IRAQI Academic Scientific Journals

ISSN: 1813-162X (Print); 2312-7589 (Online)

Tikrit Journal of Engineering Sciences

available online at: http://www.tj-es.com

TJES Tikrit Journal of Engineering Sciences

An Effective Composite Cryptosystem for Securing Digital Images Using Genetic-RSA-DNA Algorithms

Sura F. Yousif *[a], Ali J. Abboud [b]

[a] Chemical Department, Engineering College, Diyala University, Diyala, Iraq.
[b] Department of Computer Engineering, College of Engineering, Diyala University, Diyala, Iraq.

Keywords:
Image Security; Composite Cryptosystem; Genetic; RSA; DNA.

Highlights:

- Well known security algorithms are integrated to make strong image security algorithm.
- DNA cryptography, genetic algorithm, and RSA algorithm are the main ingredients.
- The strength of the proposed approach defeated several known attacks.

ARTICLE INFO
Article history:

Received	27 Aug. 2023
Received in revised form	26 Jan. 2024
Accepted	15 Oct. 2024
Final Proofreading	20 Aug. 2025
Available online	28 Aug. 2025

Citation: Yousif SF, Abboud AJ. **An Effective Composite Cryptosystem for Securing Digital Images Using Genetic-RSA-DNA Algorithms**. *Tikrit Journal of Engineering Sciences* 2025; **32**(3): 1593.
http://doi.org/10.25130/tjes.32.3.31

Corresponding author:

Sura F. Yousif

Chemical Department, Engineering College, Diyala University, Diyala, Iraq.

Abstract: Image communication in unsafe networks like the Internet is becoming widespread for its important applications in military, medicine, and entertainment. Such insecure channels impose real challenges in protecting the security and privacy of transmitted multimedia contents such as images. Many approaches have been proposed in the literature to overcome these challenges; however, each one of these methods has its limitations. The best way to reap the benefits of these approaches is to integrate them into a single approach to remove some or all of their limitations. Hence, in this research, a new integrated approach is proposed to secure images transmitted through insecure communication channels. It is composed of integrating a genetic algorithm, DNA cryptography, and the RSA algorithm in the form of successive security layers. First, the original image and the generated 256-bit hash values are encoded via DNA complementary rules to obtain the coded DNA sequences. Second, the resultant two encoded chains are summed by performing a DNA XOR process to get the scrambled DNA image. Third, the scrambled image is decoded to gain the ciphered image. Fourth, the genetic technique is applied to the cipher image from the prior level to get the ciphered image from the second encryption layer. Finally, the RSA algorithm is employed on the cipher image from the last step to earn the final encrypted image. The experimental results and statistical tests prove the viability and ability of the proposed approach to secure images against several known attacks and overcome the limitations of single-algorithm security approaches. To sum up, this approach can be utilized for secure image transmission in real-time telecommunication systems.

نظام تشفير مركب فعال لتأمين الصور الرقمية باستخدام خوارزميات Genetic-RSA-DNA

سرى فهمي يوسف[١]، علي جاسم عبود[٢]

[١] قسم الهندسة الكيمياوية/ كلية هندسة / جامعة ديالى / بعقوبة - العراق.

[٢] قسم هندسة الحاسوب/ كلية الهندسة / جامعة ديالى / بعقوبة - العراق.

الخلاصة

أصبح التواصل بالصور في الشبكات الغير امنة مثل الإنترنت منتشرًا على نطاق واسع لتطبيقاته المهمة في المجالات العسكرية والطبية والترفيهية. تفرض مثل هذه القنوات غير الآمنة تحديات حقيقية في حماية أمن وخصوصية محتويات الوسائط المتعددة المنقولة مثل الصور. وقد تم اقتراح العديد من الأساليب في الأدبيات للتغلب على هذه التحديات ولكن كل واحدة من هذه الأساليب لها محدوديتها. أفضل طريقة لجني فوائد هذه الأساليب هي دمجها في نهج واحد لإزالة بعض أو كل القيود المفروضة عليها. ومن هنا تم في هذا البحث اقتراح نهج متكامل جديد لتأمين الصور المنقولة في قنوات الاتصال غير الآمنة. وهو يتألف من دمج الخوارزمية الجينية وتشفير الحمض النووي وخوارزمية RSA في شكل طبقات أمان متتالية. أولاً، يتم تشفير الصورة الأصلية وقيم التجزئة ٢٥٦ بت التي تم إنشاؤها عبر قواعد الحمض النووي التكميلية للحصول على تسلسل الحمض النووي المشفر. ثانياً، يتم جمع السلسلتين المشفرتين الناتجتين عن طريق إجراء عملية DNA XOR للحصول على صورة DNA المشوشة. ثالثاً، يتم فك تشفير الصورة المشوشة للحصول على الصورة المشفرة. رابعاً: يتم تطبيق التقنية الوراثية على الصورة المشفرة من المستوى السابق للحصول على الصورة المشفرة من طبقة التشفير الثانية. وأخيرًا، يتم استخدام خوارزمية RSA على الصورة المشفرة من الخطوة الأخيرة للحصول على الصورة المشفرة النهائية. اثبتت النتائج التجريبية والاختبارات الإحصائية جدوى النهج المقترح وقدرته على تأمين الصور ضد العديد من الهجمات المعروفة والتغلب على القيود المفروضة على أساليب أمان الخوارزمية الفردية. خلاصة القول، يمكن استخدام هذا النهج لنقل الصور بشكل آمن في أنظمة الاتصالات في الوقت الحقيقي.

الكلمات الدالة: أمان الصورة، نظام التشفير المركب، الخوارزمية الجينية، RSA، DNA.

