Hybrid Image Encryption using Non-Adjacent Bits Dynamic Encoding DNA with RSA and Chaotic Systems

Marwa A. Elmenyawi[1], Nada M. Abdel Aziem[2]
Benha Faculty of Engineering, Benha University Benha, Egypt[1]
Arab Academy for Science, Technology and Maritime Transport - Arab League[1, 2]

***Abstract*—Image encryption is a crucial aspect that helps to maintain the images' confidentiality and security in diverse applications. Ongoing research is focused on improving the efficiency and effectiveness of encryption. Image encryption has many practical applications in today's digital world, such as securing confidential images transmitted over networks, protecting sensitive personal information stored in images, and ensuring the privacy of medical images. The suggested work represents a breakthrough in image encryption by proposing a model that leverages the power of DNA, RSA, and chaos. This model has three phases: key generation, confusion, and diffusion. The key generation phase employs a hash function and hyperchaotic technique to generate a strong key. During the confusion phase, the positions of pixels are rearranged, either at the image level or within blocks, using the Duffing chaotic map. Once the scrambling level is determined, each pixel undergoes two successive scrambling steps, with Henon and Arnold's chaotic map to change its location. During the diffusion phase, the encryption model employs a two- approach to ensure maximum security. Firstly, it utilizes dynamic DNA cryptography for non-adjacent bits, followed by robust RSA cryptography. The experimental results indicate that the model possesses a strong security level randomness and can withstand different attacks.**

Keywords—Cryptography; image encryption; hash function; chaotic map; DNA encoding; DNA operations; RSA algorithm

I. Introduction

In today's digital era, digital images are widely used for personal, professional, or commercial purposes. Therefore, these images require protection from unauthorized access. Image encryption and information hiding are the two basic approaches for securing digital images. Image encryption prevents hackers from recognizing the images by employing complex mathematical operations to transform image data into an unreadable form. Therefore, the hacker's attempts are wasted. There is a need to design and implement an algorithm characterized by its security and efficiency to succeed in withdrawing the different attacks. There are two phases in image encryption: diffusion and scrambling. The pixel positions are altered during the scrambling phase, whereas the pixel values are changed during the diffusion phase.

Various methods are employed during the confusion phase. Some of these works used the Arnold transform [1-3], Zigzag transformation [4,5], Fisher-Yates [6], and Josephus traversal [7]. Other works implemented scrambling over two steps, such as L-shape and Arnold transforms as in [8], new filling curve design and Josephus traversal [9].

In terms of diffusion, S. Wang. et al. [10] suggested using the DNA sequence in the diffusion phase. Four sequences were derived from a 4D chaotic system and utilized to select the rules for encoding, computing, and decoding. They utilized higher dimensional chaos to offer a large key space. J. Yu et al. [11] began with the diffusion phase. They encoded the three matrices of RGB image using DNA sequence where a chaotic system chooses the rule. A new operation, known as DNA triploid mutation, was introduced to achieve cryptographic translation of DNA bases. Finally, they permuted the image using row-column permutation. C. Zou et al. [12] utilized two types of DNA strands: long and short. The image was permuted using two short DNA strands, while the long DNA strand was used in the diffusion stage. If the DNA sequence follows the property of the Watson-Crick base pairing, the XOR operation of DNA is performed; otherwise, the DNA addition operation is used.

B. Jasra and A. Moon [13] split the color image into three planes and encoded each plane using a DNA sequence based on the chosen row-level. A substitution algorithm relies on elliptic curves to accomplish effective encryption and authentication. J. Wang et al. [1] suggested a new type of chaos; Logistic-Sine self-embedding. They proved this type's chaotic features and adopted a 0-1 test to find the chaos's presence in the time series. They encrypted the plain image using a Logistic-Sine self-embedding chaotic system. Similarly, X. Li [14] introduced another chaotic sequence, which was 5D, and they showed that the 5D chaotic did not have a prominent Lyapunov exponent yet possessed several good characteristics. The 5D chaotic sequence was utilized to choose the DNA encoding, computing, and decoding.

The encryption model presented in [15] depended on a fused magic cube produced by fusing two magic cubes. The cipher image's pixels value was obtained from the plain image's pixels value by employing the fuse magic cube. J. Zheng and Q. Zeng [16] constructed an S-box using the obtained key from the Logistic map, generating a 16 x 16 matrix ranging from 0 to 255 with no repeated values. The image was diffused by traversing the scrambled image in order according to the generated keys.

A technique for encrypting color images was introduced in [17], which employs 3D chaos, RSA, DNA, and LSB. The image is initially encrypted using a DNA method and Lorenz chaotic map. The secret key is then encrypted employing RSA and hidden within a cipher image using LSB. In another approach proposed by M. Liu and G. Ye [18], image encryption is achieved by utilizing dynamic DNA alongside a hyperchaotic system. The dynamic DNA coding selects DNA rules in a randomized manner, guided by the employed chaotic map. This study also incorporates RSA to protect the secret key's confidentiality during transmission and management. U. Mir et al. [19] introduced an encryption method for color images using RSA and chaos in the domain of Hartley. The image is first ciphered utilizing RSA and then transformed from the time to frequency domain using a Hartley domain.

K. Jiao et al. [20] introduced an encryption approach combining RSA and a generalized Arnold chaotic map. The RSA algorithm obtains the map parameters, generating the keystream for a diffusion operation on the plaintext image. The confusion operation is then employed to conceal the image data and produce the cipher image. Babu M et al. [21] used chaotic Maps, RSA, and DNA sequences to encrypt images. This paper divided the plain image into several blocks. Secondly, different encryption schemes were utilized on each block, such as Secure Force, DNA Sequence, Arnold Map, and RSA encryption. Thirdly, a discrete cosine transformation algorithm was applied to the merged blocks, after which an XOR operation was conducted with a randomly generated key to produce the encrypted image.

This paper aims to enhance image encryption security by minimizing pixel correlation, maximizing randomness and unpredictability, and withstanding various types of attacks. The proposed encryption approach has three phases: key generation, confusion, and diffusion. The integration of different chaotic maps leads to a substantial expansion of the key space and makes encryption impervious to brute-force attacks. A robust key generation process achieves cryptosystem robustness against various attacks. Chaos and hash functions, SHA and MD5, produce the encryption key. The advantages of the SHA function are its irreversibility and a one-time pad key, while the chaos is characterized by randomness and unpredicted ability. The final encryption key is obtained by applying these hash functions to the user-specified key and the plaintext image. The encryption security approach is strengthened by utilizing the plaintext image and the user key.

Scrambling a pixel's locations can be done over the whole image or the divided blocks of the whole image, depending on the Duffing chaotic map. Moreover, two levels of confusion are implemented using Arnold and Henon's chaotic maps. The scrambling phase achieves high randomness between the pixels and decreases the correlation between the pixels to the minimum compared to the previous research, as illustrated by the results of the suggested method. The final phase is diffusion, implemented over two steps: DNA and RSA. In the DNA algorithm, the pair of bits to be replaced with the DNA sequence is not successive as customary in state-of-the-art research. The proposed algorithm incorporates dynamic DNA to select various rules for each pixel and DNA computations to improve its efficiency. The encoding, computation, and decoding rules are determined using a 4D hyperchaotic sequence.

The second diffusion level is to implement the RSA algorithm, which is different from the state-of-the-art research where most of the paper used RSA along with DNA utilized RSA outputs as the initial values of chaotic sequences. Despite using multiple steps and various types of chaotic maps, we tried to keep the algorithm's runtime comparable to previous research. We conducted experimental testing and analysis to demonstrate the proposed approach's superiority and feasibility, showcasing its resilience against multiple attacks such as differential, plaintext, brute-force, occlusion, and noise attacks. Moreover, the correlation is lower than in the most recent research.

The paper's structure is as follows: the second section provides an overview of chaos and DNA cryptography. The third section outlines the suggested approach for image encryption/decryption, followed by the fourth section thoroughly explores the results, conducts in-depth analysis, and compares them with similar research. Lastly, the fifth section introduces the paper's conclusion.

II. Background

A. *Chaotic*

Chaotic systems are characterized by unique features appropriate in encryption, like sensitivity to initial conditions, irregular behavior, and unpredictability. Therefore, using the chaotic sequence in the encryption system can give a high-security degree and robustness against attacks. Our suggested algorithm employs various types of chaotic systems at different stages to increase the key space and enhance security.

