241:CCGC
18:AGCG	50:ATAC	82:TGTT	114:GCTT	146:GTGC	178:TACG	210:ACTT	242:GCGG
19:TCTA	51:AAAG	83:CAAA	115:CGCT	147:AGGC	179:GGGC	211:CGGA	243:GAAC
20:ACCC	52:CATT	84:TGGA	116:TTCA	148:TATT	180:AGCA	212:ATGG	244:CACG
21:ACGT	53:GTGT	85:GCTA	117:GTTC	149:TTGA	181:TAGA	213:AGTG	245:AAAT
22:CTTG	54:GGCT	86:TCTT	118:CAGC	150:CCGG	182:TAGG	214:TTGG	246:TTTT
23:TAAC	55:GAGT	87:CAGG	119:CTCC	151:AAAC	183:TCGT	215:TTAG	247:GTCA
24:TCAG	56:AATG	88:TATA	120:CGAG	152:TACT	184:AGTA	216:CTAG	248:CGAC
25:TGCG	57:CGAT	89:ACTC	121:AAAA	153:ATGA	185:GTGG	217:ATCT	249:CCAT
26:TGAA	58:CCTC	90:ATAT	122:AGAT	154:ATTG	186:GCCA	218:CACT	250:GTCT
27:GTTA	59:AATT	91:GGAA	123:TCCT	155:GCGT	187:ACCG	219:TCGC	251:TACA
28:TTAT	60:AGGT	92:ATCG	124:AGAG	156:TTTG	188:CCCA	220:CTAC	252:AGCT
29:GCGC	61:ATGC	93:TCCA	125:GTAA	157:TAAA	189:CCAG	221:CTCT	253:AACT
30:TTAA	62:GGTG	94:ACAA	126:CTGA	158:ACGA	190:CCTT	222:GGTT	254:TGAG
31:GACC	63:GAGC	95:ATGT	127:AGTC	159:CCGT	191:TCGG	223:CAGT	255:TGTA

process, there is a significant increase in entropy, approaching a value close to 8, thereby enhancing the security of the encrypted data against entropy-based attacks.

4.2. Differential analysis

Evaluating the resistance of the image encryption algorithm to differential attacks involves measuring the values NPCR, UACI, and PSNR, as presented in Table 6, Table 7, and Table 10 respectively.

Number of pixels change rate (NPCR). The NPCR metric measures the percentage of differing pixels between two images. A high NPCR value indicates a better resistance to differential attacks, with the ideal value being greater or equal to 99.609375%. The value of NPCR is calculated as:

$$\text{NPCR} = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{D(i,j)}{M\times N}\times 100\% \tag{15}$$

where M and N are the dimensions of the images, and $D(i,j)$ is defined as:

$$D(i,j) = \begin{cases} 0 & \text{if } C_1(i,j) = C_2(i,j) \\ 1 & \text{if } C_1(i,j) \neq C_2(i,j) \end{cases} \tag{16}$$

Here, $C_1(i,j)$ and $C_2(i,j)$ represent the pixel values at position (i,j) in the two images being compared.

The NPCR for the standard images Lena and Baboon closely approach the ideal value of NPCR, showing approximately 99.60% variation between the pixels of the encrypted and original images. In contrast, the NPCR value for the two MRI brain scan images slightly exceeds the target threshold at 99.62% pixel variation between the original and encrypted images. Taken together, the average NPCR for the four images equates to 99.613667% demonstrating a close proximity to the desired value.