1.INTRODUCTION

With the fast advances in network techniques, a revolution has occurred in digital communications. The transmission of text, audio, video, and digital images over the Internet has become more popular in this era. Many threats face the transmission of these multimedia signals in shared and open internet networks. Thus, people have many concerns about the privacy, confidentiality, and security of their data. The cryptography algorithms are the most common and efficient techniques to secure the digital signals' contents [1-4]. These algorithms transform the digital signals into an unrecognizable form at the sender's side using certain encryption processes and a confidential key. The ciphered information is then deciphered at the receiver's end using the same processes and the same secret key. Depending on the cryptographic key used, the cryptographic techniques are categorized into two techniques. The identical key techniques are among the most commonly used tools to attain high data security and privacy. The sender and recipient in this kind of technique utilize the same key for enciphering and deciphering the secret message. In the dissimilar keys techniques, a double key is employed, one is the public key used by the sender of the message for encryption, and another is the private key used by the receiver to decrypt the received message. One-way functions are used to implement the asymmetric cryptographic algorithms. Mathematically, the computation of these functions is easy in one direction; however, it is very difficult in the reverse one. Factorizing large prime numbers is an example of a one-way function since multiplying two prime numbers gives their product, which is very easy; however, revealing the factors is very difficult because there are many possibilities. This problem is one of the great difficulties in mathematics, e.g., RSA and Diffie-Hellman algorithms [5-8]. DNA enciphering is a successful and efficient image ciphering scheme. Adleman was the inventor of DNA computing in 1994. Due to the excellent properties of the DNA technique, like low energy use, large storage density, and enormous parallelism, it has been utilized in the field of cryptography. The basic conception of DNA image encryption is classified into two stages: First, it is utilized to encode pixels of the input image into a DNA chain, and then the original image is produced by employing those rules. Second, by relying on the DNA operation rules, the opener image is produced by the pixels of the coded input image to construct the ciphered image. Thus, the image encryption approaches based on the DNA technology have exclusive advantages over conventional cryptographic approaches [9-13]. Several image encryption techniques have recently been introduced to enhance the security of the cryptosystem. For example, studies in references [9-19] presented an image encryption method by merging DNA coding and chaos, whereas the methods in references [5, 6] presented an image cryptosystem based upon a genetic algorithm. The input image is encrypted/decrypted in Ref. [7] by adopting RSA, Genetic, and AES techniques. Also, the authors in Ref. [8] suggested a new mechanism to encrypt the digital image via AES and RSA algorithms. An encryption model was developed in Ref. [20], which relies on chaotic maps and AES technology. Moreover, symmetric cryptography and chaotic systems were exploited in Ref. [21] for digital image encryption/decryption. The plain image was encoded in Ref. [22] by utilizing a public key cryptosystem and chaotic maps. Besides, an image encryption scheme is described in Ref. [23], depending on RSA and chaotic systems. Finally, a modified approach was introduced in Ref. [24] by merging genetic algorithm and chaos theory. However, a single-dimensional chaos organization is easily implemented and has a simple structure; however, it possesses few flaws, such as fragile defense, few parameters, and limited key capacity. On the other hand, high-dimensional or hyperchaotic systems possess a complex structure and more parameters; however, they are unsuitable in real-time applications because they increase the computational complexity and implementation costs and decrease the encryption speed. Furthermore, several chaos-based image encryption schemes lack sufficient security because they can be easily cracked by some cryptographic attacks, such as differential, known/plaintext, and chosen/plaintext attacks [3, 9, 15, 16]. In addition, some image cryptosystems are sorts of symmetric or asymmetric algorithms. Symmetric approaches, in which the ciphering and deciphering keys are identical, have some defects in key management and security. Also, they are vulnerable to various known attacks. Meanwhile, asymmetric mechanisms, in which the encryption and decryption keys are different, suffer from some problems related to key length, authentication, encryption speed, and the difficulty of message decryption if the private key is lost [23]. It can be concluded that chaotic systems are not always the suitable choice for image enciphering, and image cryptosystems using either the DNA-only algorithm or the symmetric algorithm. The asymmetric algorithm possesses some flaws in defense, efficiency, and rapidity. Therefore, a combination of various cryptographic security techniques should be used to build a secure and robust cryptosystem. Besides, a method should be developed to encrypt both grayscale and color images to meet the real-time application requirements. To overcome the above shortcomings and security flaws of classical cryptographic techniques, this study presents a secure, efficient, robust, and fast cryptosystem for grayscale and color images that relies upon the principle of DNA sequence operation integrated with the most popular cryptographic techniques: genetic and RSA technologies. Genetic technique is a type of symmetric algorithm whose security relies upon three optimization operations: selection, crossover, and mutation [24]. On the other hand, the RSA technique is a kind of asymmetric algorithm whose security relies upon the difficulty of factoring large integers [23]. The strength of these two mechanisms is merged with the outstanding features of DNA complementary rules/operations to construct a powerful encryption scheme. A single image in this work is encrypted with three various security technologies. The encryption methods are used in a cascade manner to secure the input images. After the implementation of the cryptographic approach on the plain image, the final image is highly safe and approximately the same as the noisy image. The decryption of the encrypted images is very hard because the attacker may break one encryption system; however, it has no chance of breaking three strong encryption systems. DNA coding rules/operations are first executed to remove the relationships between pixels and shuffle the pixels' locations/values. In contrast, genetic and RSA techniques are employed to accomplish the encryption phase and improve the security. The main goal of this research is to propose an image cryptosystem that provides enhanced security for the transmitted image over an open network by performing triple encryption using three cryptographic techniques. The proposed article has the following contributions: (1) Design a digital image cryptographic scheme for encrypting and decrypting gray and color images, which combines the DNA, genetic, and RSA methods to enhance the image encryption performance, (2) design a secure and robust cryptosystem by combining symmetric and asymmetric encryption approaches, (3) design an efficient and fast system easy to implement in software and hardware environments, and (4) giving security analysis for the proposed approach and evaluating its robustness and effectiveness in practical situations.