The 1D logistic map [22] generates chaotic dynamics in a discrete-time system. The Logistic chaotic map provides high speed, low arithmetic operations, and low computational overhead. It is a nonlinear recursive function defined by Eq. (1).

$$x_{n+1} = rx_n\,(1 - x_n) \tag{1}$$

Where r is a parameter that determines the map behavior. x_0 should be ϵ[0,1]and r has to be within interval 0<r⩽4 to produce the chaotic behavior.

The Henon Chaotic Map [19] refers to a discrete-time dynamical map in 2D. Eq. (2) and (3) define the equations of the Henon map:

$$x_{n+1} = 1 - a\ x_n^2 + \ y_n \tag{2}$$

$$y_{n+1} = bx_n \tag{3}$$

To attain chaotic behavior in the Henon Chaotic Map, the control parameters a and b need to be assigned the values (1.4, 0.3).

The Arnold Chaotic Map [23] is often employed to scramble and alter pixel locations. It is described in Eq. (4) and (5).

$$x_{n+1} = (2x_n + \ y_n) mod\ m \tag{4}$$

$$y_{n+1} = (x_n + y_n) mod\ m \tag{5}$$

The "mod m" operation ensures that the coordinates do not exceed the image size.

Quantum Chaotic Map [24] is a classical dynamical system that is described to develop the function for solving computing of the quantum. It is used to generate many random numbers. The quantum map mathematical expression is given in Eq. (6)-(8).

$$x_{n+1} = r(x_n - |x_n|^2) - r y_n \quad (6)$$

$$y_{n+1} = -y_n e^{-2\beta} + e^{-\beta} r[(2 - x_n - x_n^*)y_n - x_n z_n^* - x_n^* z_n] \quad (7)$$

$$z_{n+1} = -z_n e^{-2\beta} + e^{-\beta} r[2(1 - x_n^*)z_n - 2x_n y_n - x_n] \quad (8)$$

Where x ϵ [0,1], y ϵ [0, 0.1], z ϵ [0, 0.2], r ϵ [0,4], β ϵ [6, ∞) ,$x^*$and z^* are complex conjugates of x and z.

Duffing Chaotic [23] produces chaotic dynamics in a discrete-time system. The Eq. (9) and (10) represent this map. It can be utilized to design damped oscillators.

$$x_{n+1} = y_n \quad (9)$$

$$y_{n+1} = -b\, x_n + a\, y_n - y_n^3 \quad (10)$$

Its control parameters, a and b, should be 2.75 and 0.2 to maintain chaotic behaviour in the Duffing Chaotic system.

B. DNA

Images are encrypted using nucleic acid bases via a DNA encryption system, which subsequently carries out several additional DNA operations. The four nucleic acid bases identified by the Watson-Crick basic pairing principles are C (Cytosine), A (Adenine), T (Thymine) and G (Guanine). Every two bases complement each other; A is the complement of T and likely C and G. These four bases can be represented in binary using the numbers 00, 01, 10, and 11. There are 24 possible combinations for the DNA bases represented in binary. However, only eight satisfy the Watson-Crick complementary rule. Table I lists these eight coding principles. An illustration of the encoding process is as follows: Suppose a pixel value of 90 is represented in binary as 01011010. This number is then encoded as the AATT sequence if rule 4 is applied. There are a different number of operations according to the binary system. The operations used in this paper are addition, subtraction, multiplication, XNOR, XOR, right rotate and left rotate [10].

TABLE I. THE RULES OF DNA

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

III. PROPOSED CRYPTOSYSTEM APPROACH

This section will showcase the construction of our suggested algorithm, which comprises three phases: key generation, confusion, and diffusion. The complete layout of our suggested algorithm is introduced in Fig. 1, and each phase will be elaborated in further detail in the preceding subsections. Our suggested approach is applicable to both grayscale and color images. For a color image with a size of MxNx3, the three colors are separated into three matrices with an M x N size, and each matrix is processed as a plain image.

A. Key Generation

The immunity of the cryptosystem to various attacks is based on producing a solid key. The suggested algorithm employs the hash functions and chaotic sequence to generate the encryption key. The hash function has the advantage of its irreversibility and is a one-time pad key, while the chaos is characterized by randomness and unpredictability.

We chose to use MD5 and SHA-256 as the hash functions. The MD5 is faster than the SHA-256, but the SHA-256 is more complex than MD5. A hash value H_k is generated based on combining the original image and a random user-specified key. Generating the key from the original image enables the system to withstand chosen/known-plaintext attacks. The final key value, K_i ,consists of 32 bits decimal.

The final key is used to generate the second input of DNA operation using the hyperchaotic map, Eq. (11)-(14). To enhance the approach's resistance to brute force attacks, we opted for the hyperchaotic map, which offers a large key space. The hyperchaotic initial values are calculated according to Eq. (15)-(18) and utilizing the final key, K_i . The matrix M x N constitutes the representation of the second input, possessing identical dimensions to those of the original image.

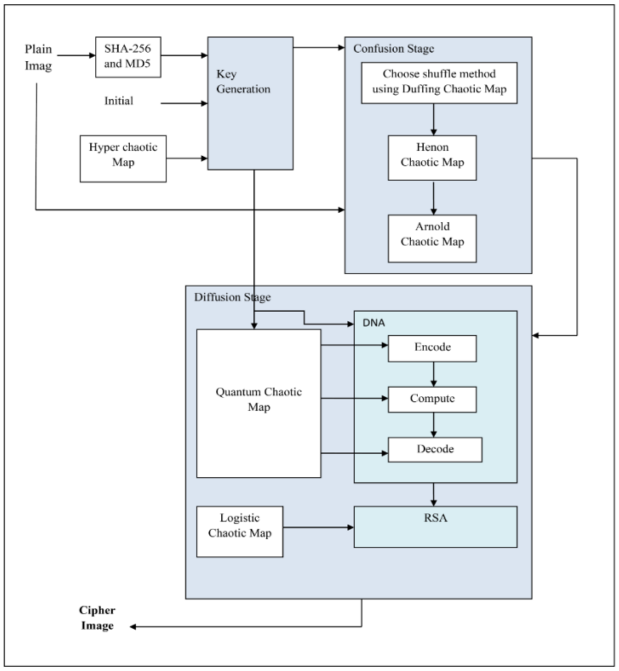


Fig. 1. The proposed image encryption approach's block diagram.

$$x_{n+1} = a\,(x_n - \; y_n) + r\,w_n \tag{11}$$

$$y_{n+1} = b\,x_n + \; x_n z_n - q\,y_n \tag{12}$$

$$z_{n+1} = \; -x_n y_n - c\,z_n \tag{13}$$

$$w_{n+1} = d\,x_n z_n - k\,w_n \tag{14}$$

$$x_0 = \frac{(((((k_1 \oplus k_2) \wedge k_3) \oplus k_4) \wedge k_5) \oplus k_6)}{256} \tag{15}$$

$$y_0 = \frac{(((((k_7 \oplus k_8) \wedge k_9) \oplus k_{10}) \wedge k_{11}) \oplus k_{12})}{256} \tag{16}$$

$$z_0 = \frac{(((((k_{13} \oplus k_{14}) \wedge k_{15}) \oplus k_{16}) \wedge k_{17}) \oplus k_{18})}{256} \tag{17}$$

$$w_0 = \frac{(((((k_{25} \oplus k_{26}) \wedge k_{27}) \oplus k_{28}) \wedge k_{29}) \oplus k_{30})}{256} \tag{18}$$

Where $k_i$is XOR between the generated hash value and the initial secret key.

B. *Confusion Stage*

This stage aims to change the pixel locations. Two alternatives were presented to achieve this goal: either process each pixel across the entire image or partition the image into blocks and rearrange the position of pixels within each block. The option number is calculated by Eq. (19), where y is determined from Duffing chaos in Eq. (9) and (10). Eq. (20)-(23) are utilized to compute the Duffing map's initial values and control parameters.

$$sh_{no} = floor(mod(avg(y), 2)) \tag{19}$$

$$x_0 = \frac{k_{19} \oplus k_{20} \oplus k_{21} \oplus k_{22} \oplus k_{23}}{16} \tag{20}$$

$$y_0 = \frac{k_{25} \oplus k_{26} \oplus k_{27} \oplus k_{28} \oplus k_{29} \oplus k_{30} \oplus k_{31}}{64} \tag{21}$$

$$a = \frac{(k_{15} \oplus k_{16}) \oplus (k_{17} \wedge k_{18})}{16} \tag{22}$$

$$b = \frac{k_1 \oplus k_2 \oplus k_3 \oplus k_4 \oplus k_5 \oplus k_6}{512} \tag{23}$$

If the option number is one, we will change the pixel locations on the image's level depending on two chaotic systems: Arnold and Henon maps. The first step is to convert the original location, i and j, of each pixel to i_h and j_h locations using Henon, as shown in Eq. (24) and (25). The final locations, i_f and j_f , are generated using Arnold chaotic using the generated locations from the previous step, as illustrated in Eq. (26) and (27).

$$i_h = mod(round(|1 - a\,i^2 + j|), M) + 1 \tag{24}$$

$$j_h = mod\;(round\;(\;i + c), N) + 1 \tag{25}$$

$$i_f = mod(i_h - 1 + \; j_h - 1, M) + 1 \tag{26}$$

$$j_f = mod(i_h - 1 + 2(j_h - 1), N) + 1 \tag{27}$$

The other alternative is to modify the pixel location at the block level. The initial step involves partitioning the image into four blocks and shuffling their positions. Then, the pixel location within the block is rearranged by applying the same procedures as in the first method.