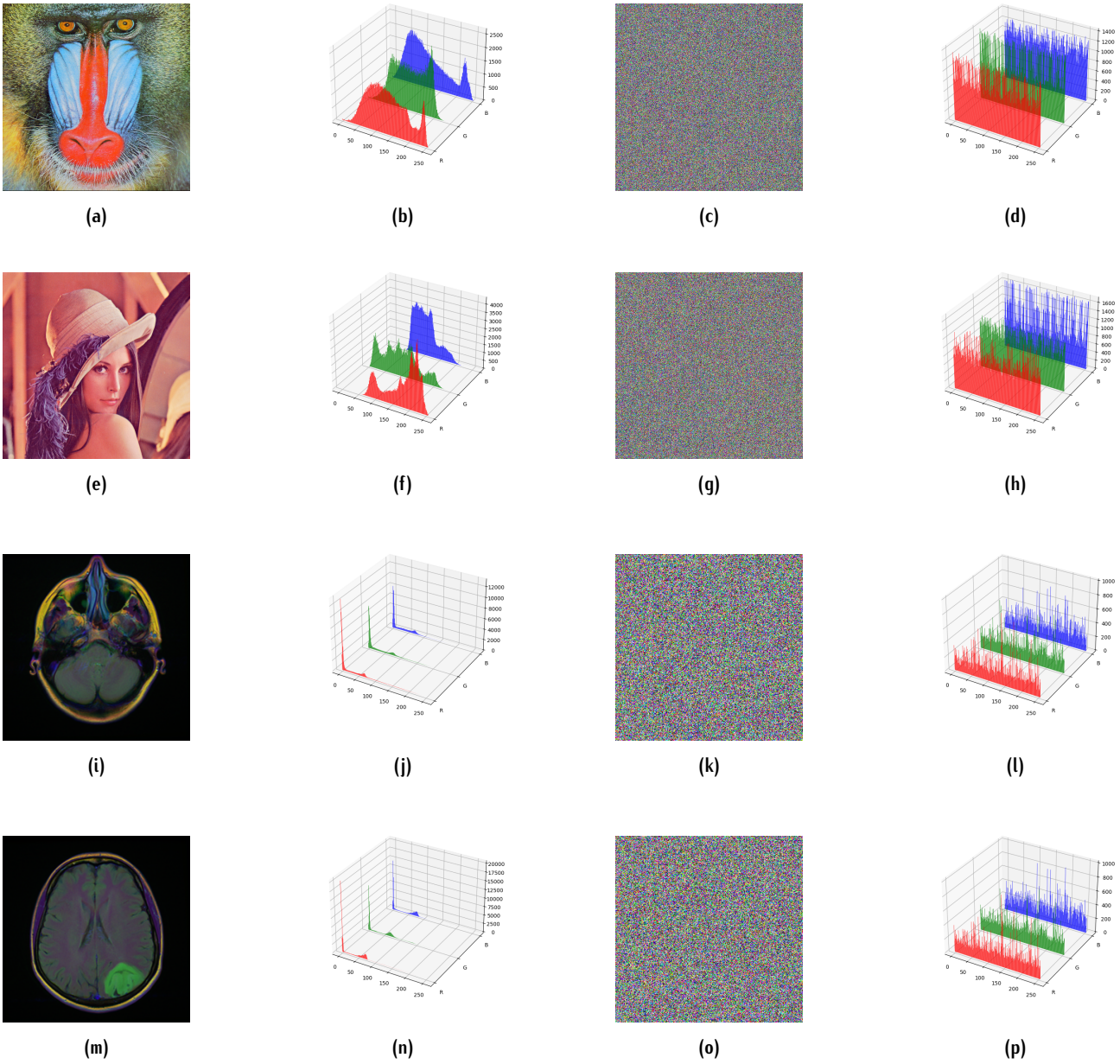

Figure 2. Histograms of original and encrypted images: (a) The original image of Baboon; (b) The histogram of Baboon; (c) The encrypted image of Baboon; (d) The histogram of encrypted Baboon; (e) The original image of Lena; (f) The histogram of Lena; (g) The encrypted image of Lena; (h) The histogram of encrypted Lena; (i) The original image of MRI Brain 1; (j) The histogram of MRI Brain 1; (k) The encrypted image of MRI Brain 1; (l) The histogram of encrypted MRI Brain 1; (m) The original image of MRI Brain 2; (n) The histogram of MRI Brain 2; (o) The encrypted image of MRI Brain 2; (p) The histogram of encrypted MRI Brain 2;

Unified averaging changing intensity (UACI). UACI is used to calculate the average change in intensity between two images. A high UACI indicates a significant difference in pixel intensities, suggesting a stronger encryption method with better resistance to differential attacks. The formula for UACI is given as:

$$\text{UACI} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_{ij} - C'_{ij}|}{L} \times 100\% \qquad (17)$$

where M and N are the dimensions of the images, L is the maximum possible pixel value in an image, C_{ij} and C'_{ij} represent the pixel values at position (i, j) in the two images being compared.

