Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Enhancing Image Data Security: DNA Cryptography and XOR-Based Feistel Encryption

Madhav Dhakal[1], Subarna Shakya[2]

[1]Central Department of Computer Science and Information Technology, Tribhuvan University, Nepal

[2]Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus, Tribhuvan University, Kathmandu, Nepal

E-mail: [1]madhav.dhakal@mu.edu.np, [2]drss@ioe.edu.np

Abstract

This research aims to generate a robust encryption technique for grayscale images with the integration of Deoxyribonucleic acid (DNA) based computing and mathematical logical XOR operations. It addresses the challenges of securing high-dimensional image data during transmission. In this work, the first layer is based on the Feistel structure concept with logical XOR operations to enhance security, while the second layer utilizes the central dogma, translating pixel data into DNA nucleotides and performing transcription and translation processes to further transform the image. The permutation box, dynamically generated and based on the input image's data, ensures increased randomness and resistance to key-guessing attacks. The method divides the original image into same-size blocks, performs XOR operations with the key, and transforms the data into a ciphertext image using DNA cryptography principles. Experimental result shows that the proposed scheme achieves better performance, improves the randomness of ciphertext image, and enhances resilience against differential attack, statistical attack, and noise attack, making it a robust solution for secure image data transmission.

Keywords: Data Security, DNA, Digital Image Data, Encryption, Decryption

1. Introduction

In the present technophile era, secure transmission of digital images is becoming increasingly important. Due to the advancement of computer and network technology, the way

DOI: https://doi.org/10.36548/jiip.2025.1.001
Received:18.01.2025, received in revised form: 16.02.2025, accepted: 02.03.2025, published: 14.03.2025
© 2025 Inventive Research Organization. This is an open access article under the Creative Commons Attribution 4.0 International (CC BY 4.0) License

of communication has changed. Multimedia communication has gradually become an important means for people to exchange information, and because of the openness and transmission of sensitive data through public unsecured networks, the risk of compromising the confidentiality of the transmitted data is increased. To secure transmission through a public network, various techniques for the protection of data through IoT enable devices to employ data protection techniques. Cryptography is considered as the best method for transmission of data. In cryptography, at the sender end, the plaintext image is translated into scrambled form through confusion and diffusion operations; and at the receiver side, the plaintext image is recovered through the decryption process. The primary objective of cryptography is to cover up the plaintext, the key, or both from the eavesdropper.

Cryptography can be implemented in two ways based on the key used for encryption/decryption process: symmetric key encryption and asymmetric key encryption. Advanced Encryption Standard (AES), Data Encryption Standard (DES) etc., are symmetric key cryptographic techniques. These techniques use the same key for encryption and decryption of messages, whereas asymmetric key cryptography includes Rivest-Shamir-Adleman (RSA), Elliptic Curve Cryptography (ECC), etc. These techniques use different keys for the encryption and decryption processes.

Due to the high correlation in pixels, high pixel redundancy, and large volumes of data, traditional cryptographic methods like AES, DES, and RSA are not appropriate for image data security [1]. But the main concern of image security is to maintain the attack-resistant encryption techniques.

Nowadays, DNA computing is extensively used to maintain the confidentiality of text and image data during transmission. This concept is used to change plaintext data into ciphertext data to produce a robust security mechanism that prohibits unauthorized persons from accessing the original plaintext data. In contrast to the conventional methods, which use the numbers 0 and 1, DNA bases: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T) are employed for data encryption and decryption process. Most cryptographic techniques use the same strategy, transforming data to ASCII values and then to binary equivalents. These binary pairs are mapped to each of the four DNA bases and stored in a DNA sequence [2].

The main contribution of this work can be summarized as:

- Integration of Feistel structure with the central dogma, an innovative approach for image encryption and lays the groundwork for further exploration of hybrid encryption techniques of DNA cryptography.
- The use of input image for pseudorandom number generation using Linear Congruential Generator increases the randomness of the image.

The remaining sections of the present work are structured as follows: Section 2 presents a review of related literature; section 3 details the proposed working technique. Research methodology is mentioned in Section 4, followed by various security analysis techniques in Section 5. Section 6 covers the result and discussion, and then, at last the conclusion of the work is provided in Section 7.

1.1 Research Gap Analysis

Due to the limitations of traditional mathematical cryptographic techniques, AES and RSA, these algorithms don't adequately address the specific needs of high-dimensional image encryption, such as reducing pixel correlation, increasing entropy, and ensuring resilience against computational attacks. Thus, by integrating these DNA computing techniques and mathematical logical XOR operations with the concept of the Feistel structure of cryptography, the research aims to create a better framework for image data security. This gap in research presents an exciting opportunity to explore new ways of securing high-dimensional images, ensuring they are better protected during transmission.

1.2 Objective of the Study

To formulate a robust encryption technique for grayscale image that integrates DNA based computing techniques with a cryptographic framework inspired by the Feistel structure is the primary purpose of this study. To achieve this main objective, specific objectives are set as to provide:

- High-level security while maintaining a low computational cost.
- Resistance to various types of attack and generate more randomness on ciphertext image data.

1.3. Statement of the Problem

In recent years, it is a crucial issue for all academic researchers to maintain the confidentiality of data. Traditional cryptographic processes offer security, but due to the lack of high randomness of pixel values and high volume of data, it is vulnerable to computational attacks and may not be optimized for high-dimensional image encryption. To address these challenges, this research proposes this concept with a secure shared key concept. The proposed concept can increase the randomness of the image and ensure that adjacent pixels in all directions of the encrypted image do not retain their original relationships.

1.4 Research Hypothesis

To fulfill the objectives of this study, the hypotheses are defined as follows:

H_0: The encrypted image has a uniform random distribution of pixels.

H_1: The encrypted image does not have a uniform random distribution of pixels.