2.RELATED THEORY

2.1.DNA Rules and Algebraic Operations

DNA series is built using 4 nucleic acid bases: Adenine (A), Thymine (T), Cytosine (C), and Guanine (G). Where "A, T" and "C, G" represent the base pairs or complementary pairs in DNA computing. Generally, there are only eight types of encoding combinations that are suitable for the complementarity rule between the four bases. The rules are illustrated in Table 1. The gray amount of pixels in the image is expressed as eight bits, which corresponds to its binary sequence. This chain is coded using a DNA series of 4 bits. As an example, suppose a pixel's grayscale score is 225, then the corresponding binary sequence to this pixel is 11100001. This binary sequence is coded to a DNA series: TGAC using Rule 1, or TCAG using Rule 2, and so on. The resultant DNA sequence is decoded inversely to obtain the original pixel value [9-11]. Rule 1 is used in the present work to encode and decode the input image. Also, the researchers have presented some algebraic and biological operations that depend on the DNA sequence, like subtraction (-), addition (+), XNOR ($\odot$), and XOR ($\oplus$) processes. These processes are performed according to conventional operations in the binary systems [3, 12]. Table 2 shows the details of XOR and XNOR operations, and the results of these two operations in each column and row are distinctive. Hence, the XOR operation is applied in this scheme for digital image encryption.

Table 1 The (8) Rules for DNA Series [12].

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
G	01	10	00	11	00	11	01	10
C	10	01	11	00	11	00	10	01

Table 2 DNA Series Logical Processes (XOR, XNOR) [3].

⊕	A	C	T	G	⊙	A	C	T	G
A	A	C	T	G	A	C	A	G	T
T	T	G	A	C	T	T	G	C	A
C	C	A	G	T	C	A	C	T	G
G	G	T	C	A	G	T	G	A	C

2.2.Genetic Algorithm

The Genetic Algorithm, or GA, was first introduced by John Holland in 1975. GA is a search technology used to solve optimization problems, such as natural selection, inheritance, crossover, and mutation. This technique consists of fundamental (3) operators: mutation, selection, and crossover. The selection operation is a quantitative criterion in which a set of individuals called chromosomes is selected from the population. Any chromosome is assessed by utilizing a fitness formula to provide suitable values that are going to be normalized. Crossover operation is performed by choosing two chromosomes, and then a new chromosome is produced by taking the properties from the first and second chromosomes. Mutation includes string-based alterations to the elements that are similar to biological mutations. The mutation process is performed by bit inversion in the binary string, i.e., bit 1 is flipped or mutated to bit 0 and vice versa [5-7].

2.3.RSA Algorithm

RSA is a famous cryptographic algorithm that is used in internet communications, such as Microsoft Explorer, Netscape Navigator, and Chrome. RSA is based on asymmetric key cryptography. Two keys are used by this algorithm to secure all transactions. The basic principle of RSA is based on mathematical processes called one-way functions. These functions are very easy to solve in one direction; however, they are very difficult to solve in the reverse manner. Three main phases are involved in the RSA algorithm: key generation, encryption, and decryption.

2.3.1.Key Generation

- Two distinct prime random numbers are selected: b and c.
- m (the modulus) is calculated as $(m = b \times c)$.
- α (the Euler's totient function) is computed as $\alpha = (b - 1) \times (c - 1)$.
- A third integer number e (the public exponent) is selected, such that $\gcd(e, \alpha) = 1$.
- d (the private exponent) is calculated as $(d = e^{-1}(\mathrm{mod}\ \alpha))$.

Thus, (e, m) represents the public key (encryption key), while (d, m) represents the secret key (decryption key). The α, b, and *c* values should be kept confidential because they can be used for calculating the decryption key.

2.3.2.Encryption/Decryption

Let M be the plaintext required to be transmitted; the ciphertext C is computed from Eq. (1).

$$\mathbf{C = M^{e} mod\ m} \quad (1)$$

The original message M is retrieved from the cipher message C using Eq. (2) [7, 8, 23]:

$$\mathbf{M = C^{d} mod\ m} \quad (2)$$

3.PROPOSED APPROACH

The proposed image encryption approach relies upon the incorporation of DNA, genetic, and RSA technologies. The encryption process starts by encoding the input image and the hash bits' values generated via the hash function into two DNA sequences by applying the DNA coding rules. Second, the XOR process in the DNA algorithm is executed to sum up the two DNA sequences to obtain a scrambled DNA image. Then, the resulting DNA image is decoded using the DNA decoding rules to acquire the cipher image from the first security layer. Third, the output image from the second part is encrypted again using the genetic mechanism to get the secure image from the second enciphering layer. Lastly, to complete the encryption operation, the resultant image from the earlier layer is enciphered using the RSA algorithm to get the ultimate ciphered image. The framework of the presented image security model is displayed in Fig. 1. The ciphering and deciphering procedures are described in the following subsections.