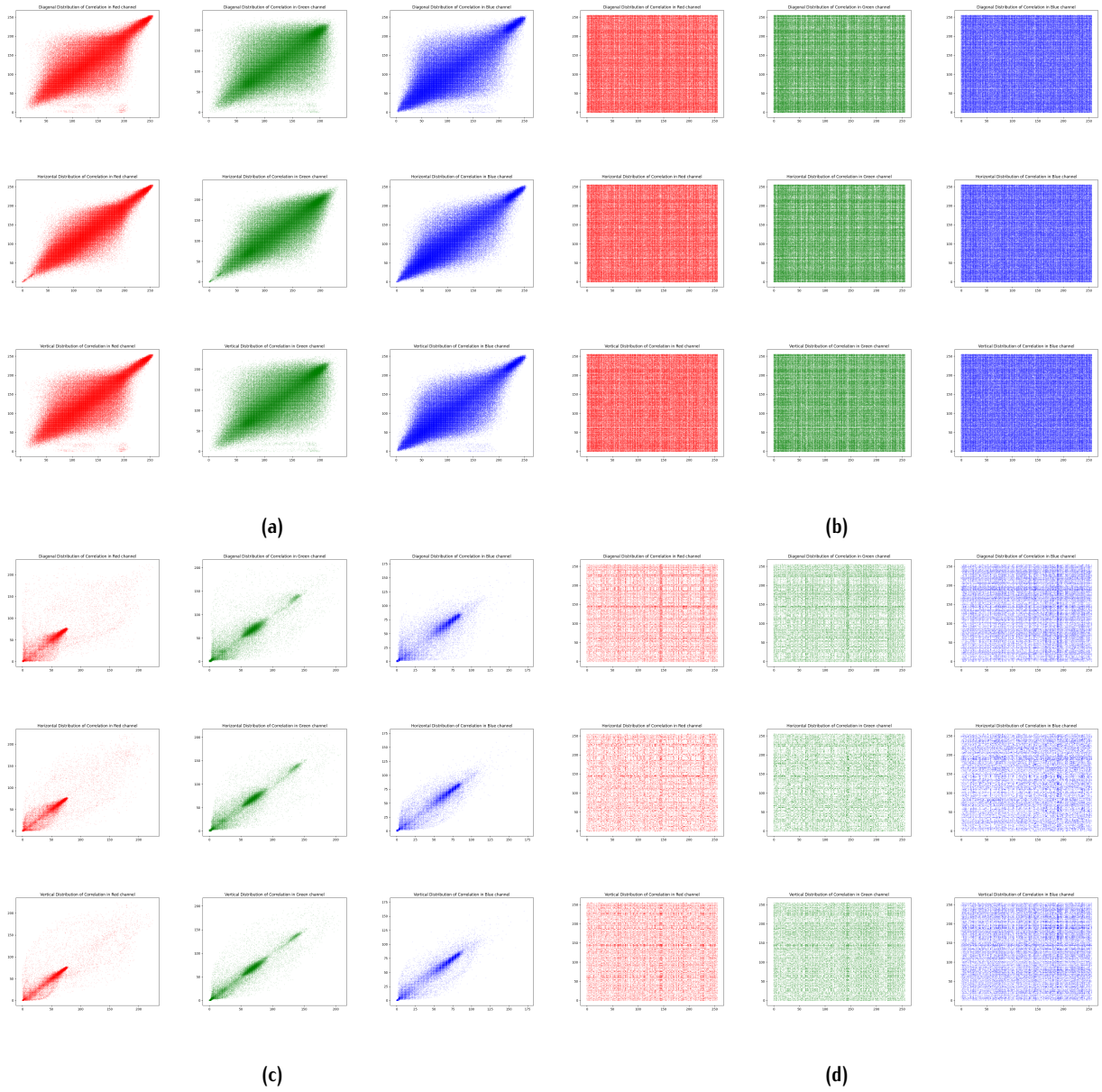

Figure 3. RGB channel correlation distribution: (a) Correlation distribution of Baboon image; (b) Correlation distribution of encrypted Baboon image; (c) Correlation distribution of MRI Brain 2; (d) Correlation distribution of encrypted MRI Brain 2

The UACI values for the tested images exceed the ideal UACI value of 33.463541%, signifying that around 50% of pixel intensities in the encrypted images differ from those in the originals, making recovery of the original image difficult without the knowledge of the decryption key.

Peak signal-to-noise ratio (PSNR). PSNR assesses the accuracy of an image reconstruction in comparison to its original form. Enhanced PSNR values denote greater similarity between the original and reconstructed images. The PSNR value can be determined by the following formula:

$$PSNR = 20 \cdot \log\left(\frac{255}{\sqrt{MSE}}\right) \tag{18}$$

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(I_o(i) - I_r(i))^2 \tag{19}$$

where MSE is the Mean Squares Error, M and N are the dimensions of the images, $I_o(i,j)$ and $I_r(i,j)$ represent the pixel values at the position (i,j) of the original and reconstructed image, respectively.

Table 4. Correlation coefficients of original and encrypted images

		Original			Encrypted		
		Red	Green	Blue	Red	Green	Blue
Baboon	H	0.923066	0.865479	0.907344	-0.005701	-0.003868	-0.002679
	V	0.865959	0.765007	0.880892	0.013083	0.000562	0.003204
	D	0.854341	0.734795	0.839855	-0.007372	-0.00123	-0.000655
Lena	H	0.979774	0.969067	0.932742	-0.007853	-0.005017	-0.000631
	V	0.989316	0.982493	0.957605	0.013492	0.001267	-0.002211
	D	0.969694	0.955546	0.918286	-0.006954	-0.005264	-0.003411
MRI Brain 1	H	0.956226	0.961021	0.96403	0.005755	-0.002326	-0.011069
	V	0.966706	0.974455	0.975343	0.003673	0.003574	0.009407
	D	0.932983	0.942331	0.944834	-0.004018	-0.000677	-0.000967
MRI Brain 2	H	0.953186	0.965065	0.980856	0.000333	0.005746	-0.011049
	V	0.954607	0.971075	0.983415	0.003122	-0.005185	0.007722
	D	0.918622	0.94108	0.967311	-0.005862	-0.003382	-0.007006

Table 5. Information entropy test results

		Original	Encrypted	Decrypted
Baboon	R	7.706672	7.970384	7.706672
	G	7.474432	7.946146	7.474432
	B	7.752217	7.987623	7.752217
Average		7.644440	**7.968051**	7.644440
Lena	R	7.253102	7.971971	7.253102
	G	7.594038	7.980907	7.594038
	B	6.968427	7.890859	6.968427
Average		7.271856	**7.947912**	7.271856
MRI Brain 1	R	5.367468	7.825469	5.367468
	G	5.706242	7.862473	5.706242
	B	5.522936	7.826703	5.522936
Average		5.532215	**7.838215**	5.532215
MRI Brain 2	R	4.825784	7.856646	4.825784
	G	5.318319	7.885874	5.318319
	B	4.979613	7.844465	4.979613
Average		5.041239	**7.862328**	5.041239

Table 6. NPCR test results

	Red	Green	Blue	NPCR(%)
Baboon	99.617767	99.593735	99.594879	99.602127
Lena	99.590683	99.594116	99.61586	99.600220
MRI Brain 1	99.630737	99.612427	99.632263	99.625142
MRI Brain 2	99.607849	99.620056	99.653625	99.627177
Average	99.611759	99.605083	99.624157	**99.613667**