2. Related Work

The risk associated with the transmission of data through public networks is increasing day by day thus, researchers have focused on the issue of image encryption technology. In 1994, Adleman started the first DNA computer experiment and started a new concept of cryptography in the information age for data security [3]. Researchers have been very interested in DNA-based image encryption in the past year because of its many advantageous features, including tremendous parallelism, enormous storage, and extremely low power consumption [4]. The stages of DNA-based image encryption technique include: encoding plaintext data using DNA encoding rules, DNA operations i.e. arithmetic and logical, and DNA decoding to retain the original image. Using DNA encoding principles, the bit stream of the plaintext image is encoded to a DNA sequence in the first stage. The DNA operations are carried out over the DNA sequence in the second step. The third stage uses DNA to decode the outcomes of DNA operations to a bit stream.

In 2022, authors introduced the iEncrypt algorithm, a symmetrical key block cryptographic method that integrates the DES and DNA-based encoding [5]. The iEncrypt

algorithm addresses the limitation into the 64-bit blocks of the standard DES encryption and behaves like a 128-byte block and secret key algorithm. The proposed algorithm converts the plaintext input data and secret key into DNA nucleotides. Four methods make up the IEncrypt algorithm: rounding in encryption/decryption, DNA XOR operator, conversion from binary to nucleotides and vice versa, and the DNA nucleotides round key generator. The results of the experiments show that both iEncrypt versions are immune to most cryptanalysis attacks. Conventional cryptosystems depend upon the complex mathematical operations for encryption and decryption, demanding significant computing power.

The technique employs Watson-Crick's complementary rules to encode the plaintext into nucleotide bases. The algorithms carried out straightforward DNA-based operations on both binary values and DNA-based sequences to produce complex sequences generated with DNA nucleotides. These sequences demonstrate a desirable level of randomness, a crucial attribute for practical cryptosystems [6].

3. Proposed Work

3.1 Encryption Algorithm

Input:

$I_{(MxN)}$: Original Plaintext Image

S_k: Secure Key

Output:

$C_{(MxN)}$: Ciphertext Image

Step 1: Translate the Plaintext image *I* of size *M x N* and Secure Key S_k into 8 bit binary sequence B_S

Let, *I(x,y)* be the image pixel at location *(x,y)* of image.

$$B(I) = \bigcup_{x=1}^{M} \bigcup_{y=1}^{N} b(I(x,y))$$

For Secure Key, $B(S_k) = b(S_k)$

Step 2: Divide binary data into 'n' blocks of S_b bits.

Let L be the total binary length, then: $n = \left\lceil \frac{L}{S_b} \right\rceil$

Step 3: Apply padding 0's at LSB of last block (if necessary)

Let B_{last} be last block of data and P_{bits} be the require padding bits, if $B_{last} < S_b$ then require bit for padding are: $P_{bits} = S_b - |B_{last}|$ then apply $B_{last} = B_{last} \cup P_{bits}$

Step 4: Initialize the empty binary string, $B_{emp} = \emptyset$, to store the encrypted data.

Step 5: Apply permutation on each block to randomize the block value as $B_{per} = P(B)$

Where, B_{per} is permuted binary block

Step 6: Apply the following Substitution & XOR operation on each block of binary data:

- Split the Block B_i and Secure key S_k into equal two halves as:

Left Data =LB_i , Right Data =RB_i

Left Secure Key =LS_k,, Right Secure Key= RS_k

- Perform XOR operation with the secure key as: $L_{enc} = LB_i \oplus RS_k$ and $R_{enc} = RB_i \oplus LS_k$
- Concatenate, L_{enc} and R_{enc} of each block to generate the encrypted block as: $B_{encr(i)} = L_{enc} \cup R_{enc}$

Step 7: Resulting binary encrypted data is mapped to a randomly generated DNA sequence A, C, G,T.

- Generate mRNA sequence by Replacing Thymine (T) with Uracil (U), i.e. $T \rightarrow U$
- Generate tRNA by swapping : $A \rightarrow U,\ U \rightarrow A,\ G \rightarrow C,\ C \rightarrow G$
- Obtain Reverse tRNA by Replacing $U \rightarrow T$
- Convert Reverse tRNA sequence to binary

Step 8: Append binary data to the encrypted binary data

Step 9: Encrypted binary data is reshaped into M x N and saved as ciphertext image $C_{(MxN)}$

End

Overall process of image encryption on the basis of proposed technique is illustrated in Figure 1.

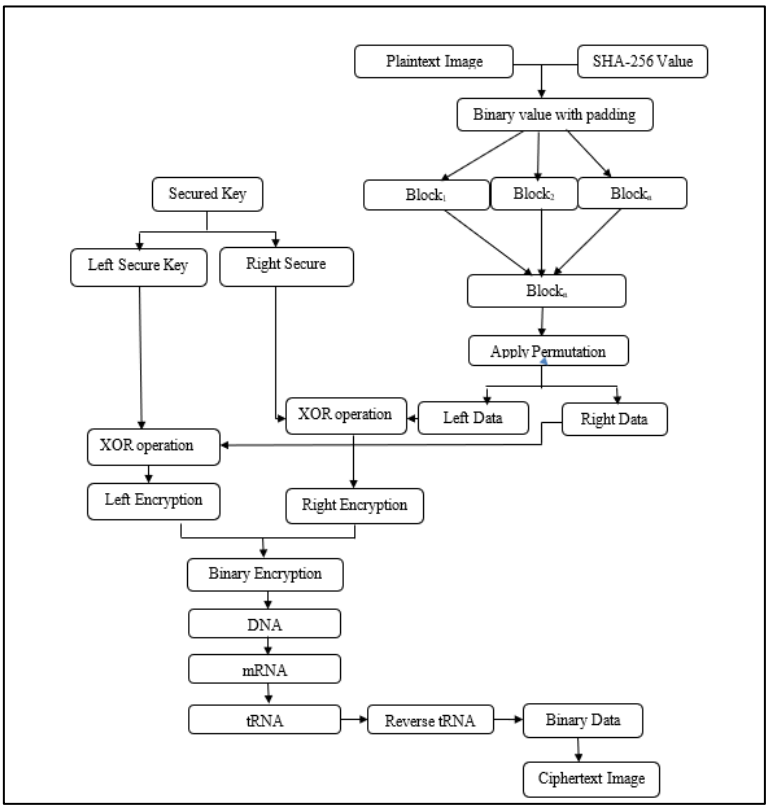


Figure 1. Workflow of Proposed Model

3.2 Algorithm for Permutation box Generation

In this research, a permutation box is generated to randomize the pixel values of each block, which introduces a high degree of unpredictability during the encryption process and ensures that whenever different blocks contain similar pixel values, their encrypted results will differ significantly. To generate the permutation box for randomization, first the seed value is computed with the summing all image data pixel values, which is deterministic, image-specific and ensuring reproducibility.