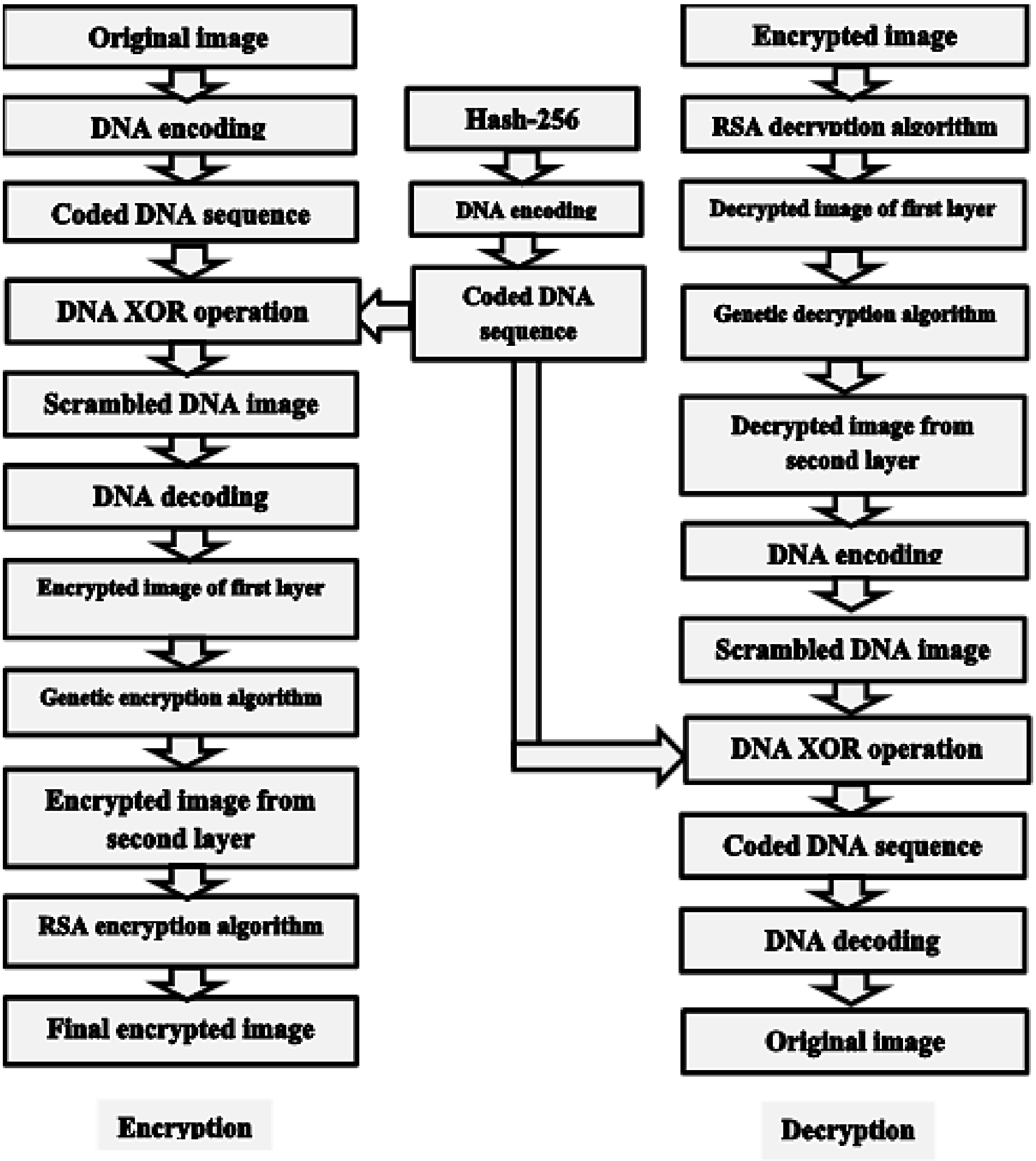


Fig. 1 Block Diagram of the Presented Image Enciphering and Deciphering Model.

3.1.Image Encryption

The procedure for image encryption is summarized in the following steps:

- Step 1: Transform the plain grayscale image K_1 (M, N) to a binary matrix K_2 (M × N, 8).
- Step 2: Reshape binary matrix K_2 to a one-dimensional array K_3 (1, M × N × 8).
- Step 3: Apply rule (1) as described in the DNA encoding laws in Table 1 on the binary array K_3 to obtain the coded DNA array K_4 (1, M × N × 4).
- Step 4: Use SHA-256 to produce 256-bit hash values A_1(M, N), then these values are converted to their equivalent binary matrix A_2 (M × N, 8). Next, A_2 is mutated to a single-dimensional matrix A_3 with the same size as K_3 (1, M × N × 8).
- Step 5: Run rule (1) as mentioned in the DNA encoding rules in Table 1 on the binary array A_3 to obtain the coded DNA array A_4 (1, M × N × 4).
- Step 6: Add the two arrays K_4 and A_4 based on the DNA XOR process rules in Table 2 to obtain a new scrambled array C_1 (1, M × N × 4) as described in Eq. (3).

$$\mathbf{C_1 = XOR\ (K_4, A_4)} \quad \mathbf{(3)}$$

- Step 7: The array C_1 is then decoded according to Table 1 /rule1 to acquire the binary decoded array C_2 (1, M × N × 8).
- Step 8: Reshape array C_2 into a binary matrix C_3 (M × N, 8).
- Step 9: Convert the binary matrix C_3 into a decimal matrix C_4 (M × N, 1).
- Step 10: The matrix C_4 is then reshaped to gain the ciphered image R (M, N) from the first encryption layer (DNA algorithm).

- Step 11: Divide R into equal blocks of size $(8,8)$ to obtain $\mathrm{L}_1, \mathrm{L}_2, \ldots, \mathrm{L}_{(\mathrm{M}\times\mathrm{N})/64}$. These blocks are then converted into binary matrices $\mathrm{L}_1{}', \mathrm{L}_2{}', \ldots., \mathrm{L}_{(\mathrm{M}\times\mathrm{N})/64}{}'$, where each block is of size $(64,\ 8)$.
- Step 12: Perform the crossover process according to Refs. [5] and [6] via circular shift operation of each bit in $\mathrm{L}_1{}', \mathrm{L}_2{}', \ldots., \mathrm{L}_{(\mathrm{M}\times\mathrm{N})/64}{}'$, two bits to the right, which yields the matrices $\mathrm{P}_1, \mathrm{P}_2, \ldots., \mathrm{P}_{(\mathrm{M}\times\mathrm{N})/64}$ as shown in Eq. (4).

$$\mathbf{P_1 = circshift\ (\,L_1{}', 2),\ \ P_2 = circshift\ (\,L_2{}', 2)\qquad P_{(M\times N)/64} = circshift\ \ (L_{(M\times N)/64}{}', 2)} \qquad \mathbf{(4)}$$

where $\mathrm{circshift}\,(\mathrm{L}, \mathrm{r})$ indicates the right cyclic shift of values in the array L by r positions. After that, these binary matrices are converted to their decimal equivalent matrices $\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_{(\mathrm{M}\times\mathrm{N})/64}$ with size $(64, 1)$.