Table 7. UACI test results

	Red	Green	Blue	UACI(%)
Baboon	49.991155	49.989259	50.061715	50.014043
Lena	49.784631	50.030267	49.935037	49.916645
MRI Brain 1	50.38383	50.419611	50.754607	50.519349
MRI Brain 2	50.535774	50.550318	50.848699	50.64493
Average	50.173847	50.247364	50.400014	**50.273742**

Table 8. UACI results of key sensitivity analysis

Key	Image	Red	Green	Blue	UACI (%)
$Key_1(x_0 + 10^{-15})$	Baboon	49.429119	49.385199	49.415788	49.410035
	Lena	49.539250	49.517599	49.441355	49.499401
	MRI Brain 1	49.602574	49.189186	49.305174	49.365645
	MRI Brain 2	49.391371	49.288470	49.238819	49.306220
$Key_2(r + 10^{-15})$	Baboon	49.516824	49.484631	49.320456	49.440637
	Lena	49.312237	49.455762	49.414015	49.394005
	MRI Brain 1	49.569538	49.238950	49.490443	49.432977
	MRI Brain 2	49.603924	49.257824	49.598181	49.486643
$Key_3(x_0 + 10^{-15}, r + 10^{-15})$	Baboon	49.372122	49.380100	49.487665	49.413296
	Lena	49.470076	49.269876	49.564701	49.434884
	MRI Brain 1	49.368942	49.648565	49.522474	49.513327
	MRI Brain 2	49.403155	49.323934	49.369043	49.365377
Average		49.464928	49.370008	49.430676	**49.421871**

Table 9. NPCR results of key sensitivity analysis

Key	Image	Red	Green	Blue	NPCR (%)
Key1$(x_0 + 10^{-15})$	Baboon	98.567963	98.522568	98.397446	98.495992
	Lena	98.567963	98.522568	98.397446	98.495992
	MRI Brain 1	98.524475	98.274231	98.202515	98.333740
	MRI Brain 2	98.524475	98.274231	98.202515	98.333740
Key2$(r + 10^{-15})$	Baboon	98.422623	98.479080	98.399353	98.433685
	Lena	98.422623	98.479080	98.399353	98.433685
	MRI Brain 1	98.492432	98.394775	98.558044	98.481750
	MRI Brain 2	98.492432	98.394775	98.558044	98.481750
Key3$(x_0 + 10^{-15}, r + 10^{-15})$	Baboon	98.449326	98.300552	98.540497	98.430125
	Lena	98.449326	98.300552	98.540497	98.430125
	MRI Brain 1	98.469543	98.345947	98.448181	98.421224
	MRI Brain 2	98.469543	98.345947	98.448181	98.421224
Average		98.487727	98.386192	98.424339	**98.432753**

An infinite PSNR (∞) resulting from a test between the original and decrypted images implies a perfect reconstruction of the original image, with no loss of information during the encryption and decryption process.

4.3. Exhaustive attack analysis

Key sensitivity analysis. The concept of key sensitivity in image encryption means that slight changes in the key produce significantly different encrypted images. It also implies that the original image (decrypted) cannot be derived using an incorrect decryption key.

Table 10. PSNR test results

	Red	Green	Blue
Baboon	∞	∞	∞
Lena	∞	∞	∞
MRI Brain 1	∞	∞	∞
MRI Brain 2	∞	∞	∞

In our analysis, we encrypted color images of Lena and Baboon, as well as two MRI images, using four slightly different keys: $Key_0(r = 3.6501, x_0 = 0.1)$, $Key_1(r = 3.6501, x_0 = 0.1 + 10^{-15})$, and $Key_2(r = 3.65010.1 + 10^{-15}, x_0 = 0.1)$, and $Key_3(r = 3.65010.1 + 10^{-15}, x_0 = 0.1 + 10^{-15})$. We then calculated the UACI (Table 8) and NPCR (Table 9) values between the encrypted images produced by Key_0 and the other keys.

The results indicate an average NPCR of 98.432753%, suggesting that around 98.43% of the pixels in the encrypted image change when the key is changed slightly. An average UACI of 49.421871%, suggests that around 49.42% of pixel intensities differ between the encrypted images with small changes in the key.

In summary, high values of NPCR close to 100% and UACI close to 50% demonstrate the strong key sensitivity of our image encryption algorithm.