C. *Diffusion Stage*

The confusion stage alone did not meet the encryption security requirements, necessitating the addition of a diffusion stage. The diffusion stage effectively conceals the original image information and increases attack resistance. In our proposed system, this stage consists of two steps, DNA and RSA, to offer more security to the cryptosystem.

1) DNA: In the DNA step, the encryption algorithm converts each pixel into its corresponding binary representation. Then, every pair of bits is substituted with a DNA sequence of four bases based on one of the eight DNA encoding rules shown in Table I. The DNA rule choice is made dynamically and calculated from Eq. (28), depending on the quantum map. The generated key is employed to obtain the quantum sequence's initial values, as shown in Eq. (29)-(31). The number of the generated quantum sequences equals the number of pixels to confuse the attacker, which rule is chosen and makes the cryptographic system unpredictable. Unlike the previous research, we did not replace the adjacent bits in each pixel; instead, we constitute a pair from the bit i and bit i+2, not i+1.

$$R_i = floor(8|x_i|) + 1 \tag{28}$$

$$x_0 = \frac{((((((((k_1 \oplus k_2) \oplus k_3) \oplus k_4) \oplus k_5) \oplus k_6) \oplus k_7) \oplus k_8) \oplus k_9)}{512} \tag{29}$$

$$y_0 = \frac{((((((((k_{10} \oplus k_{11}) \wedge k_{12}) \oplus k_{13}) \wedge k_{14}) \oplus k_{15}) \wedge k_{16}) \oplus k_{17}) \wedge k_{18})}{512} \tag{30}$$

$$z_0 = \frac{((((((((k_{19} \oplus k_{20}) \wedge k_{21}) \oplus k_{22}) \wedge k_{23}) \oplus k_{24}) \wedge k_{25}) \oplus k_{26}) \wedge k_{27})}{512} \tag{31}$$

$$op_i = 7 * floor(10^8 |z_i|) \tag{32}$$

An illustration of the encoding process is as follows: Suppose a pixel value of 90 is represented in binary as 01011010. This number is then encoded using rule 4 as the GCCG sequence, not AATT. Here, the first A is determined using the first and third bit, which gives pair 11, equivalent to A according to rule 4.

After the encoding step, we apply a DNA operation on the encoded DNA sequence and the generated input from the previous stage to obtain another DNA sequence. The DNA operation, op_i, is selected based on the quantum sequence, as shown in Eq. (32). Finally, we perform the DNA decoding, which is the encoding reverse. The DNA sequence is converted to its equivalent binary using the rules in Table I. The rule is selected according to the quantum sequence shown in Eq. (28). The corresponding binary bits are reordered in odd and even positions, not as successive bits. For example, if the sequence is ATCG and the used rule is 8, the binary equivalent is 11001001, then it is reordered to 10101001. Finally, the decimal equivalent of the binary number is produced to use as the input of the RSA step.

2) RSA: The RSA step involves generating random prime numbers p and q using a Logistic chaotic map. The public and private keys are then calculated based on the result generated from the Logistic chaotic map, as shown in Algorithm (1). Afterwards, image encryption is achieved using the public key, and decryption using the private key is illustrated in Algorithm (2).

Algorithm (1) Public and private key generation

```
Input: The prime numbers (p, q)
Output: Public key (PU), Private key (PR)
$x=q*p$
$\Phi(x)=(q-1)*(p-1)$
Choose e in condition that $1<e<\Phi(x)$ and $\gcd(\Phi(x),e)=1$
$d \equiv e^{-1} (\mathrm{mod}(\Phi(x))$
$PU= \{x,e\}$
$PR=\{x,d\}$
```

Algorithm (2) Encryption/Decryption of the image

```
Input: Public key (PU), Private key (PR)
Output: Cipher image (C), Plain image(P)
$C=P^{e}$ mod x
$P=C^{d}$ mod x
```

D. Image Decryption

In the decryption phase, the encrypted image is converted back to its initial form, and the decrypted image reproduces the original image. This process follows a pattern inverse to that of encryption. The encrypted image first undergoes the diffusion phase, utilizing RSA followed by DNA techniques. Next, it enters the confusion phase, where Arnold and Henon chaotic maps are used, and the key generated from chaotic and hash functions, MD5 and SHA-256, is applied. The final output of this process is the original image.

IV. Experimental Results and Analysis

The effectiveness and robustness of the suggested algorithm through various security tests are demonstrated in this section. Using MATLAB R2021a, the algorithm was simulated on a computer with an Intel(R) Core (TM) i5-6200U CPU @ 2.3GHz 2.4GHz and 8 GB of memory. There are different colors and grey images with different sizes utilized for testing. The grey images used in testing are Male 1024 x1024, Lena 512x512, Barbara 512x512, Lake 512x512, Cameraman 256x256, and Kitten 256x256. The color images are Shreveport 1024x1024, Lena 512x512, Baboon 512x512, Lena 256x256, and Couple 256x256. Most of the images used in the study were sourced from the USC-SIPI database. Illustrations depicting the plaintext, cipher, and decrypted images are presented in Fig. 2 and Fig. 3 (a, c, and e). The right keys can be utilized to restore all images during the decryption tests. Any change in the secret keys, even if slight, leads to incorrect image decrypting, as will be proved in the following subsections. According to [25], there are four categories to estimate the proposed algorithm's performance.

- The visual perception evaluation

This category aims to generate an uncorrelated cipher image from the plain image. The performance metrics included in this category are PSNR, MSE, Correlation, Entropy, and Histogram analysis. Another test carried out within this category is the similarity test using the SSIM performance, which is adopted to evaluate the matching degree between the plaintext and cipher images.

- The high-performance evaluation

The goal of this category is to evaluate the diffusion characteristics. The tests to accomplish this goal are NPCR and UACI.

- The processing time.

- The strength of the cryptosystem evaluation

The cryptosystem strength is measured through key space, key sensitivity, and attack resistance.

A. The Visual Perception Evaluation

1) Histogram analysis: The distribution of tones inside an image, made up of pixels with various grey values, is essential. The histogram, which can adequately depict the amount of each pixel value, represents the tonal distribution of an image. An image's histogram gives statistical information that can be used to assess how strong an encryption system stands up to statistical attacks. Any plain image's histogram is covered by an angled or curved bar. There is much information in this bar. Hackers can use this bar to further their harmful goals. The cipher's task is to modify the pixel intensity values to produce a histogram with a uniform bar above it. Any information leaking in the image is greatly discouraged by the histogram's uniformity of the bar.

Additionally, it intimidates hackers and prevents them from succeeding with histogram attacks. Fig. 2 and 3(b,d) introduces the plain and encrypted images histogram. Apparently, the pixel distribution in cipher images is relatively uniform across all channels, but plain images have several peaks.

2) PSNR and MSE: Several measures are used to evaluate image quality, including PSNR and MSE, which quantify the level of difference between the original and cipher images. Equations (33-34) provide the mathematical expressions for PSNR and MSE.

$$PSNR = 10\ x\ log_{10} \frac{p^2}{MSE} \tag{33}$$

$$MSE = \frac{1}{M\ x\ N} \sum_{i=1}^{M} \sum_{j=1}^{N} |O(i,j) - C(i,j)|^2 \tag{34}$$

The maximum pixel value, represented by 8 bits, is denoted by p = 255. The plaintext and cipher images are denoted by O(i,j) and C(i,j). The higher MSE and lower PSNR values determine the method's efficiency and security. Table II presents the PSNR and MSE values for grayscale and color image channels. As elaborated in Table II, the suggested algorithm achieves low PSNR and high MSE values, indicating its strong security and efficiency. Moreover, Table III compares the suggested algorithm's performance with previous research, pointing to its better performance relative to other techniques concerning both PSNR and MSE.