- Step 13: Reshape the matrices $\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_{(\mathrm{M}\times\mathrm{N})/64}$ into matrices with size $(8, 8)$ to obtain $\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_{(\mathrm{M}\times\mathrm{N})/64}$.
- Step 14: Generate arbitrary matrices $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_{(\mathrm{M}\times\mathrm{N})/64}$ with the same size of $\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_{(\mathrm{M}\times\mathrm{N})/64}$.
- Step 15: Carry out the mutation process by applying the BITXOR operation between the two matrices obtained from Steps 13 and 14 to produce $\mathrm{Z}_1, \mathrm{Z}_2, \ldots, \mathrm{Z}_{(\mathrm{M}\times\mathrm{N})/64}$ according to Eq. (5).

$$\mathrm{Z_1(i,j) = bitxor(S_1(i,j), X_1(i,j))}$$
$$\mathrm{Z_2(i,j) = bitxor\big(S_2(i,j), X_2(i,j)\big)} \rightarrow \mathrm{Z_{(M\times N)/64}(i,j) = bitxor\left(S_{(M\times N)/64}(i,j), X_{(M\times N)/64}(i,j)\right)} \qquad \mathbf{(5)}$$

- Step 16: Implement the mod operation on $\mathrm{Z}_1, \mathrm{Z}_2, \ldots, \mathrm{Z}_{(\mathrm{M}\times\mathrm{N})/64}$ to gain $\mathrm{E}_1, \mathrm{E}_2, \ldots., \mathrm{E}_{(\mathrm{M}\times\mathrm{N})/64}$ according to Eq. (6).

$$\mathrm{E_1(i,j) = Z_1\ (i,j)\ mod\ 256,\ E_2(i,j) = Z_2\ (i,j)\ mod\ 256} \rightarrow \mathrm{E_{(M\times N)/64}(i,j) = Z_{(M\times N)/64}\ (i,j)\ mod\ 256} \qquad \mathbf{(6)}$$

- Step 17: Convert $\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_{(\mathrm{M}\times\mathrm{N})/64}$ to their corresponding binary matrices $\mathrm{Z}_1{}', \mathrm{Z}_2{}', \ldots, \mathrm{Z}_{(\mathrm{M}\times\mathrm{N})/64}{}'$ of size (64, 8). Then, NOT operation (~) is applied to accomplish the mutation operation by flipping the bit value to get the matrices $\mathrm{Y}_1, \mathrm{Y}_2, \ldots, \mathrm{Y}_{(\mathrm{M}\times\mathrm{N})/64}$, as defined in Eq. (7).

$$\mathrm{Y_1(i,j) = {\sim}(Z_1{}'(i,j)),\ Y_2(i,j) = {\sim}(Z_2{}'(i,j))} \rightarrow \mathrm{Y_{(M\times N)/64}(i,j) = {\sim}(Z_{(M\times N)/64}{}'(i,j))} \qquad \mathbf{(7)}$$

- Step 18: Transform $\mathrm{Y}_1, \mathrm{Y}_2, \ldots, \mathrm{Y}_{(\mathrm{M}\times\mathrm{N})/64}$ to their equivalent decimal matrices $\mathrm{Y}_1{}', \mathrm{Y}_2{}', \ldots, \mathrm{Y}_{(\mathrm{M}\times\mathrm{N})/64}{}'$, each of size $(64, 1)$. After this, $\mathrm{Y}_1{}', \mathrm{Y}_2{}', \ldots, \mathrm{Y}_{(\mathrm{M}\times\mathrm{N})/64}{}'$ are reshaped into $(8, 8)$ blocks to produce $\mathrm{W}_1, \mathrm{W}_2, \ldots., \mathrm{W}_{(\mathrm{M}\times\mathrm{N})/64}$.
- Step 19: Reconstruct the blocks $\mathrm{W}_1, \mathrm{W}_2, \ldots, \mathrm{W}_{(\mathrm{M}\times\mathrm{N})/64}$ to acquire the ciphered image W of size (M, N) from the second encryption layer (genetic algorithm).
- Step 20: Utilize RSA encryption on the matrix W via the public key e and n, according to Eq. (1), to obtain the eventual enciphered image D (M, N) from the third encryption layer (RSA algorithm) using Eq. (8).

$$\mathrm{D(i,j) = (W\ (i,j))^e\ mod\ n} \qquad \mathbf{(8)}$$

The input color image is divided into three color matrices (R, G, and B). Then, the steps 1-20 are implemented on each matrix separately. The final step is combining the three encrypted blocks to acquire the ciphered image.

3.2.Image Decryption

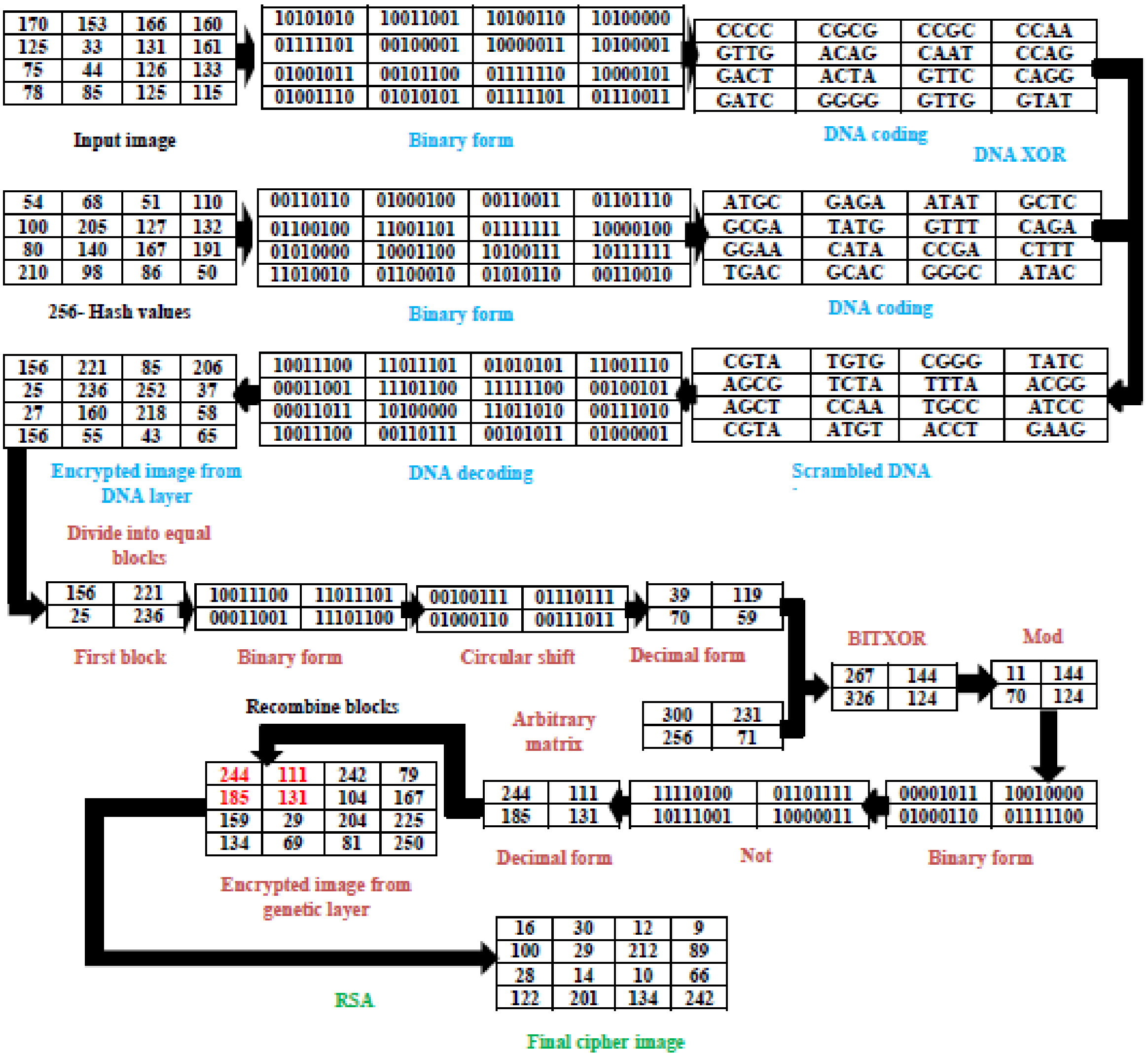
The deciphering procedures at the receptor are similar to the ciphering procedures at the transmitter; however, in the opposite order. The deciphering operation demands the secret keys to decipher the encrypted image. The decryption begins from the cipher image D and ends with the original input image K_1. Algorithm (1) exhibits the ciphering operation of the presented method.