The process to generate the permutation box is as follows:

1. Input:
    - Seed: Seed value for the pseudo random number generator
    - Bit: Bits in the permutation Box.
2. Initialization:
    - Initialize an empty list $nums = \text{“ ”}$ to store randomly generated number
    - Set the variable *seed* to the input seed value

3. Generate Unique Random number:
    - Generate a pseudo random number *rand num* using the formula $rand\_num = ((seed * [48271] + [12345])//65536) mod\ bits$
    - If *rand num* is not already in $nums$, append it to $nums$
4. Return: Return the list $nums$ containing the generated permutation box.
5. End

3.3 Decryption Algorithm

Decryption of the ciphertext image is carried out on the receiver end. It is the reverse process of encryption. The detailed steps involved in recovering the plaintext image from the ciphertext image are as follows:

1. **Convert Ciphertext Image to Binary Data**: Extract the binary equivalent of the encrypted image.
2. **Reverse DNA Encoding**: Apply the reverse DNA operation to reconstruct the original binary sequence.
3. **Divide into Data Blocks:** Split the binary data into 'n' blocks, follow the similar process during encryption.
4. **Perform XOR Operations**

- Compute the XOR operation between the secured key and the binary blocks.
- Additionally, perform XOR between the binary equivalent of the hash value of the original plaintext image and the binary value of the encrypted image.

5. **Reconstruct the Image**

- Processed binary blocks are merged to reconstruct the binary data of the plaintext image.
- Convert the binary data back into an image format, restoring the original plaintext image.

4. Research Methodology

4.1 Data

For the encryption purpose, image datasets are taken from the Image Databases https://sipi.usc.edu/database/database.php and https://www.imageprocessingplace.com/ of size 512 x 512.

4.2 Encoding Rules

For encoding, first ASCII is converted into binary equivalents, and pairs of binary equivalents are transformed into DNA bases. On the basis of Watson-Crick complementary rules, the encoding binary values into DNA are given in Table 1.

Table 1. Binary to DNA Encoding [7]

Binary Digit	**DNA Base Rule**							
00	A	A	C	C	G	G	T	T
01	C	G	A	T	A	T	C	G
10	G	C	T	A	T	A	G	C
11	T	T	G	G	C	C	A	A

Let the encoding rule for the DNA bases is 00 → A, 01→ G, 10 → C and 11 →T. For example, to encrypt the character ‘A’ using DNA encoding technique, first determine its binary equivalent which is 01000001 and then mapping these binary values with DNA nucleotides and the result is ‘GAAG’.

4.3 Selection of DNA Encoding Rule

In this encryption/decryption technique, the DNA-encoded matrix is accomplished by using the different DNA encoding rules on the basis of pixel values of the image. Among the eight different DNA encoding techniques, the selection of the rule is defined as:

Rule: (Index Io (i, j) mod 8), in grayscale image the index of pixel value is given by (Index Io (i, j)). Consider that index of pixel value is 129, then 129 mod 8 is equal to 1. The Rule1: 00 = A, 01 = G, 10 = C and 11 = T is applied to convert 8-bit {10000001} into 4-bit {CAAG}.

5. Experimental Result and Security Analysis

This section analyzed the image on the basis of encryption or decryption algorithm. In this study, a different images of same size. Results are analyzed in the laptop having 16GB RAM Intel(R) Core(TM) i7-8550U, CPU @1.8GHz processor, window 10. Python program with third party library OpenCV, numpy, scipy.states, matplotlib.pyplot and standard library time and math are used for all encryption and decryption purpose. Figure 2 shows the encrypted and decrypted form of experimented images.

5.1 Encryption/Decryption Time

The stuidy tested the algorithm with key lengths of size: 128-bit, 256-bit, and 512-bit. The results indicate that while longer keys provide increase key space, they also add slight computational overhead. In this experiment, Table 2 summarizes the overall encryption and decryption time for each image with different key values. The recorded results represent the average computation time for encryption and decryption, determined by executing the process 50 times and calculating the mean of all 50 runtimes. This result shows that encryption or decryption time with 512 bit key size is approximately 30% larger than the time taken by a key with 128 bits and approximately 15% larger than a key size of 256 bits.

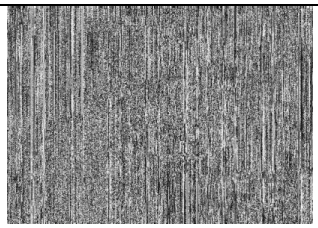
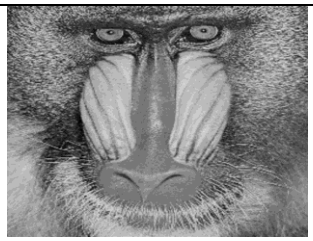
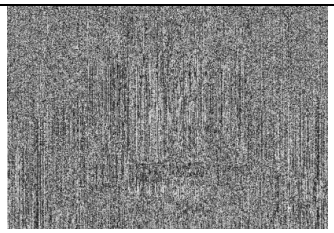
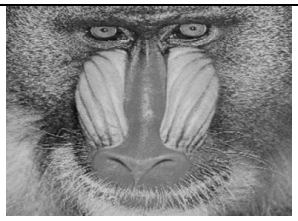
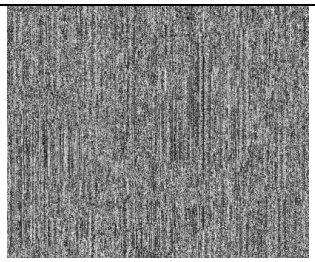
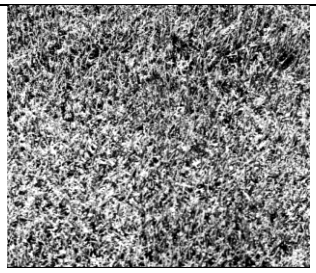
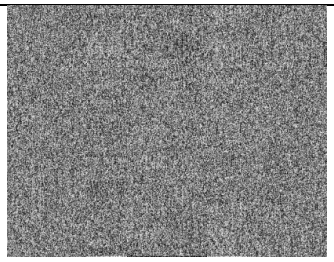
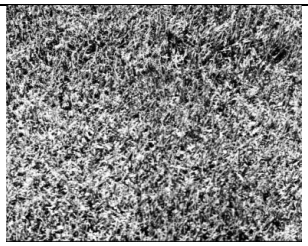
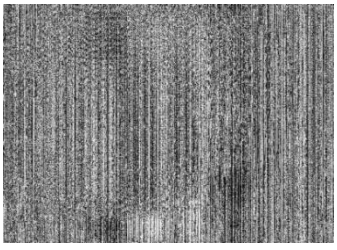
Lena_original	Lena_encrypted	Lena_decrypted
		