Key space analysis. The brute-force attack involves generating all possible keys and testing them one by one until attempting to decrypt the ciphertext. With a key space of 2^n, the attacker needs to try 2^{n-1} keys before finding the correct one, where n represents the number of bits in the key. To resist this attack, an encryption method requires a large key size, typically greater than 2^{100} for image encryption. To evaluate if our proposed method resists brute-force attacks, we quantify its key space size as follows:

- **DNA Encoding Table :** This table represents a random mapping of DNA sequences with a size $B \geqslant 4$ to 256 unique numbers. Given that DNA is composed of four bases, the number of possible sequences of length B is 4^B. Due to the unique and random arrangement of these 256 numbers and DNA sequences, calculating the key space ks_1 requires considering both the selection and permutation between them, as represented by the equation (20):

$$ks_1 = \binom{4^B}{256} \times 256! \tag{20}$$

Where $\binom{4^B}{256}$ is the binomial coefficient, defined as:

$$\binom{4^B}{256} = \frac{(4^B)!}{256! \times (4^B - 256)!} \tag{21}$$

By combining the formula (20) and (21), the key space ks_1 is simplified to:

$$ks_1 = \frac{(4^B)!}{(4^B - 256)!} \tag{22}$$

In our study, we generate the DNA encoding table using 4 as the value of B, which means:

$$ks_1 = 256! \tag{23}$$

To express 256! as (2^n), we use Stirling's approximation for factorials [17], leading to:

$$ks_1 \approx 2^{1683.89} \approx 2^{1684} \tag{24}$$

- **Permutation keys:** Since the image had the size of H*W , the number of row permutation keys is H and the number of column keys is W. The key space ks_2 and ks_3 for each, respectively, are defined as:

$$ks_2 = H! \tag{25}$$

$$ks_3 = W! \tag{26}$$

- **Logistic map :** Using logarithms (see equation (29)) to express the key space size of logistic map parameters in the form 2^k and considering that the accuracy of the computer is 10^{-15}, the key space ks_4 of x_0 and ks_5 of r are calculated as follows:

$$ks_4 = 10^{15} \approx 2^{48.83} \approx 2^{50} \tag{27}$$

$$ks_5 = 10^{15} \approx 2^{48.83} \approx 2^{50} \tag{28}$$

$$k = log_2(ks_i) \tag{29}$$

- **DNA encoding rules:** Selecting 7 out of 8 encoding rules is described by equation (30), while representing the key space ks_6 for selection.

$$ks_6 = \binom{8}{7} = \frac{(8)!}{7! \times (8-7)!} = 8 = 2^3 \tag{30}$$

- **RGB keys:** The key space of three H×W keys for encrypting RGB channels of an image derived from the logistic map and take 256 possible values representing by ks_7 (equation (31)).

$$ks_7 = 256^{3 \times H \times W} = (2^8)^{3 \times H \times W} = 2^{24 \times H \times W} \tag{31}$$

The total key space ks_{total} is the combination of the seven key spaces, computed by multiplying them together.

$$ks_{total} = \prod_{i=1}^{7} ks_i \tag{32}$$

$$ks_{total} \approx 2^{1787+24\times H\times W} \times H! \times W! \tag{33}$$

The total keyspace ks_{total} of the proposed method depends on the image size. Even when considering the minimum values of image dimensions, such as 2×2, $ks_{total} \approx 2^{1885}$, which is significantly greater than 2^{100}. Consequently, the proposed cryptosystem offers a large key space, providing a strong defense against exhaustive attacks.

4.4. Noise attacks

Ciphered images can be affected by noise during transmission or storage, where a good and robust encryption algorithm should resist the noise attacks. A pepper noise attack, which replaces a random pixel with black, is applied to encrypted images with different densities of 0.1, 0.15, and 0.2 as shown in Figure 4. The results demonstrate that the proposed method is robust against noise attacks and can reconstruct a visible decrypted image even after data corruption.

4.5. Known plaintext attacks

Known plaintext attacks are a common type of cryptographic attack in which the attacker knows both the original and encrypted images, and attempts to predict the encryption key through statistical analysis. To test if our proposed method resists known plaintext attacks, we encrypt two plaintext images (one all-white and one all-black) and evaluate their information entropy (see Table 11), Histograms (see Figure 5), and correlation coefficient (see Table 12).

Table 11. Information entropy test of white and black images

		Original	Encrypted
White image	R	0	7.439829
	G	0	7.449233
	B	0	7.440157
Black image	R	0	7.439829
	G	0	7.449233
	B	0	7.440157

The pixel values in a white or black image are identical, indicating no variation or distribution across the RGB channels. This uniformity results in an entropy of 0 and a correlation coefficient of Not-a-Number (NaN).

The encrypted white and black images exhibit entropy values close to 8, correlation coefficients near 0, and a pixel distribution distinct from the original images. This suggests that the encryption scheme randomizes pixel values and eliminates any pixel relationships. In summary, our proposed encryption technique resists Known plaintext attacks.

4.6. Comparative analysis

To ensure the effectiveness of our proposed encryption method, we compared it with existing studies on color image encryption using DNA and chaotic maps. Although these studies used varying image datasets and resolutions, we evaluated them against our method by calculating the average of standard cryptographic metrics (NPCR, UACI, and entropy) for each study, as well as comparing the key space, as detailed in Table 13.

The entropy value of our method is 7.8503 for encrypted MRI images and 7.9580 for encrypted standard images, with an average of 7.9041. While this is slightly lower than the entropy of other methods (ranging from 7.9821 to 7.9995), considering that the entropy of the original MRI images is around 5 (see Table 5), our method demonstrates a remarkable increase in entropy, approaching the ideal value of 8. This signifies a significant enhancement in randomness, ensuring a high level of security and more resistance to statistical attacks. Comparing the proposed method with existing techniques in terms of NPCR shows a similar level of encryption robustness in relation to sensitivity to pixel changes, resulting an NPCR of 99.6137%, comparable to the 99.60%-99.65% range observed in other studies. Our method attains a significantly higher UACI of 50.2737% versus other methods, which range from 31.0833% to 33.5543%, making it highly secure against differential attacks. Unlike existing studies, our method demonstrates notable resistance to brute-force attacks with a key space greater than or equals 2^{1885}.