3) SSIM: SSIM is a test used to indicate how much the cipher image is similar to the plaintext image depending on the amount of structural information modification of the plaintext image. The SSIM is obtained as indicated in Eq. (35). A decrease in the similarity index indicates a decrease in the match between the original and encrypted images and an increase in the degree of changed structural information.

The calculated values of the SSIM index are introduced in Table II. Table III shows that lower SSIM values close to zero indicate the lower similarity of the suggested algorithm. Moreover, the suggested algorithm gives better results than the other research, meaning the proposed algorithm offers less similarity to the other research, as indicated in Table III.

$$\begin{cases} SSIM(O,E) = \frac{(2\,\mu_O\mu_E + C_1)(2\,\sigma_{OE} + C_2)}{(\mu_O^2\mu_E^2 + C_1)(\sigma_O^2\sigma_E^2 + C_2)} \\ C_1 = (K_1\,L)^2 \\ C_2 = (K_2\,L)^2 \end{cases} \tag{35}$$

Where the $\mu_O$and μ_E are the original and cipher images mean. σ_O^2 and σ_E^2 are the plaintext and cipher image variance. The original and cipher images covariance is denoted by σ_{OE}. L is set to 255, while $K_1$and K_2 are set to 0.01 and 0.03, respectively.

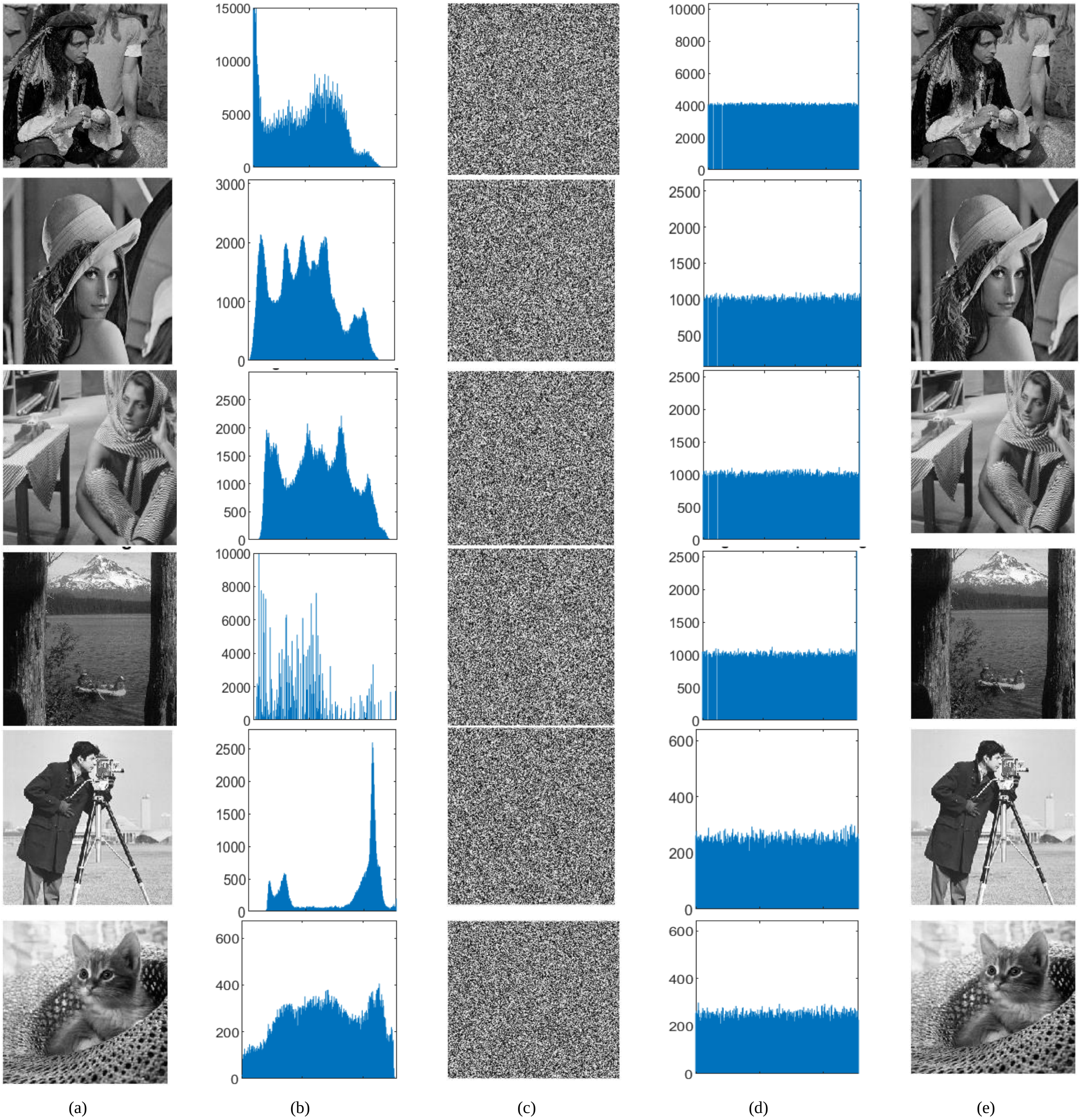


Fig. 2. The results of grayscale images (a) plain image (b) plain image histogram (c) cipher image (d)) cipher image histogram (e) decrypted image.

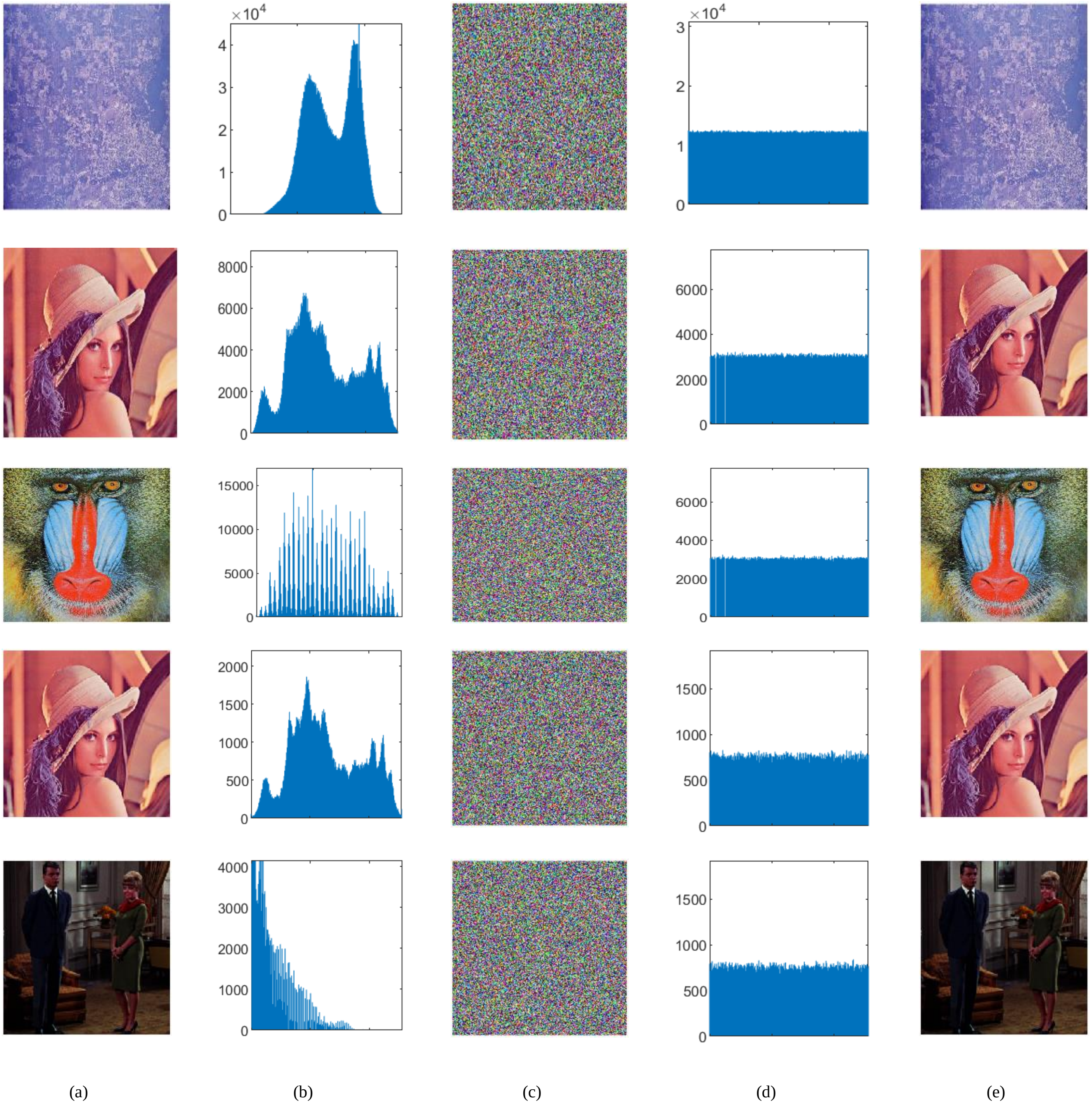


Fig. 3. The results of color images (a) plain image (b) plain image histogram (c) cipher image (d)) cipher image histogram (e) decrypted image.