Baboon_original	Baboon_encrypted	Baboon_decrypted
		
Man_original	Man_encrypted	Man_decrypted
		
Grass_original	Grass_encrypted	Grass_decrypted
		
Fingerprint_original	Fingerprint_encrypted	Fingerprint_decrypted
		

Figure 2. Encrypted and Decrypted Images

Table 2. Encryption / Decryption Time with Different Key Size

Image	Encryption Time (in sec.)			Decryption Time (in sec.)		
	128 bit	256 bit	512 bit	128 bit	256 bit	512 bit
Lena	2.6825	3.0812	3.6250	3.5929	4.2819	4.9218
Baboon	3.4218	4.0312	4.6875	3.4086	4.1188	4.7343
Man	3.5000	4.2001	5.0000	3.4390	4.1171	4.8437
Grass	3.8162	4.5687	5.3750	3.7419	4.3217	5.1718
Fingerprint	3.3186	3.3647	4.6093	3.2019	3.2624	4.2656

5.2 Statistical Attack Analysis

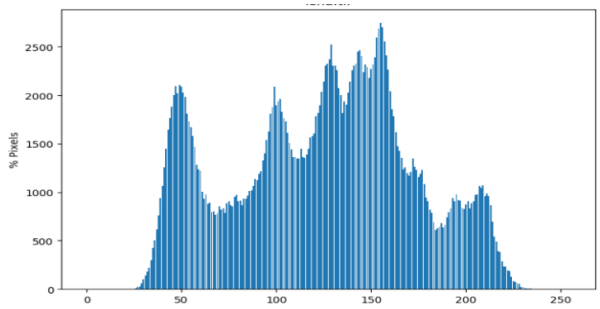
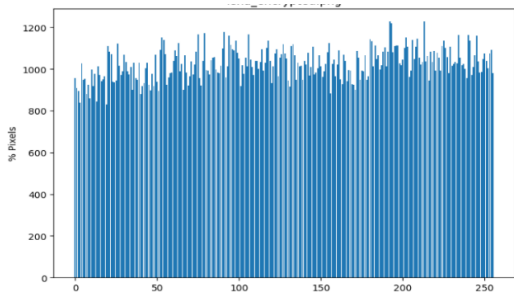
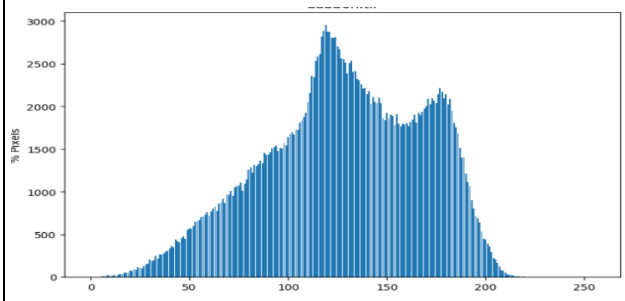
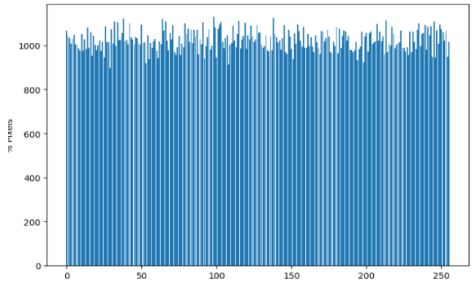
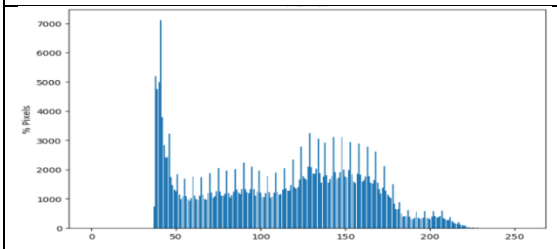
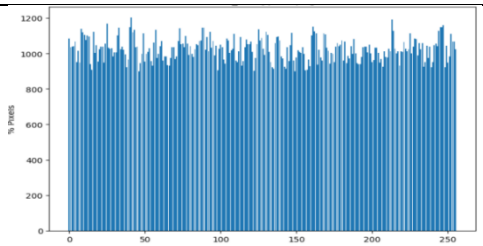
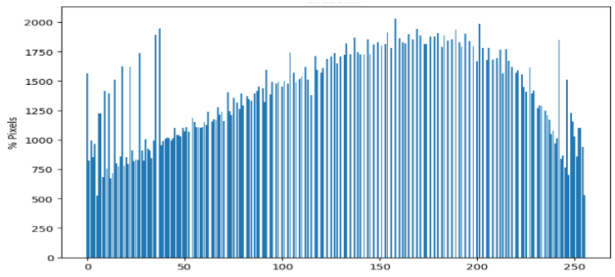
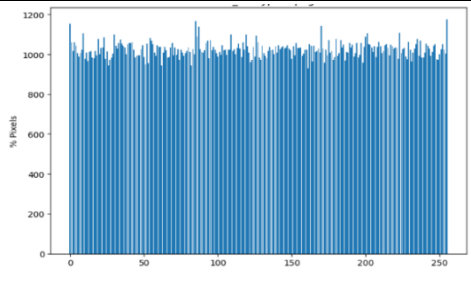
5.2.1 Histogram Analysis

Histogram plot shows the pixel density of image. Experimental result shows that the pixels in histogram of input image are not uniform randomly distributed, while pixel intensities in encrypted image histogram is uniform. This resists the statistical analysis attack. The distribution of pixel intensities in plaintext image and ciphertext image is shown in Figure 3.