3.3.Case Study of the Proposed Scheme

An example of the presented image enciphering process is clarified in Fig. 2. The input image has a size of (4×4) in this example. First, the plain image is changed from a decimal format to its corresponding binary format. Also, the hash values obtained from the SHA-256 function are transformed into their binary form. In the next step, the DNA coding rules are applied to the resultant two binary arrays to gain the encoded DNA arrays. This action is followed by employing the DNA XOR operation on coded arrays to get the chaotic DNA matrix. The final step in this phase is using the decoding rules of DNA to extract the encrypted image from the DNA algorithm layer. Second, the output matrix from the DNA layer is divided into equal blocks: the block size is assumed to be (2×2), and the first block is taken as an example. The mentioned block is converted to a binary format, and then a circular shift process is implemented on the block bits. Next, the gained values are converted into their decimal state. An arbitrary matrix of size 2×2 is produced, and two decimal matrices of (2×2) size are BITXORed, followed by the mode operation to return the matrix values in the range [0-255]. In the subsequent step, the decimal values obtained from the previous step are converted to a binary state to apply the NOT operation. After utilizing the NOT operation, the resulting bits for each pixel are transformed into their decimal values. The ultimate step in this phase is recombining the remaining blocks after implementing the above procedures on them to give the encrypted image from the genetic algorithm layer. Finally, the RSA algorithm is applied to the subsequent image from the genetic layer to get the final cipher image. If this technique is applied to a 256 × 256 image, then the obtained image after each encryption phase is given in Fig. 3.

Algorithm 1: The Image Encryption Process of the Proposed Cryptosystem.

```
Input: Plain image $K_1$
$[M, N]$=size ( $K_1$)
$K_2 = $ dec2 bin ($K_1$)                                   % $K_2$ of size ($M \times N, 8$)
$K_3 = $ reshape ($K_2, [1, M \times N \times 8]$)
$K_4 = $ DNA Encode ($K_3$)                                 % $K_4$ of size ($1, M \times N \times 4$)
Apply SHA-256 to produce 256-bit hash values $A_1$          % $A_1$ of size ($M, N$)
$A_2 = $ dec2 bin ($A_1$)                                   % $A_2$ of size ($M \times N, 8$)
$A_3 = $ reshape ($A_2, [1, M \times N \times 8]$)
$A_4 = $ DNA Encode ($A_3$)                                 % $A_4$ of size ($1, M \times N \times 4$)
$C_1 = $ DNA XOR ($K_4, A_4$)                               % $C_1$ of size ($1, M \times N \times 4$)
$C_2 = $ DNA Decode ($C_1$)                                 % $C_2$ of size ($1, M \times N \times 8$)
$C_3 = $ reshape ($C_2, [M \times N, 8]$)
$C_4 = $ bin2 dec ($C_3$)                                   % $C_4$ of size ($M \times N, 1$)
$R = $ reshape ($C_4, [M, N]$)                              % $R$ ( $M, N$) is the encrypted image from the DNA algorithm
$L_1, L_2, \ldots, L_{(M\times N)/64} = $ Divide R into (8, 8) blocks
$L_1', L_2', \ldots, L_{(M\times N)/64}' = $ dec2bin $(L_1, L_2, \ldots, L_{(M\times N)/64})$   % $L_1', L_2', \ldots, L_{(M\times N)/64}'$ of size (64, 8)
$P_1, P_2, \ldots, P_{(M\times N)/64} = $ circshift ( $L_1', L_2', \ldots, L_{(M\times N)/64}'$ , 2 )
$T_1, T_2, \ldots, T_{(M\times N)/64} = $ bin2dec ($P_1, P_2, \ldots, P_{(M\times N)/64}$)   % $T_1, T_2, \ldots, T_{(M\times N)/64}$ of size (64, 1)
$S_1, S_2, \ldots, S_{(M\times N)/64} = $ reshape $(T_1, T_2, \ldots, T_{(M\times N)/64}, [8, 8])$
Generate $X_1, X_2, \ldots, X_{(M\times N)/64}$            % $X_1, X_2, \ldots, X_{(M\times N)/64}$ with the same size of $S_1, S_2, \ldots, S_{(M\times N)/64}$
$Z_1, Z_2, \ldots, Z_{(M\times N)/64} = $ ( $S_1, S_2, \ldots, S_{(M\times N)/64}$) bitxor ($X_1, X_2, \ldots, X_{(M\times N)/64}$)
$E_1, E_2, \ldots, E_{(M\times N)/64} = (Z_1, Z_2, \ldots, Z_{(M\times N)/64})$ mod 256
$Z_1', Z_2', \ldots, Z_{(M\times N)/64}' = $ dec2bin ($E_1, E_2, \ldots, E_{(M\times N)/64}$)   % $Z_1', Z_2', \ldots, Z_{(M\times N)/64}'$ of size (64, 8)
$Y_1, Y_2, \ldots, Y_{(M\times N)/64} = \sim$ ( $Z_1', Z_2', \ldots, Z_{(M\times N)/64}'$)
$Y_1', Y_2', \ldots, Y_{(M\times N)/64}' = $ bin2dec ($Y_1, Y_2, \ldots, Y_{(M\times N)/64}$)   % $Y_1', Y_2', \ldots, Y_{(M\times N)/64}'$ of size (64, 1)
$W_1, W_2, \ldots, W_{(M\times N)/64} = $ reshape ( $Y_1', Y_2', \ldots, Y_{(M\times N)/64}', [8, 8]$)
$W = $ Recombine ($W_1, W_2, \ldots, W_{(M\times N)/64}$)   % $W$ ($M, N$) is the encrypted image from the genetic algorithm
$D(i, j) = (W(i, j))^e \bmod n$                             %$D$ ($M, N$) is the final encrypted image from the RSA algorithm
Output: Cipher image D
```