In summary, the proposed encryption method ensures a high security with a superior resistance to differential and brute-force attacks. While its entropy is lower than other techniques, it still achieves a significant increase compared to the original MRI images, ensuring high randomness. Additionally, the high NPCR and improved UACI make our method ideal for medical image encryption, secure cloud storage, biometric image protection, and watermarking.

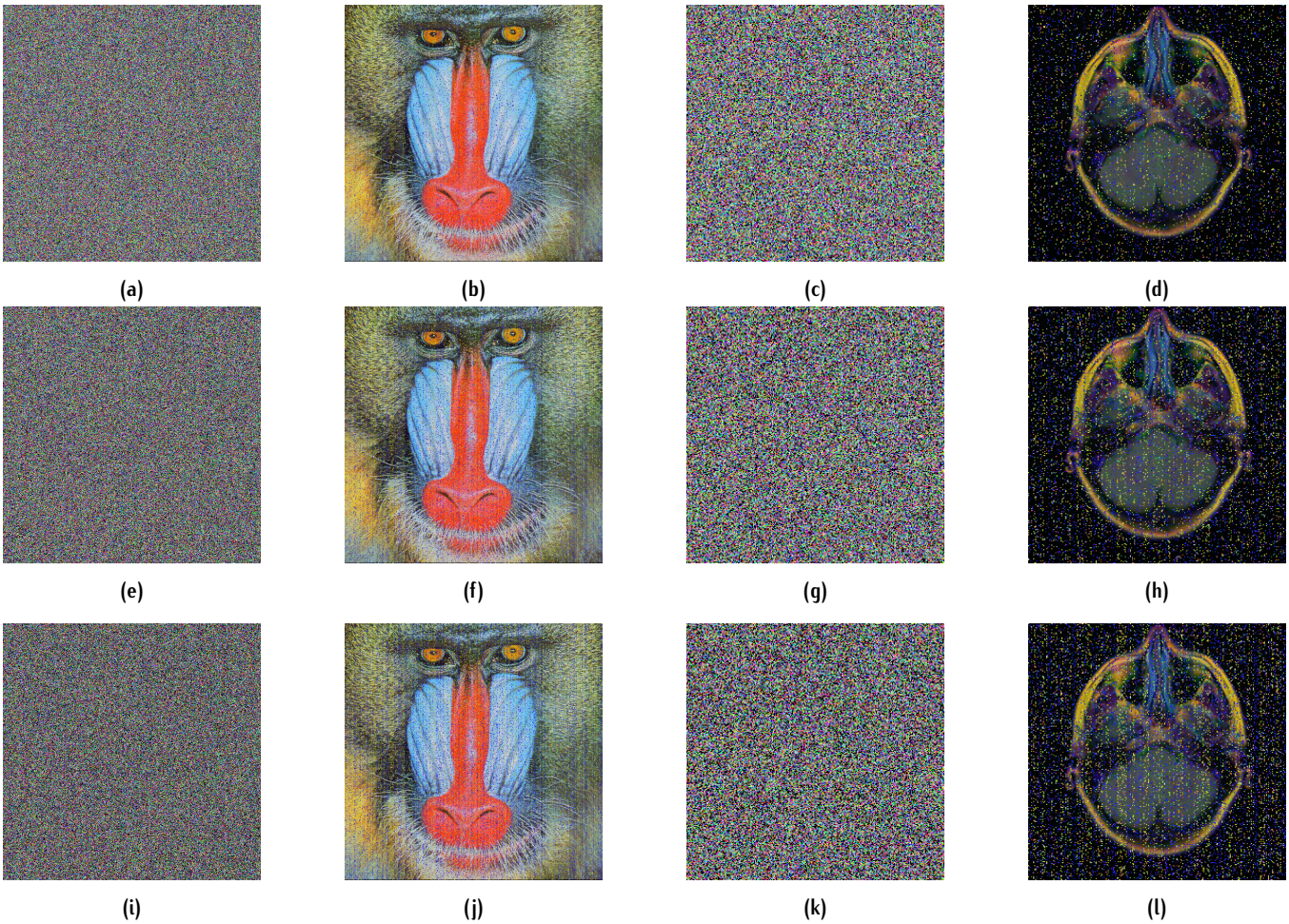

Figure 4. Effects of Noise on Encrypted and Decrypted Images: Top Row (Noise Density 0.1), Middle Row (Noise Density 0.15), Bottom Row (Noise Density 0.2)

Table 12. Correlation coefficients of original and encrypted white and black images

		Original			Encrypted		
		Red	Green	Blue	Red	Green	Blue
White image	H	NaN	NaN	NaN	-0.034117	-0.003305	-0.045045
	V	NaN	NaN	NaN	0.001960	0.006953	0.012779
	D	NaN	NaN	NaN	-0.002561	-0.007787	-0.015641
Black image	H	NaN	NaN	NaN	-0.003415	-0.047553	-0.009004
	V	NaN	NaN	NaN	-0.002726	0.012047	0.011738
	D	NaN	NaN	NaN	-0.000854	-0.007890	-0.007373