Chi-square test is used to quantitatively confirm the uniform distribution of pixel values in the histogram of ciphertext image. The Chi-square definition is as follows [8]:

$$\chi^2 = \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i} \quad (1)$$

Where o_i , e_i represents the observed and expected occurrence frequency for each gray value in a 256 grayscale image. We set the significance level, α = 0.05 and compute the $\chi^2$and P-values for the test images. As shown in the Table 3, calculated chi-square values for all images are below the theoretical threshold value 293.2478. P-value is greater than significance value so, pixel distribution is considered uniform, and thus null hypothesis is accepted.

Lena_plaintext image_histogram	Lena_encrypted image_histogram
	
Baboon_plaintext image_histogram	**Baboon_encrypted image_histogram**
	
Man_plaintext image_histogram	**Man_encrypted image_histogram**
	
Grass_plaintext image_histogram	**Grass_encrypted image_histogram**
	

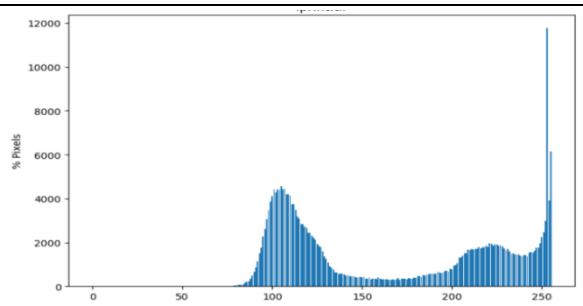
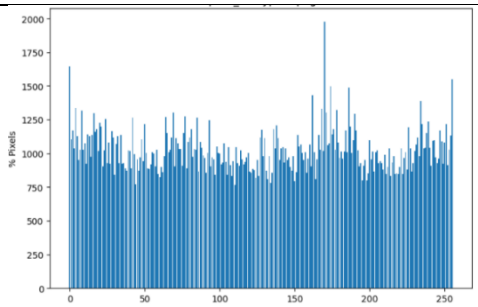
Fingerprint_plaintext image_histogram	Fingerprint_encrypted image_histogram
	

Figure 3. Histogram of Plaintext Image and Ciphertext Image

Table 3. χ^2- Test of Histograms

Image(512 x 512)	**Chi-square(χ^2)value**	**P-value**	**Decision**
Lena	258.78	0.42	H0 accepted
Baboon	244.44	0.67	H0 accepted
Man	240.35	0.73	H0 accepted
Grass	290.52	0.06	H0 accepted
Fingerprint	247.52	0.61	H0 accepted

The distribution of pixel intensities is quantified by an image histogram's variance. Lower histogram variance in the context of the encrypted image indicates more uniformity, as the pixel values are more equally distributed over the intensity range. By encrypting the same plaintext image with different secret keys and calculating the histogram variances for each resulting ciphered image, the research can verify the consistency and strength of the encryption algorithm. If the encryption algorithm is effective, the resulting ciphered images should exhibit similar levels of uniformity regardless of the key used. This would mean that the encryption process is reliably producing uniformly distributed pixel values; as a result, an attacker cannot identify the pixel pattern of the original content [9]. By using the following equation, the histogram's variance is determined:

$$var(X) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(x_i - x_j)^2 \quad (2)$$

Where, $X = \{x_1, x_2, \ldots, x_{256}\}$ is the vector of the histogram values and x_i and x_j are the numbers of pixels with gray values *i* and *j* respectively.

The variance of the histogram of the tested images with two different secured keys is listed in Table 4. From this table, it is confirmed that variance values with separate keys are comparable levels of uniformity.

Table 4. Comparison of Histogram Variance Values

Images	Variance of Original Image	Variance of ciphertext image with key1	Variance of ciphertext image with key2
Lena	633378.8750	5524.9219	5643.6172
Baboon	845463.3359	2120.5859	2129.8594
Man	1070181.9219	4082.3828	3963.4609
Grass	459898.1484	1425.83	1516.79
Fingerprint	1896152.5938	24357.70	26158.69

5.2.2 Correlation Analysis

In all directions, i.e., horizontal, vertical, and diagonal, high correlation exists between the adjacent pixels of the plaintext image. It means moving towards 1, whereas the correlation of adjacent pixels of ciphertext image has low correlation i.e., moving towards 0. If the correlations of pixels in the image are high, then attacker can easily reconstruct and break the pattern of that image. So, to maintain security, the correlation between pixels should be reduced. This analysis also shows the prevention of statistical analysis attacks. For this, different types of images, including standard images: Lena image, Baboon image, Man image, Grass image and structured image: Fingerprint image were used. Table 5 shows the correlation test result of plaintext image and ciphertext image. For correlation analysis, the following formula is used:

$$r_{x,y} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\ \sqrt{D(y)}} \quad (3)$$

Table 5. Correlation Test Result of Plain/Ciphertext Images

Image	Correlation			
	Plaintext image		Ciphertext image	
Lena	H	0.09719	H	0.01119
	V	0.9704	V	0.36394
	D	0.92409	D	0.00224
Baboon	H	0.91576	H	-0.01419
	V	0.89442	V	0.24692
	D	0.78159	D	-0.02007
Man	H	0.95611	H	0.00695
	V	0.95978	V	0.38556
	D	0.88206	D	-0.00254
Grass	H	0.7055	H	9e-05
	V	0.7859	V	0.1266
	D	0.5594	D	-0.0120
Fingerprint	H	0.9552	H	-0.0368
	V	0.9322	V	0.3184
	D	0.8530	D	-0.0567

5.2.3 Entropy Analysis

Entropy quantifies the randomness between image's pixels. In the grayscale image the intensity of pixel value is range from 0 to 255. Entropy of grayscale image is given by

$$H(x) = -\sum_{i=0}^{255} p(x_i)\log p(x_i) \quad (4)$$

Where,$x_i \in p(x)_i$ is the probability of occurrence of x_i, the theoretical value of information entropy in grayscale image is 8. In this experiment, entropy of ciphertext image is extremely close to the 8, which shows the good random distribution of pixels. So, it is concluded that the chance of attack is minimum and proposed algorithm is strong against the attack. The summary of entropy of tested image is listed in Table 6.