TABLE II. THE MSE, PSNR, MAE, AND SSIM PERFORMANCE

Image	Color	MSE	PSNR	SSIM
Male 1024 x1024	***Grey***	10475	7.9293	0.0001
Barbara512 x 512	***Grey***	8589.41	8.7912	0.0002
Lake 512 x 512	***Grey***	10893.21	7.7592	-0.0004
Kitten 256 x256	***Grey***	9685.53	8.2696	0.0005
Cameraman256 x 256	***Grey***	11675.60	7.4580	-1.3E-05
Shreveport 1024 x 1024	***R***	6463.01	10.0265	0.0005
	G	6151.77	10.2408	-0.0002
	B	9052.82	8.5630	-0.0001
Baboon 512 x 512	***R***	8627.04	8.7722	0.00097
	G	7902.03	9.1534	0.0006
	B	9950.34	8.1524	0.0003
Lena 256 x 256	***R***	10666.34	7.8506	-0.0006
	G	9048.21	8.5652	-0.0004
	B	7025.62	9.6640	0.0030
Couple 256 x 256	***R***	14092.71	6.6409	-0.0007
	G	15923.85	6.1103	-4.40354E-05
	B	16261.35	6.0192	0.0002

4) Correlation analysis: Correlation analysis is a measure of how closely two variables are related. In plaintext images, adjacent pixels tend to be highly correlated, which can be utilized to attack the image. If the neighboring pixels correlation is excessively high, it makes it easier for attackers to predict the next pixel value. By breaking the correlation between pixels, statistical attacks can be prevented. As the correlation between pixels approaches zero, it becomes progressively more challenging for a potential attacker to deduce any insights into the original plaintext image. The correlation values mathematical formulas in three directions are computed by Eq. (36). In Table IV, the correlation coefficients between the original and cipher images are depicted for all three directions. As depicted in the table, the original image coefficients are near one, meaning a solid correlation between pixels.

$$\begin{cases} R_{xy} = \frac{cov\ (x,y)}{\sqrt{V(x)V(y)}} \\ cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \\ V(x) = \frac{1}{N}\sum_{i=1}^{N}(\ x_i - E(x))^2 \\ E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \end{cases} \tag{36}$$

Where the pixel total number is denoted by N, the values of the neighboring pair are x and y. The variance, mean, and covariance are V(x), E(x), and Cov(x,y), respectively.

The correlation values within the encrypted images approach zero, indicating a minimal correlation among pixels. Moreover, Table V shows that the correlation values for the proposed algorithm outperform the previous methods, which satisfies our objective of minimizing the correlation. In Fig. 4, the correlation distribution among adjacent pixels is depicted for the three directions of the 512 x 512 color Lena image. The figure shows the equal cipher image distribution, meaning the correlation between pixels is low. Moreover, scrambling over pixels on the block's level gives better results than scrambling pixels on the image's level, as introduced in Table VI.

TABLE III. COMPARISON OF MSE, PSNR, AND SSIM

Image	Ref.	MSE			PSNR			SSIM		
Grey Lena 512	***Ours***	9236.99			8.4755			-0.000208231		
	[26]	7797.7			9.2111			0.0350		
		R	***G***	***B***	***R***	***G***	***B***	***R***	***G***	***B***
Color Lena 512 x 152	***Ours***	10492.31	9218.75	7207.19	7.9221	8.4841	9.5531	0.0008	0.0005	0.0003
	[26]	10,637			7.8625			0.0331		
	[17]	8828.6			8.6719			0.0200		

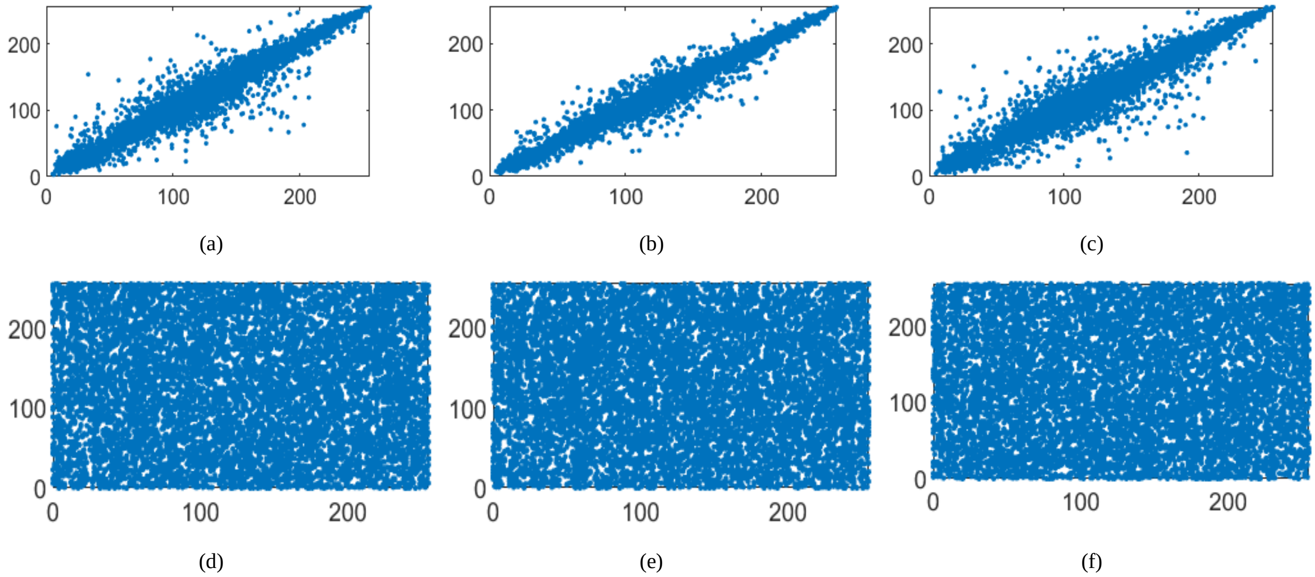
TABLE IV. CORRELATION COEFFICIENTS

Image		Plaintext Image			Encrypted Image		
		V	***H***	***D***	***V***	***H***	***D***
Male	***Grey***	0.9813	0.9774	0.9671	-0.0002	-0.0003	-0.0008
Barbara		0.9589	0.8954	0.8830	0.0021	-0.0002	-0.0004
Lake		0.9679	0.9545	0.9395	-0.0001	-0.00056	0.0012
Kitten		0.9228	0.9505	0.8840	-0.00095	-0.0031	-5.6E-05
Cameraman		0.9549	0.9196	0.8962	-0.0015	0.0033	-0.0078
Shreveport	***R***	0.8231	0.8295	0.7621	-0.00086	-1.34999E-05	0.00046

	G	0.8207	0.8245	0.7621	0.0006	-0.0002	0.0004
	B	0.8617	0.8646	0.8188	-0.00057	0.00086	0.0002
Baboon	***R***	0.8595	0.9105	0.8474	-0.0001	-0.0003	0.0001
	G	0.7755	0.8594	0.7434	0.0004	0.00075	0.0037
	B	0.8697	0.8953	0.8296	0.0007	-0.0006	-0.0013
Couple	***R***	0.9562	0.9493	0.9176	0.00099	0.0005	0.0084
	G	0.9534	0.9308	0.9002	-0.0058	-9.03962E-05	0.0023
	B	0.9442	0.9178	0.8880	-0.0004	-0.0053	-0.0001