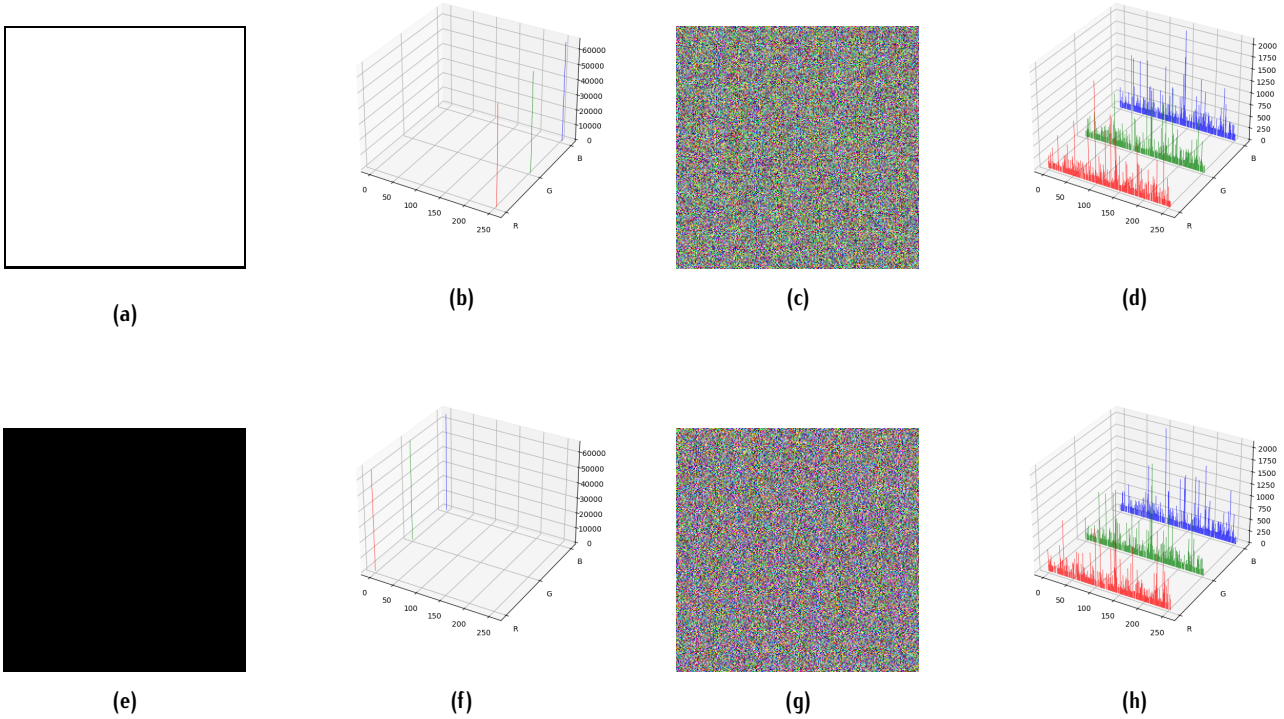

Figure 5. Histograms of original and encrypted white and black images: (a) The white image; (b) The histogram of white image; (c) The encrypted white image; (d) The histogram of encrypted white image; (e) The original black image; (f) The histogram of black image; (g) The encrypted black image; (h) The histogram of encrypted black image;

Table 13. Comparative analysis of different encryption methods

Method	Entropy	NPCR	UACI	Key space
Proposed	**7.9041**	**99.6137**	**50.2737**	$\gtrsim 2^{1885}$
Proposed - Standard Images	7.9580	99.6012	49.9653	
Proposed - MRI Images	7.8503	99.6262	50.5821	
[2]	7.9984	99.6094	33.4635	-
[3]	7.9973	99.6107	33.4593	$10^{90} \approx 2^{299}$
[4]	7.9993	99.6567	33.5543	$10^{64} \approx 2^{213}$
[5]	7.9973	99.6192	31.0833	-
[6]	7.9821	99.6222	31.0857	$10^{112} \approx 2^{372}$
[7]	7.9973	99.6157	33.4830	$10^{266} \approx 2^{884}$
[8]	7.9972	99.6300	33.5024	$10^{60} \approx 2^{200}$
[9]	7.9929	99.5937	33.3690	-
[10]	7.9972	99.6093	33.4622	$10^{154} \approx 2^{511}$
[11]	7.9995	99.6059	33.4385	$15^{31} \approx 2^{121}$

5. Conclusion

This paper describes a novel algorithm based DNA using logistic map to encrypt and decrypt color images. The proposed method randomly generates a DNA encoding/decoding table to generate the row-column permutation of the image. After the permutation of the image, the logistic map is used to generate three keys for RGB channels and seven DNA encoding-decoding rules, three are used to encode the keys into DNA sequence, the second three to encode the image RGB channel, and the last to perform the DNA-XOR operation. Finally, decode the result into integers using the DNA encoding/decoding table and generate the encrypted image. Through security analysis, including statistical analysis, correlation analysis, and exhaustive attack analysis, it is shown that the proposed algorithm resists various attacks such as statistical attacks, differential attacks, exhaustive attacks, noise attacks, and known-plaintext attacks. A comparative evaluation with existing research highlights the enhanced security features such as key sensitivity, resistance to differential attacks, and key space expansion. As a result, the proposed method is well-suited for applications like medical image encryption, secure cloud storage, and biometric image and watermarking protection. In the future, we will test our proposed method with complex chaotic systems and integrate it with other cryptographic techniques to ensure authentication while optimizing it to reduce encryption and decryption time. Additionally, we will evaluate the proposed method with other types of data, such as text.