Table 6. Entropy Analysis of Plaintext Image and Ciphertext Image

Image	Entropy	
	Original Image	**Ciphertext Image**
Lena	7.4451	7.9977
Baboon	7.2925	7.9985
Man	7.2367	7.9972
Grass	7.5222	7.9990
Fingerprint	6.2611	7.9882

5.3 Differential Attack Analysis

5.3.1 NPCR/UACI Analysis

In Differential attack analysis, Number of Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are commonly used to evaluate the sensitivity of the encryption algorithm [10]. A good encryption technique should ensure that any tiny change in the original plaintext image should result in a significant variation in the ciphertext image. The optimal values of NPCR and UACI for the grayscale image of 256 bit pixels are 99.6094% and 33.4635%, respectively [11]. The summary report of NPCR/UACI values of this study is tabulated in Table 7.

NPCR and UACI values for the tested images are comparable with the optimal values. Thus it is concluded that the encryption algorithm is highly sensitive while change in the plaintext image and the presented algorithm can effectively resist differential attack.

Formula for calculation of NPCR is:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \; x \; N} x100\ \% \tag{5}$$

$$D(i,j) = \begin{cases} 0 & C_1(i,j) = C_2(i,j) \\ 1 & Otherwise \end{cases}$$

Formula for UACI is:

$$UACI = \sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255 * M * N} * 100\% \quad (6)$$

Where C_1 is the encrypted form of the original image $O_1$and C_2 is the encrypted image of O_2, where O_2 is obtained by a single pixel change in O_1 , M and N are the weight and height of the image, and 255 is the maximum possible intensity value.

Table 7. Summary Report of NPCR/UACI Result

Image	NPCR	UACI
Lena	99.6154	33.4510
Baboon	99.6070	33.4753
Man	99.6112	33.4594
Grass	99.3076	33.4477
Fingerprint	99.3866	33.4216

5.3.2 Avalanche Effect Analysis

Avalanche effect is a phenomenon where a tiny bit alteration in secret key causes the output data to change significantly. Because of this behavior, it is difficult for an attacker to predict the output based on partial knowledge of the input. According to [12], an algorithm is said to be secure when its avalanche effect is at least 50%. In the experiment, avalanche effect is computed as follows:

1. Convert the ciphertext image$s$ I_1, I_2 into binary form: I_1 is generated by the original secret key and I_2 is generated by tiny value change in the original secret key.

2. Compute the total number of bits: In this experiment, both images are of 512 x 512 pixels; thus total number of bits(T)= 512 x 512 x 8 =2097152 bits.

3. Compute the Hamming Distance(HD):

$H(I_1, I_2) = \sum_{i=1}^{T} \left[b_1^i - b_2^i \right]$, where b_1^i and b_2^i are the i^{th} bit of both ciphertext images respectively. Here, total flipped bits = 1081865 bits

4. Now, compute the Avalanche Effect(AE) using equation:

$$AE = \left(\frac{H(I_1,I_2}{T}\right) x100 \quad (7)$$

AE=51.58%

For all images, the avalanche effect in approximately 51% and Standard Deviation (SD) is used to analyze the variation in a number of pixels differences:

Compute the absolute difference between corresponding pixels as:

$$\mathrm{D(i,j)} = \left(\mathrm{I_1(i,j) - I_2(i,j)}\right) \quad (8)$$

Now, calculate SD as:

$\mathrm{SD} = \sqrt{\frac{1}{\mathrm{N}}\sum_{\mathrm{i=1}}^{\mathrm{N}}(\mathrm{D_i} - \mu)^2}$, μ is mean of pixel differences (9)

In the experiment, *SD* = 72.47, this result indicates the strength of encryption.

5.4 Calculation of PSNR, MSE

Peak Signal to Noise Ration (PSNR) is the ratio between the highest achievable power of a signal and the power of unwanted noise that can degrade how faithfully the signal is reproduced. PSNR is used to check the quality of ciphertext image. It is used to confirm whether noise affects the quality of an image or not [13]. PSNR is inversely proportional to the MSE of the image i.e. the encryption algorithm that produce the higher PSNR value has a better encryption technique. It is calculated as

$$\mathrm{PSNR} = 20 * \log_{10}(\frac{255}{\sqrt{\mathrm{MSE}}})\mathrm{dB} \quad (10)$$

MSE calculates the differences between two images as equation:

$$\mathrm{MSE} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{M}}\sum_{\mathrm{j=1}}^{\mathrm{N}}[\ \mathrm{I_1(i,j) - I_2(i,j)}\ ]^2}{\mathrm{M\ x\ N}} \quad (11)$$

Where $\mathrm{I_1}$ and $\mathrm{I_2}$ are the pixel values of the original and decrypted images, and (i, j) is the pixel location. M and N are the dimensions of the image. If the resultant value obtained from the above equation is closer to zero, the image is decrypted successfully, and thus, ensures that the encryption algorithm is working properly.

For $PSNR = \infty$, it must hold that the Mean Squared Error (MSE) equals 0, i.e., no difference exists between the original image $I1(i,j)$ and the decrypted image $I2(i,j)$. This is possible when the decrypted image is an exact replica of the original image as $I1(i,j) = I2(i,j), \quad \forall\, i,j.$

5.5 Key Space Analysis

Key space represents the entire number of potential keys required to decrypt the image**.** An algorithm is said to be secure if its key space is larger than the accuracy of computer 2^100. In the encryption technique, 128-bit, 256-bit, and 512-bit secure key is used. Thus, the total number of possible keys is 2^128 or 2^256 or 2^512, which is almost impossible to guess by decrypting the image; thus, the proposed algorithm resists brute force attacks.