170	153	166	160
125	33	131	161
75	44	126	133
78	85	125	115

10101010	10011001	10100110	10100000
01111101	00100001	10000011	10100001
01001011	00101100	01111110	10000101
01001110	01010101	01111101	01110011

CCCC	CGCG	CCGC	CCAA
GTTG	ACAG	CAAT	CCAG
GACT	ACTA	GTTC	CAGG
GATC	GGGG	GTTG	GTAT

54	68	51	110
100	205	127	132
80	140	167	191
210	98	86	50

00110110	01000100	00110011	01101110
01100100	11001101	01111111	10000100
01010000	10001100	10100111	10111111
11010010	01100010	01010110	00110010

ATGC	GAGA	ATAT	GCTC
GCGA	TATG	GTTT	CAGA
GGAA	CATA	CCGA	CTTT
TGAC	GCAC	GGGC	ATAC

CGTA	TGTG	CGGG	TATC
AGCG	TCTA	TTTA	ACGG
AGCT	CCAA	TGCC	ATCC
CGTA	ATGT	ACCT	GAAG

10011100	11011101	01010101	11001110
00011001	11101100	11111100	00100101
00011011	10100000	11011010	00111010
10011100	00110111	00101011	01000001

156	221	85	206
25	236	252	37
27	160	218	58
156	55	43	65

156	221
25	236

10011100	11011101
00011001	11101100

00100111	01110111
01000110	00111011

39	119
70	59

300	231
256	71

267	144
326	124

11	144
70	124

00001011	10010000
01000110	01111100

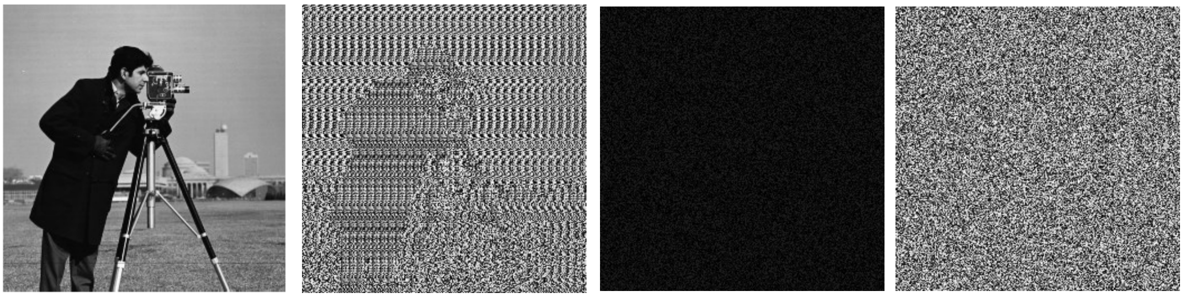
11110100	01101111
10111001	10000011

244	111
185	131

244	111	242	79
185	131	104	167
159	29	204	225
134	69	81	250

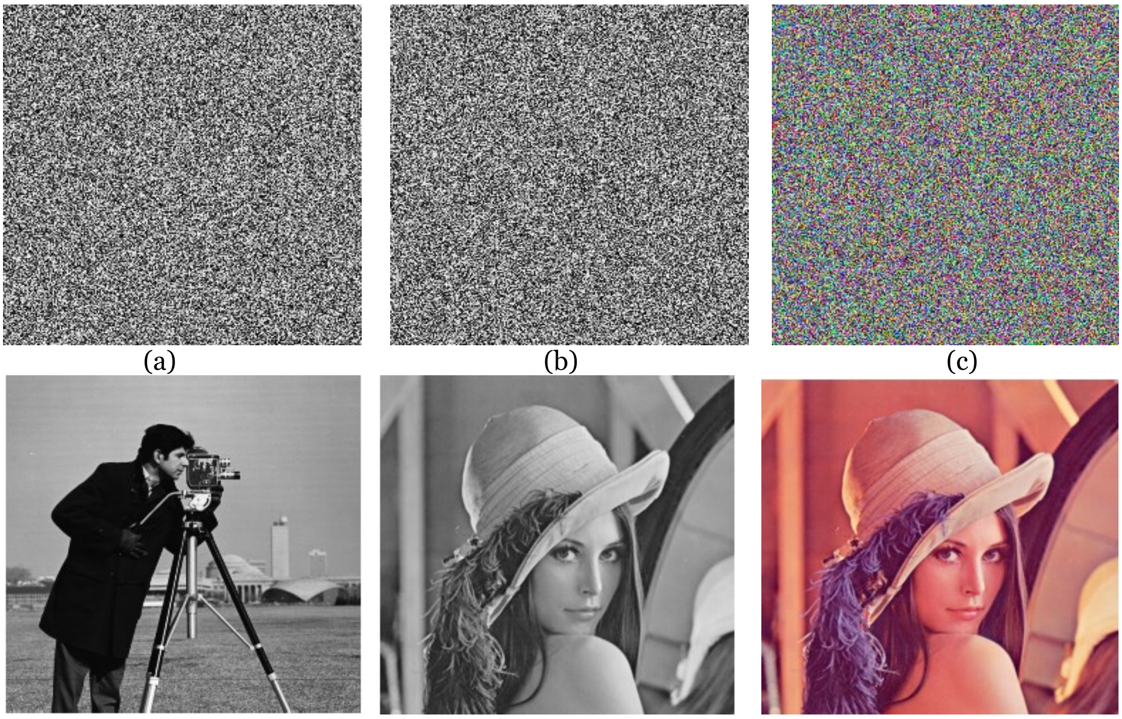
16	30	12	9
100	29	212	89
28	14	10	66
122	201	134	242