Declarations

Funding: This research did not receive support from any organization for the submitted work.
Conflict of Interest: The author declares that they have no conflict of interest.
Informed Consent: Informed consent was obtained from all individual participants included in the study.
Data Availability: The Brain MRI Segmentation datasets utilized in the experimentation were acquired from Kagel's 'MATEUSZ BUDA' repository.

References

[1] Dixit, P., Gupta, A.K., Trivedi, M.C., Yadav, V.K.: Traditional and hybrid encryption techniques: a survey. In: Networking Communication and Data Knowledge Engineering: Volume 2, pp. 239-248 (2018). Springer. https://doi.org/10.1007/978-981-10-4600-1_22

[2] Liu, H., Zhao, B., Huang, L.: A remote-sensing image encryption scheme using dna bases probability and two-dimensional logistic map. IEEE Access 7, 65450-65459 (2019). https://doi.org/10.1109/access.2019.2917498

[3] Huang, L., Wang, S., Xiang, J., Sun, Y.: Chaotic color image encryption scheme using deoxyribonucleic acid (dna) coding calculations and arithmetic over the galois field. Mathematical Problems in Engineering 2020(1), 3965281 (2020). https://doi.org/10.1155/2020/3965281

[4] Zheng, J., Liu, L.: Novel image encryption by combining dynamic dna sequence encryption and the improved 2d logistic sine map. IET image processing 14(11), 2310-2320 (2020). https://doi.org/10.1049/iet-ipr.2019.1340

[5] Allawi, S.T., Alshibani, D.R.: Color image encryption using lfsr, dna, and 3d chaotic maps. International journal of electrical and computer engineering systems 13(10), 885-893 (2022). https://doi.org/10.32985/ijeces.13.10.4

[6] Gabr, M., Younis, H., Ibrahim, M., Alajmy, S., Khalid, I., Azab, E., Elias, R., Alexan, W.: Application of dna coding, the lorenz differential equations and a variation of the logistic map in a multi-stage cryptosystem. Symmetry 14(12), 2559 (2022). https://doi.org/10.3390/sym14122559

[7] Wang, Q., Zhang, X., Zhao, X.: Color image encryption algorithm based on bidirectional spiral transformation and dna coding. Physica Scripta 98(2), 025211 (2023). https://doi.org/10.1088/1402-4896/acb322

[8] Shraida, G., Younis, H., Al-Amiedy, T., Anbar, M., Younis, H., Hasbullah, I.: An efficient color-image encryption method using dna sequence and chaos cipher. Comput. Mater. Contin 75, 2641-2654 (2023). https://doi.org/10.32604/cmc.2023.035793

[9] Alrubaie, A.H., Khodher, M.A.A., Abdulameer, A.T.: Image encryption based on 2dna encoding and chaotic 2d logistic map. Journal of Engineering and Applied Science 70(1), 60 (2023). https://doi.org/10.1186/s44147-023-00228-2

[10] Wang, Q., Zhang, X., Zhao, X.: Color image encryption algorithm based on novel 2d hyper-chaotic system and dna crossover and mutation. Nonlinear Dynamics 111(24), 22679-22705 (2023). https://doi.org/10.1007/s11071-023-09020-6

[11] Yan, S., Li, L., Gu, B., Sun, X., Ren, Y., Zhang, Y.: A color image encryption scheme based on chaotic mapping, chaotic system, and dna coding. Applied Intelligence 53(24), 31181-31206 (2023). https://doi.org/10.1007/s10489-023-04759-2

[12] Arroyo, D., Alvarez, G., Fernandez, V.: On the inadequacy of the logistic map for cryptographic applications. arXiv preprint arXiv:0805.4355 (2008). https://doi.org/10.48550/arXiv.0805.4355

[13] Watson, J.D., Crick, F.H.: The structure of dna. In: Cold Spring Harbor Symposia on Quantitative Biology, vol. 18, pp. 123-131 (1953). Cold Spring Harbor Laboratory Press

[14] Mahjabin, T., Olteanu, A., Xiao, Y., Han, W., Li, T., Sun, W.: A survey on dna-based cryptography and steganography. IEEE Access (2023). https://doi.org/10.1109/ACCESS.2023.3324875

[15] Henderi, H., Wahyuningsih, T., Rahwanto, E.: Comparison of min-max normal- ization and z-score normalization in the k-nearest neighbor (knn) algorithm to test the accuracy of types of breast cancer. International

Journal of Informatics and Information Systems 4(1), 13-20 (2021). https://doi.org/10.47738/ijiis.v4i1.73
[16] Baoxian Zhou, April 25, 2024, "Brain MRI Segmentation", IEEE Dataport, doi: https://dx.doi.org/10.21227/pv7k-b062.
[17] Fowler, D.: The factorial function: Stirling's formula. The Mathematical Gazette 84(499), 42-50 (2000). https://doi.org/10.2307/3621473