TABLE V. COMPARISON OF CORRELATION COEFFICIENTS

		V			**H**			**D**		
Grey Lena 512 x 512	***Ours***	0.0001			-0.0004			0.0014		
	[26]	−0.0113			−0.0215			0.0089		
	[18]	0.0014			-0.0011			s0.0043		
		R	***G***	***B***	***R***	***G***	***B***	***R***	***G***	***B***
Color Lena 512 x 512	***Ours***	-0.0009	0.0006	0.00002	-0.00007	0.0007	0.0014	0.0002	0.0006	0.0029
	[26]	−0.0027			0.0007			−0.0104		
	[17]	0.0197			-0.0043			0.0032		
Color Lena 256	***Ours***	0.0004	0.0003	-0.0048	0.0004	0.0036	0.0033	0.0004	0.0012	-0.0006
	[11]	0.0098	0.0000	-0.0004	-0.0044	-0.0013	-0.0061	-0.0013	0.0042	-0.0093
	[28]	0.0019			0.0020			-0.0025		
	[27]	0.003	-0.004	-0.0008	0.0003	0.001	-0.0009	0.0008	0.002	0.002
	[29]	0.0063	-0.0023	0.0087	-0.0015	0.0035	0.0053	0.0043	-0.0081	0.0011
	[19]	−0.0002	−0.0051	0.0016	0.0019	0.0024	0.0007	0.0008	0.0018	−0.0019



(a) (b) (c)

(d) (e) (f)

Fig. 4. Distribution of correlation values between original and cipher images in different directions for 512 x 512 color Lena image (a) original image's horizontal correlation (b) original image's vertical correlation (c) original image's diagonal correlation (d) cipher image's horizontal correlation (e) cipher image's vertical correlation (f) cipher image's diagonal correlation.

TABLE VI. COMPARISON BETWEEN THE SCRAMBLING ON THE IMAGE'S LEVEL AND THE BLOCK'S LEVEL

		V			H			D		
		R	***G***	***B***	***R***	***G***	***B***	***R***	***G***	***B***
Grey Lena 512 x 512	***Block***	0.0001			-0.0004			0.0014		
	Pixel	0.0014			0.0015			-0.0024		
Color Lena 512 x 512	***Block***	-0.0009	0.0006	0.00002	-0.00007	0.0007	0.0014	0.0002	0.00055	0.0029
	Pixel	0.0015	-0.0024	-0.0009	-0.0009	-0.0009	-0.0013	0.00198	0.0022	-0.0006
Color Lena 256	***Block***	0.0004	0.0003	-0.0048	0.0004	0.0036	0.0033	0.00036	0.0012	-0.0006
	Pixel	-0.008	0.0051	-0.0063	-0.0100	0.0021	-0.00085	-0.0002	-0.0038	-0.0024

5) Entropy: Entropy is a metric that indicates an image's level of randomness and unpredictability. The entropy can be calculated as given in Eq. (37).

$$Entropy = -\sum_{i=0}^{2^n-1} P(x_i)Log_2(P(x_i)) \qquad (37)$$

Where the probability of x is P(x).

An entropy value of approximately eight is considered the ideal value for a cipher image [30]. Table VII presents the entropy value of the suggested algorithm, while Table VIII compares it with the recent work. The comparison reveals that the proposed method gives a better or similar result than the recent work, which satisfies our goal of maximizing the randomness and unpredictability of the image.

TABLE VII. THE ENTROPY, NPCR, AND UACI VALUES FOR THE SUGGESTED ALGORITHM

Image	Color	Entropy	NPCR	UACI
Male	***Grey***	7.99985	99.60	33.58
Barbara	***Grey***	7.99936	99.60	33.52
Lake	***Grey***	7.99936	99.60	33.56
Kitten	***Grey***	7.99752	99.62	33.48
Cameraman	***Grey***	7.99785	99.60	33.48
Shreveport	***R***	7.99983	99.61	33.48
	G	7.99984	99.61	33.52
	B	7.99983	99.60	33.48
Baboon	***R***	7.99926	99.62	33.60
	G	7.99922	99.61	33.62
	B	7.99933	99.60	33.63
Couple	***R***	7.99715	99.62	33.57
	G	7.99744	99.58	33.56
	B	7.99741	99.60	33.59

B. The High-Performance Evaluation

In a differential attack, an attacker aims to uncover the differences between encrypted images produced from two slightly varied versions of the original image. The attacker looks for non-random areas in the encrypted images and then looks for changes in these areas that would conclude the key used in image encryption. In slight alterations to the original image, the encryption procedure produces two cipher images: one for the original image and a second for the modified version. If a single-bit change in the original image causes the encrypted images to differ by at least 50%, then the differential attack cannot decrypt the cipher images. Two performance metrics are utilized, UACI and NPCR, to estimate the algorithm's resistance to differential attacks. The mathematical expressions for these two metrics are in Eq. (38) and (39).

$$NPCR = \frac{\sum_{x,y} D(x,y)}{N\ x\ M}\ x\ 100\% \qquad (38)$$

$$UACI = \frac{\sum_{x,y} |E_1(x,y) - E_2(x,y)|}{255\ x\ M\ xN} \qquad (39)$$

Where $E_1(x,y)$ and $E_2(x,y)$ are two encrypted images of the same original image, but only one pixel value is changed. D(x,y) is calculated as follows:

$$D(x,y) = \begin{cases} 0 & E_1(x,y) = E_2(x,y) \\ 1 & E_1(x,y) \neq E_2(x,y) \end{cases}$$

Table VII showcases the NPCR and UACI values obtained through the proposed algorithm, whereas Table VIII provides a comparative evaluation of its performance against state-of-the-art methods. The NPCR and UACI values, approaching 99.6094% and 33.4635%, respectively, are considered close to the ideal benchmarks. The two metric values achieved by the suggested algorithm are in close proximity to the ideal values, demonstrating the suggested method's robustness against differential attacks.

C. The Processing Time

Time evaluation is necessary in assessing the efficacy of cryptographic systems, with efficient systems expected to exhibit minimal encryption processes. To assess the time efficiency of our suggested algorithm, we conducted time measurements for each encryption step of the Kitten image, as shown in Fig. 5. The most consumable time is the key generation stage, as introduced in Fig. 5. Table IX displays a comparison of the encryption time consumed by the suggested approach with that of other approaches. The results show that our suggested method gives better time encryption for Lena with size 512. However, in the case of Lena's image with size 256, the results are higher than the two works as these works did not treat each color channel as a separate matrix as in our case.

TABLE VIII. THE ENTROPY, NPCR, AND UACI COMPARISON WITH PREVIOUS WORK

		Entropy			NPCR			UACI		
Grey Lena 512 x 512	***Ours***	7.9992			99.61			33.67		
	[26]	7.9993			99.61			33.701		
	[18]	7.9994			99.61			33.47		
		R	***G***	***B***	***R***	***G***	***B***	***R***	***G***	***B***
Color Lena 512 x 512	***Ours***	7.9993	7.9992	7.9994	99.60	99.59	99.60	33.63	33.52	33.53
	[13]	7.9924			99.61			33.78		
	[26]	7.9994			99.63			33.03		
	[17]	-------			99.62			30.45		
Color Lena 256	***Ours***	7.9971	7.9973	7.9976	99.64	99.63	99.66	33.69	33.49	33.49
	[11]	7.9968	7.9973	7.9974	99.60	99.61	99.61	33.53	33.38	33.67
	[10]	7.9974	7.9975	7.9973	99.63	99.62	99.62	33.51	33.32	33.46
	[27]	7.9892	7.9902	7.9896	99.61			32.95		
	[29]	7.9973	7.9973	7.9973	99.61	99.60	99.60	33.48	33.46	33.36
	[19]	7.9956	7.9954	7.9962	100	100	100	33.45	33.43	33.56

TABLE IX. COMPARISON OF ENCRYPTION TIME

	Our	**[27]**	**[29]**	**[19]**	**[11]**	**[26]**	**[17]**
Lena 256	2.318	4.455	0.375	0.282			
Lena 512	6.968	14.966	-------	------	12.117	2.0179	80.05

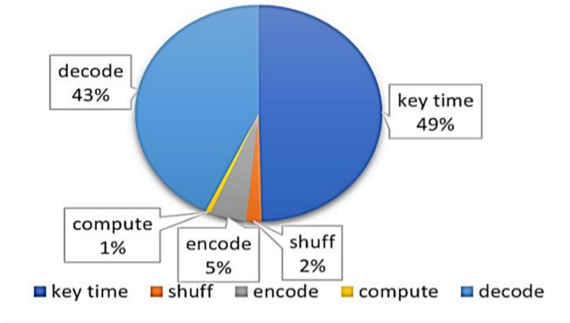


Fig. 5. The execution time of each stage in the encryption approach for the Kitten image.