The time required for decrypting the image information is computed based on the CPU performance capacity and the total time in seconds per year. The CPU performance capacity of this experimental computer is 1.8 GHz = 1.8×10^9 cycles per second (Hz) and the approximate time in a year is 3×10^7 seconds. Then, the total time required to decrypt the encrypted image is calculated as [14]:

$$T = \frac{n * 2^{Key\ Length}}{CPU\ speed \quad x\ Seconds\ per\ year} \tag{12}$$

$\frac{n * 2^{128\ or\ 256\ or\ 512}}{(1.8 * 10^9)\ x(\ 3 * 10^7)}$ years, n is the number of blocks.

This outcome shows that brute force attacks against such keys are impossible.

5.6 Noise Attack Analysis

For this analysis, ciphertext image is passed to a noisy channel, and then the strength of the proposed concept is verified. Here, the proposed algorithm is tested with a Gaussian attack with variance 0.01 and mean value 0 and Salt & Pepper attack with density 0.01, 0.05,and 0.1 to ciphertext image. Noisy decrypted images still have nearly the same visual information as the original plaintext image. It means that the proposed algorithm can resist the noisy attack. The effect of noise attack over the decrypted image is shown in Figure 4.

Lena_gaussian	Lena_salt_pepper0.01	Lena_salt_pepper0.05	Lena_salt_pepper0.1
			

Figure 4. Noise Effect Analysis Over Decrypted Image

In real-world decryption, using DNA-based Feistel encryption may be vulnerable to errors due to transmission noise, data loss, and key mismatches. The transmitted data may contain some extra bits or bits of data that can be flipped due to noise. These issues can cause failure in recovering the original data. In this issue, the Integration of a DNA-based encryption scheme and Reed-Solomon Error Correction Code is applicable to recover the original message at the receiver side rather than request for retransmitting the data. After the completion of this process, the hash integrity mechanism is benefitted to determine whether the received message is the same as the original or not.

5.7 NIST Analysis

Randomness at bit-level in the ciphertext image is calculated using the NIST test. It includes fifteen different tests. Here, the probabilistic value (p) is compared with the significance value (α) =0.01 in each test. If $p \geq \alpha$, it is an indication that the generated binary sequence shows randomness; otherwise, it does not [15]. Table 8 shows the detail analysis of bit level randomness in the present study.

Table 8. NIST Test Analysis

NIST Test	P-value	Result
Monobit Frequency	0.1223	PASS
Block Frequency	0.3504	PASS
Runs	0.2133	PASS
Long Runs	0.7399	PASS
Ranks	0.3504	PASS

DFT (Spectral)	0.5341	PASS
Non-Overlapping Templates	0.4179	PASS
Overlapping Templates	0.2223	PASS
Universal	0.3341	PASS
Linear Complexity	0.5341	PASS
Serial	0.5399	PASS
Approximate Entropy	0.1223	PASS
Cumulative Sums	0.2133	PASS
Random Excursions	0.0122	PASS
Random Excursions Variant	0.2102	PASS

5.8 Tamper Location Analysis

At the receiver end, the received data is susceptible to errors due to noise interference, data transmission losses, and key mismatches**.** These factors can impact the accuracy of the recovered plaintext, especially in DNA-based cryptosystems that rely on precise sequence mapping**.** During the transmission of DNA sequences**,** biological noise or channel noise may introduce mismatches in the DNA sequence. Even a minor error in the DNA sequence can lead to incorrect binary-to-text reconstruction**,** resulting in partial or complete decryption failure. To ensure the integrity and validity of the received message before decryption, the proposed approach incorporates hash value verification**.** Here, the research calculates and contrasts the original hash value of data with the calculation of the hash value of received data in the receiver end. This mechanism ensures that any alterations introduced during transmission are detected, preventing incorrect decryption. For this verification, eq (13) should be satisfied; otherwise, there arises a violation of data integrity.

$$H(P) = H(D_k(E_k(P))) \tag{13}$$

Where,

P : Data of original image

$C = E_k(P)$: Data of ciphertext image

$D_k(C)$: Decryption function from C

$H(x)$: Applied hash function to input x

5.9 Computational Complexity

To compute the computational complexity of image data with width and height of M x N, the proposed algorithm is analyzed from the beginning to the end of the encryption process, and the result is tabulated in Table 9:

Table 9. Computational Complexity Analysis of Algorithm

Activity	Action	Complexity
Binary conversion	Each pixel of the image is converted into 8 bit binary sequence	O(MxN)
Generation of Blocks	Divide binary data into same-size blocks	O(MxN)
Padding if necessary	Add 0's bit to last block	O(1)
Seed value computation	Sum of data of all images	O(MxN)
PRNG	For permutation sequence	O(MxN)
Substitution & XOR	Substitute bits using key and perform XOR	O(MxN)
DNA Encoding	Binary to DNA nucleotides	O(MxN)
DNA bases to mRNA	Replace Thymine (T) with Uracil (U)	O(MxN)
tRNA	Swapping between A& U and C &G	O(MxN)
Reverse tRNA	Replacing $U \rightarrow T$	O(MxN)
Binary Mapping	DNA to binary conversion	O(MxN)
Image construction	Encrypted binary to image	O(MxN)
Total computational complexity		O(MxN)

5.10 Performance Comparison

To perform the comparison of performance between DNA-related algorithms, results generated with the set of critical metrics including Shannon Entropy(SE), Number of Pixels

Change Rate(NPCR), Unified Average Changing Intensity(UACI), Key space, and Chi-square test are tabulated in Table 10.