D. The Strength of the Cryptosystem Evaluation

1) Key space analysis: There are different types of attacks to obtain the original image. One type of attack that hackers can use is Brute force, where they try to decrypt the cipher image by employing all possible keys until the correct one is found. A key space with a large size can be a good defense against brute-force attacks, which refers to the entire set of keys used for image encryption. The researchers have determined that a minimum key space of 2100 [31] is required to resist brute-force attacks. The image cryptosystem's secret key was generated using 24 parameters from different chaotic maps in the suggested method—additionally, the hash function used 256 bits.

Following the IEEE 754 floating-point standard (double), the substantial precision amounts to 53 bits, necessitating 15 decimal digits for representation. Consequently, the key space of the proposed system stands at 10^{437}, demonstrating resilience against brute-force attacks by surpassing the threshold of 2^{100}.

2) Key sensitivity: A cryptosystem is considered effective if it is extremely sensitive to keys. Key sensitivity means any minor change in key value results in a significant change in output. There are two ways to test key sensitivity. The first is by making a faint change in the key value; the result should be two different encrypted images. The other way is that the slight change in key-value results in not retrieving the original image correctly from the encrypted one. In this paper, we test the key sensitivity in two ways. In the first test, we changed the key slightly and then encrypted the plain image to get another cipher image using two different keys. We change the initial value of x and y in the hyperchaotic used to generate the final key by adding to each value 10^{-14}. The results illustrated in Fig. 6 show the robustness of the suggested algorithm regarding the slight change in key as parts c and e show the difference between the cipher image generated from the valid key and the cipher image produced from the modified keys.

The alternative approach for conducting the key sensitivity test encompasses encrypting the original image with the accurate key and decrypting the resulting cipher image using an altered key. Fig. 7 illustrates the distinction between the decrypted image derived from the correct key and the decrypted image obtained from the two adjusted keys. The difference between the two images proves the suggested algorithm's high sensitivity against a minor key change. Based on the two test results, the suggested method can resist brute-force and statistical attacks.

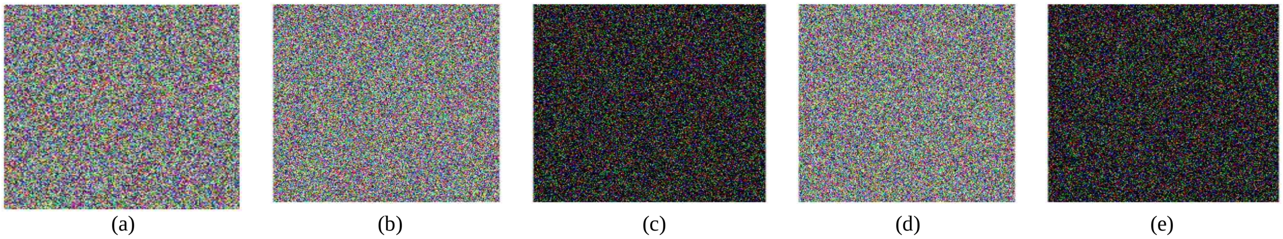


Fig. 6. Testing key sensitivity in encryption stage (a) image ciphered by the original key (b) image ciphered by the first modified key 1(x0+10^{-14}) (c) difference between a and b (d) image ciphered by the second modified key 2(y0+10^{-14}) (e) difference between c and d.

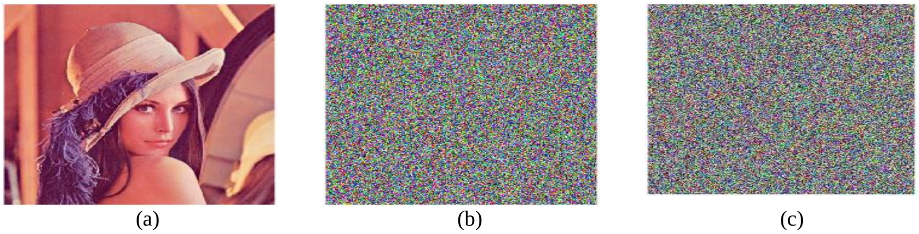


Fig. 7. Testing key sensitivity test in decryption (a) original image (b) decrypted image utilizing the first modified key (x0+10^{-14}) (c) decrypted image utilizing the second modified key (y0+10^{-14}).

3) Classical attacks: Classical attacks include four attack types: chosen-plaintext, known-plaintext, chosen-ciphertext and ciphertext-only attack. The most harmful attacks are chosen and known plaintext attacks. In these two attacks, the attacker has access to plaintext and encrypted images and tries to deduce the keys. If the cryptographic system can withstand these attacks, it would be immune to the other two types.

The cryptographic system would be immune to the known and chosen plaintext attacks if it is sensitive to the key change, which is proved in subsection 2 of section D. Moreover, the suggested system is a one-time pad system that utilizes the SHA-256 function to produce the key. Another test in measuring the immunity to the attacks of known and chosen plaintext is to use all black and all-white images as the original, as shown in Fig. 8(a) and (d). The corresponding cipher images for these two images and their histogram are displayed in Fig. 8 (b-c) and (e-f). The entropy and correlation values of the black and white images are depicted in Table X. From the results, the suggested algorithm shows strong immunity to the attacks of chosen and known plaintext attacks.

4) Image processing attacks resistance: During transmission, the cipher image may be exposed to several disruptions, and some attacks on cipher images will result in data loss. The decryption of a corrupted cipher image may thus lead to distorted or even unnoticeable results. The most famous image processing attacks are data and noise loss attacks. Minimizing the impact of data and noise loss attacks on the restored image is crucial for achieving an efficient image encryption algorithm.

The effectiveness of the proposed algorithm in withstanding noise attacks was assessed by introducing various levels of salt and pepper noise (0.5, 0.05, and 0.005) and then calculating the PSNR values between the original images and their decrypted counterparts. The results, presented in Table XI, affirm that the proposed algorithm effectively resists salt and pepper attacks.

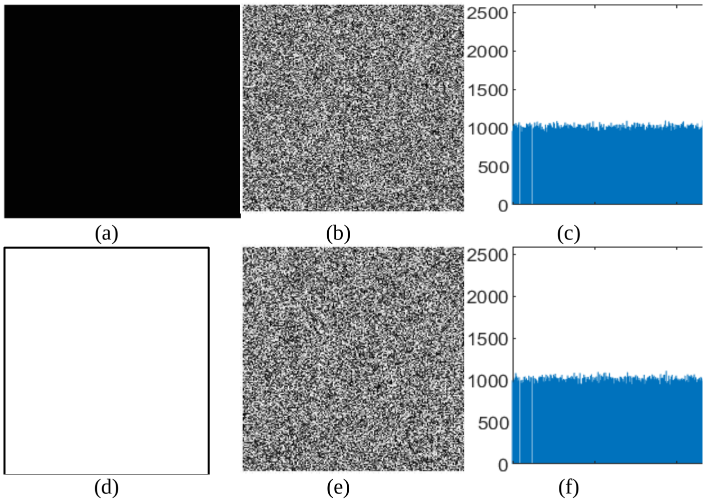


Fig. 8. Classical attack test(a-c) black image, its cipher, and its cipher histogram (d-f) white image, its cipher, and its cipher histogram.

TABLE X. THE ENTROPY AND CORRELATION OF THE WHITE AND BLACK IMAGES

image		Entropy	Correlation		
			V	***H***	***D***
White	***plain***	-0.000000	-	-	-
	cipher	7.9993	-0.00005	-0.0034	-0.00199
Black	***plain***	0.0012	-	-	-
	cipher	7.99938	0.00088	0.0011	-0.0033

Furthermore, the algorithm's ability to withstand occlusion attacks was assessed by applying varying masks to the cipher images (1/16, 1/8, and 1/4). The decrypted images resulting from this masking process are displayed in Table XI, showcasing the algorithm's robustness against cropping attacks. Despite occlusion attempts, the algorithm demonstrates its capability to recover a portion of the image's information.

TABLE XI. THE PSNR VALUE OF THE PLAINTEXT IMAGES AND THEIR DECRYPTED IMAGES AFTER APPLYING NOISE AND DATA LOSS ATTACKS

Attack	**Parameters**	**PSNR**		
		R	***G***	***B***
Salt and pepper	***0.005***	27.3081	27.8380	27.8922
	0.05	20.3436	20.8994	21.9327
	0.5	10.8365	11.5345	12.5723
Cropping	***1/16***	19.1520	20.9261	21.9408
	1/8	16.3463	17.9376	18.8688
	1/4	13.4978	14.8659	15.8419

V. CONCLUSION

The suggested technique in this paper employs DNA, RSA, and chaotic maps to produce an extremely robust and secure image encryption method. The approach encompasses three phases: key generation, confusion, and diffusion. The hash function and hyperchaotic are used to generate a robust key as the hash function is a one-time pad and chaotic produces unpredictable and random numbers. The suggested algorithm uses the original image and a user-defined key to generate the encryption key, thereby preventing the chosen/known-plaintext attack. In the confusion phase, there are two options to change the pixel's locations, either changing the location on the image's level or the block's level based on the Duffing map. After that, each pixel is subjected to two consecutive confusion steps: Henon and Arnold map. In the diffusion phase, the confusion phase output undergoes two successive diffusion steps: DNA followed by RSA cryptography. Using two steps in each phase maximizes the security and unpredictability of the suggested approach. Moreover, the suggested approach can withstand different attacks. Various security tests demonstrate the approach's effectiveness in withstanding attacks and achieving low correlation between neighboring pixels.

In future research, we will explore multi-model image encryption by combining color, texture, and depth data. Additionally, we aim to integrate machine learning techniques to optimize encryption parameters and enhance security. Another priority is reducing encryption time.

Use of AI tools declaration: The authors have not used Artificial Intelligence (AI) tools in creating this article.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

REFERENCES

[1] J. Wang and W. Jiang and H. Xu and X. Wu and J. Kim, "Image encryption based on Logistic-Sine self-embedding chaotic sequence", Optik, vol. 271, p. 170075, 2022.

[2] N. Rani, V. Mishra and S.R. Sharma, "Image encryption model based on novel magic square with differential encoding and chaotic map", Nonlinear Dynamics, vol. 110, No. 3, 2022.

[3] C. Tu, R. Cui, and K. Liu, "Design of Clothing with Encrypted Information of Lost Children Information Based on Chaotic System and DNA Theory", Autex Research Journal, vol.0, no.0, 2022,

[4] X. Zhang and Y. Hu, "Multiple-image encryption algorithm based on the 3D scrambling model and dynamic DNA coding", Optics & Laser Technology, vol. 141, p. 107073, 2021.

[5] H. Hu, Y. Cao, J. Xu, C. Ma and H. Yan, "An Image Compression and Encryption Algorithm Based on the Fractional-Order Simplest Chaotic Circuit," in IEEE Access, vol. 9, pp. 22141-22155, 2021.

[6] X. Wang, Y.Su, L. Liu, H. Zhang, and S. Di, "Color image encryption algorithm based on Fisher-Yates scrambling and DNA subsequence operation", Visual Computer, vol. 47, no. 10, 2021.

[7] Y. Xiao and Z.R. Lin and Q. Xu and J. Du and L.H. Gong, "Image encryption algorithm based on semi-tensor product theory", Journal of Modern Optics, vol. 69, no. 19, pp. 1063-1078, 2022.

[8] X. Wang and R. Si, "A new chaotic image encryption scheme based on dynamic L-shaped scrambling and combined map diffusion", Optik, vol. 245, p. 167658, 2021.

[9] N. Ying, Z.Xuncai. "An Image Encryption Algorithm Based on Filling Curve and Adjacent Pixel Bit Scrambling", Journal of Electronics & Information Technology, vol. 44, no. 3, pp. 1137-1146, 2022.

[10] S. Wang, Q. Peng, and B. Du, "Chaotic color image encryption based on 4D chaotic maps and DNA sequence", Optics and Laser Technology, vol. 148, p. 107753, 2022.

[11] J. Yu, W. Xie, Z. Zhong, and H. Wang, " Image encryption algorithm based on hyperchaotic system and a new DNA sequence operation", Chaos, Solitons and Fractals, vol. 162, p. 112456, 2022.

[12] C. Zou, X. Wang, C. Zhou, S. Xu and C. Huang, " A novel image encryption algorithm based on DNA strand exchange and diffusion", Applied Mathematics and Computation, vol. 430, p. 127291, 2022.

[13] B. Jasra and A. Moon, "Color image encryption and authentication using dynamic DNA encoding and hyperchaotic system", Expert Systems with Applications, vol. 206, p. 117861, 2022.

[14] X. Li, J. Zeng, Q. Ding, and C. Fan, "A Novel Color Image Encryption Algorithm Based on 5-D Hyperchaotic System and DNA Sequence", Entropy, vol.24, no. 9, p. 1270, 2022.

[15] N. Rani, S. Rani Sharma, and V. Mishra, "Grayscale and colored image encryption model using a novel fused magic cube", Nonlinear Dynamics, vol. 108, pp. 1773-1796, 2022.

[16] J. Zheng and Q. Zeng, " An image encryption algorithm using a dynamic S-box and chaotic maps", Applied Intelligence, vol. 52, pp. 15703–15717, 2022.

[17] N. Parekh, L. D'Mello, "CHaDRaL: RGB image Encryption based on 3D Chaotic Map, DNA, RSA and LSB", 2021 International Conference on Artificial intelligence and machine vision (AIMV), 2021.

[18] M. Liu and G. Ye, "A new DNA coding and hyperchaotic system based asymmetric image encryption algorithm", Mathematical Biosciences and Engineering, vol. 18, pp 3887-3906, 2021.

[19] U.H. Mir, D. Singh, and P.N. Lone, "Color image encryption using RSA cryptosystem with a chaotic map in Hartley domain", InformationSecurity Journal: A Global Perspective, vol. 31, issue 1, pp. 49-61, 2022.

[20] K. Jiao, G. Ye, Y. Dong, X. Huang, and J. He, "Image encryption scheme Based on a Generalized Arnold map and RSA Algorithm", Security and Communication Networks, vol. 2020, pp. 1-14, 2020.

[21] Babu M, G. Shamala Devi, M. Yamini Krishna, M. Viswa Prasanna, N. Iswarya, "Image Encryption Using Chaotic Maps and DNA Encoding", Journal of Xidian University, vol. 14, issue 4, pp 1817-1827, 2020.

[22] M. Kumar, A.Saxena, S.S.Vuppala, "A survey on chaos-based image encryption techniques" " Multimedia security using chaotic Maps: principles and Methodologies, vol. 884, pp. 1-26, 2020.

[23] K.C.Jithin, Syam Sankar, "Colour image encryption algorithm combining Arnold map, DNA sequence operation, and a Mandelbrot set", Journal of Information Security and Applications, vol. 50, pp 1-22, 2020

[24] A. Akhshani, A. Akhavan, A.Mobaraki, S.C. Lim, and Z. Hassan, "Pseudo random number generator based on quantum chaotic map", Commun Nonlinear Sci NumerSimulat 19, vol. 19, issue 1, pp 101-111, 2014.

[25] S.A. Elsaid, E.R. Alotaibi, and S. Alsaleh, "A robust hybrid cryptosystem based on DNA and Hyperchaotic for images encryption", Multimedia Tools and Applications, vol. 82, pp 1995-2019, 2022.

[26] S.F. Yousif, A.J. Abboud, R. S. Alhumaima, "A new image encryption based on bit replacing, chaos and DNA coding techniques", Multimedia Tools and Applications, vol. 81, pp 27453-27493, 2022.

[27] S. Mansoor, P. Sarosh, S.A. Parah, Habib Ullah, Mohammad Hijji, and Khan Mouhammad, "Adaptive Color Image Encryption Scheme Based on Multiple Distinct Chaotic Maps and DNA Computing", Mathematics, vol 10, 2022.

[28] R.W. Ibrahim, H. Natiq, A.AlKhayyat, A.K. Farhan, N. MG. AlSaidi, D.Baleanu, "Image Encryption Algorithm Based on New Fractional Beta ChaoticMaps", Computer Modeling in Engineering and Sciences, vol. 131, pp.119-131, 2022.

[29] X. Liu, X. Tong, Z.Wang, M. Zhang, "A novel hyperchaotic encryption algorithm for color image utilizing DNA dynamic encoding and self-adapting permutation", Multimedia Tools and Applications, vol. 81, p. 21779, 2022

[30] M.B. Farah, A. Farah, T. Farah, "An image encryption scheme based on a new hybrid chaotic map and optimized substitution box", Nonlinear Dynamic, vol. 99, pp 3041–3064, 2020.

[31] N. Iqbal, R. Naqvi, M. Atif, M.A. Khan, M. Hanif, S. Abbas, and D. Hussain, "On the Image Encryption Algorithm Based on the Chaotic System, DNA Encoding, and Castle," in IEEE Access, vol. 9, p. 118253, 2021.