Table 10. Performance Comparison of Image Encryption Algorithms

Algorithms	**NPCR**	**UACI**	**Key space**	**Entropy**	**χ^2 test**
[16]	99.6066	33.49	2^325	7.9960	--
[17]	99.60	33.46	2^209	7.9972	--
[18]	99.5376	32.5738	2^419	7.9874	--
[19]	99.58	33.43	2^335	7.9993	262.83
Proposed	99.6112	33.4617	2^512	7.9977	256.78

6 Discussion

The goal of this study is to create an encryption method for grayscale images by combining the DNA-based computing approaches with Feistel structure model of cryptography. Specific goals for this include increasing resilience to different kinds of attacks, producing more unpredictability in ciphertext image, and provide high-level security at a low computational cost. Obtained result shows that the proposed concept has effectively satisfied the objective of this study. Strong security and computational efficiency are attained by the encryption process by utilizing the DNA encoding approach, arithmetic, and complementation in conjunction with the effective and iterative Feistel structure.

Experimental metrics, such as entropy values and near-zero correlation coefficients between adjacent pixels, confirm the method's effectiveness in providing high-level security at a low computational cost. Tests reveal uniform histograms and optimal NPCR and UACI values, indicating robustness against attacks and enhanced randomness.

7 Conclusion

A new technique of grayscale image encryption cryptosystem is introduced with the concept of Feistel structure with random selection of DNA sequencing rule and pseudorandom

number generated key. Here, the original image's permuted data is decoded into a binary image, and the XOR operation is then performed using the key at last obtained data block is translated into ciphertext image with the DNA translation technique and the reverse process is carried out to obtain the plaintext image from the ciphertext image. The experimental result shows that the proposed technique significantly enhances and maintains the security of an image from various types of attack as compared with ideal values of the test during transmission. Additionally, the outcome demonstrates that the suggested method maintains nearly the same level of image quality as the original image, even while transmitted through a noisy channel.

References

[1] Arroyo, David, Gonzalo Alvarez, José María Amigó, and Shujun Li. "Cryptanalysis of a family of self-synchronizing chaotic stream ciphers." Communications in Nonlinear Science and Numerical Simulation 16, no. 2 (2011): 805-813.

[2] Namasudra, Suyel. "A secure cryptosystem using DNA cryptography and DNA steganography for the cloud-based IoT infrastructure." Computers and Electrical Engineering 104 (2022): 108426.

[3] Adleman, Leonard M. "Molecular computation of solutions to combinatorial problems." science 266, no. 5187 (1994): 1021-1024.

[4] Wang, Xing-Yuan, Ying-Qian Zhang, and Xue-Mei Bao. "A novel chaotic image encryption scheme using DNA sequence operations." Optics and Lasers in Engineering 73 (2015): 53-61.

[5] Hagras, Tarek, Doaa Salama, and Hassan Youness. "Anti-attacks encryption algorithm based on DNA computing and data encryption standard." Alexandria Engineering Journal 61, no. 12 (2022): 11651-11662.

[6] Thabit, Fursan, Sharaf Alhomdy, and Sudhir Jagtap. "A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions." International Journal of Intelligent Networks 2 (2021): 18-33.

[7] Bhaya, Chiranjeev, Arup Kumar Pal, and SK Hafizul Islam. "A novel image encryption and decryption scheme by using DNA computing." In Advances in Computers, vol. 129, pp. 129-172. Elsevier, 2023.

[8] Wang, SiCheng, ChunHua Wang, and Cong Xu. "An image encryption algorithm based on a hidden attractor chaos system and the Knuth–Durstenfeld algorithm." Optics and Lasers in Engineering 128 (2020): 105995.

[9] Zhang, Ying-Qian, and Xing-Yuan Wang. "A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice." Information Sciences 273 (2014): 329-351.

[10] Zhu, Shuqin, and Congxu Zhu. "Secure image encryption algorithm based on hyperchaos and dynamic DNA coding." Entropy 22, no. 7 (2020): 772.

[11] Hu, Ting, Ye Liu, Li-Hua Gong, Shao-Feng Guo, and Hong-Mei Yuan. "Chaotic image cryptosystem using DNA deletion and DNA insertion." Signal Processing 134 (2017): 234-243.

[12] Mohamed, Kamsiah, Mohd Nazran Mohammed Pauzi, Fakariah Hani Mohd Ali, and Suriyani Ariffin. "Analyse On Avalanche Effect In Cryptography Algorithm." European Proceedings of Multidisciplinary Sciences (2022).

[13] Akkasaligar, Prema T., and Sumangala Biradar. "Secure medical image encryption based on intensity level using Chao's theory and DNA cryptography." In 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1-6. IEEE, 2016.

[14] Şatir, Esra, and Oğuzhan Kendirli. "A symmetric DNA encryption process with a biotechnical hardware." Journal of King Saud University-Science 34, no. 3 (2022): 101838.

[15] Abdelfatah, Roayat Ismail. "Secure image transmission using chaotic-enhanced elliptic curve cryptography." IEEE Access 8 (2019): 3875-3890.

[16] Wang, Xingyuan, and Yining Su. "Image encryption based on compressed sensing and DNA encoding." Signal Processing: Image Communication 95 (2021): 116246.

[17] ur Rehman, Aqeel, Xiaofeng Liao, Ayesha Kulsoom, and Syed Ali Abbas. "Selective encryption for gray images based on chaos and DNA complementary rules." Multimedia Tools and Applications 74 (2015): 4655-4677.

[18] Liu, Hongjun, and Xingyuan Wang. "Image encryption using DNA complementary rule and chaotic maps." Applied Soft Computing 12, no. 5 (2012): 1457-1466.

[19] Chai, Xiuli, Zhihua Gan, Ke Yuan, Yiran Chen, and Xianxing Liu. "A novel image encryption scheme based on DNA sequence operations and chaotic systems." Neural Computing and Applications 31 (2019): 219-237.

Author's biography

Madhav Dhakal received M.Sc. in Computer Science and Information Technology from Tribhuvan University, Nepal. He is a Ph.D. scholar in Computer Science and Information Technology at Tribhuvan University. He is currently an Assistant Professor of Computer Science and Information Technology at Mid-West University, Nepal.

Subarna Shakya received his M.Sc. and Ph.D. degrees in Computer Engineering from Lviv Polytechnic National University, Ukraine. He is currently a Professor in the Department of Electronics and Computer Engineering at the Institute of Engineering, Pulchowk Campus, Tribhuvan University, Nepal. He served as Visiting Professor at Brown University, Rhode Island, USA. He reviews numerous national and international journals.