Fig. 2 An Example of the Image Enciphering.



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

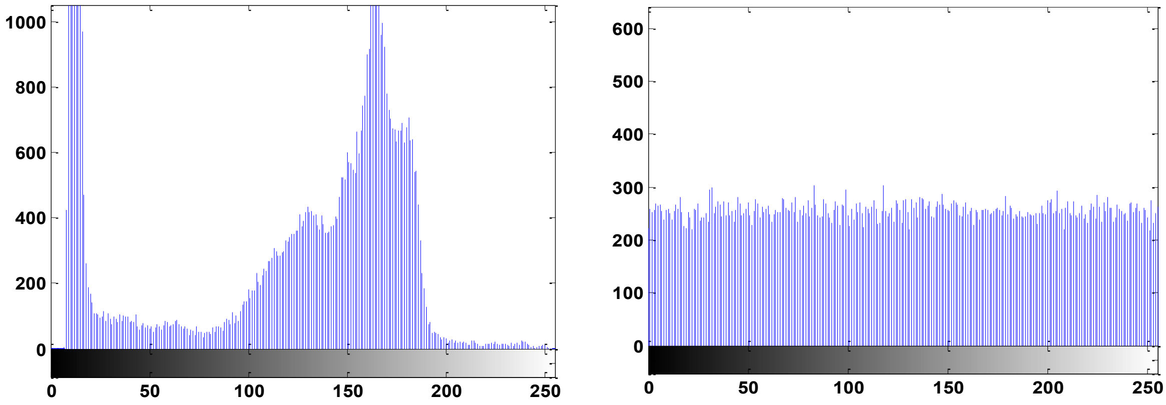
Fig. 3 Example of the Enciphering Operation on 256 × 256 Image (a) Plain Image, (b) Ciphered Image after the DNA Algorithm, (c) Ciphered Image after the DNA and Genetic Algorithms, and (d) Ciphered Image after the DNA, Genetic, and RSA Algorithms (Final Cipher Image).



(a) (b) (c)

(d) (e) (f)

Fig. 4 Enciphering and Deciphering Results (a-c) Cipher Images of Cameraman, Gray Lena, and Color Lena, respectively (d-f) Decipher of Images (a-c).



(a) (b)

Fig. 5 Histogram Analysis (a) Cameraman Image Histogram (b) Encrypted Cameraman Image Histogram.

3.4. Entropy Analysis

This test is utilized to quantify the randomness of gray pixel values in the digital image. The information entropy is described by Eq. (9).

$$\mathbf{H(x) = -\sum_{i=0}^{L-1} P(x_i) \log_2 P(x_i)} \qquad \mathbf{(9)}$$

where x_i indicates the i th possible value, P (x_i) symbolizes the probability of x_i, and L points to the gray values. The optimal value of entropy for an image is eight. The image cryptosystem is effective when the entropy outcome of the ciphered image is near eight [18, 19]. The entropy outcomes for the plain and cipher images are demonstrated in Table 3. It is evident from Table 3 that the entropy scores of the enciphered images for all test images are quite close to eight. This result implies that the plain image entropy is significantly enhanced after the encryption operation, and thus, the presented method possesses high randomness.

3.5. Correlation Coefficient Analysis

Generally, every pixel in the plain image is highly correlated with neighboring pixels. A typical image cryptosystem should reduce this correlation in the ciphered image [25]. Five thousand pairs of neighboring pixels are randomly chosen in the tests from the input image and the encrypted version along Horizontal, Vertical, and Diagonal directions. The correlation coefficient values for these pixels are computed according to Eq. (10).

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)} \times \sqrt{D(y)}}, \qquad \mathrm{cov}\,(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i\ , \quad E(y) = \frac{1}{N}\sum_{i=1}^{N} y_i, \quad D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x_i))^2, \quad D(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - E(y_i))^2 \qquad \mathbf{(10)}$$

where N indicates the number of neighboring pixels selected from the input and output images to compute the correlation test, x_i and y_i refer to the two neighboring pixels of plain and cipher images. The correlation scores for the test images and their enciphered images provided by the presented approach are reported in Table 4. It is clear from Table 4 that the correlation among nearby pixels in the input image is large (close to one), while the correlation in the output image is extremely small (close to zero). On the other hand, the correlation distributions of Peppers image and its enciphered version of Red, Green, and Blue channels in 3 directions are displayed in Fig. 6. The correlation of each channel in Horizontal, Vertical, and Diagonal directions for the original image is quite strong, as illustrated in Figs. 6 (b-j). In contrast, this correlation in the cipher image is removed, as clarified in Figs. 6 (l-t). Table 4 and Fig. 6 reveal that the suggested technique can reduce the relationship between adjacent pixels of the input image, and it can overcome the correlation analysis.

3.6. Differential Analysis

Two common criteria employed to assess the robustness of the presented method to the differential attack, namely, number of pixels change rate (NPCR) and unified average changing intensity (UACI). NPCR and UACI are given by Eqs. (11) and (12), respectively.

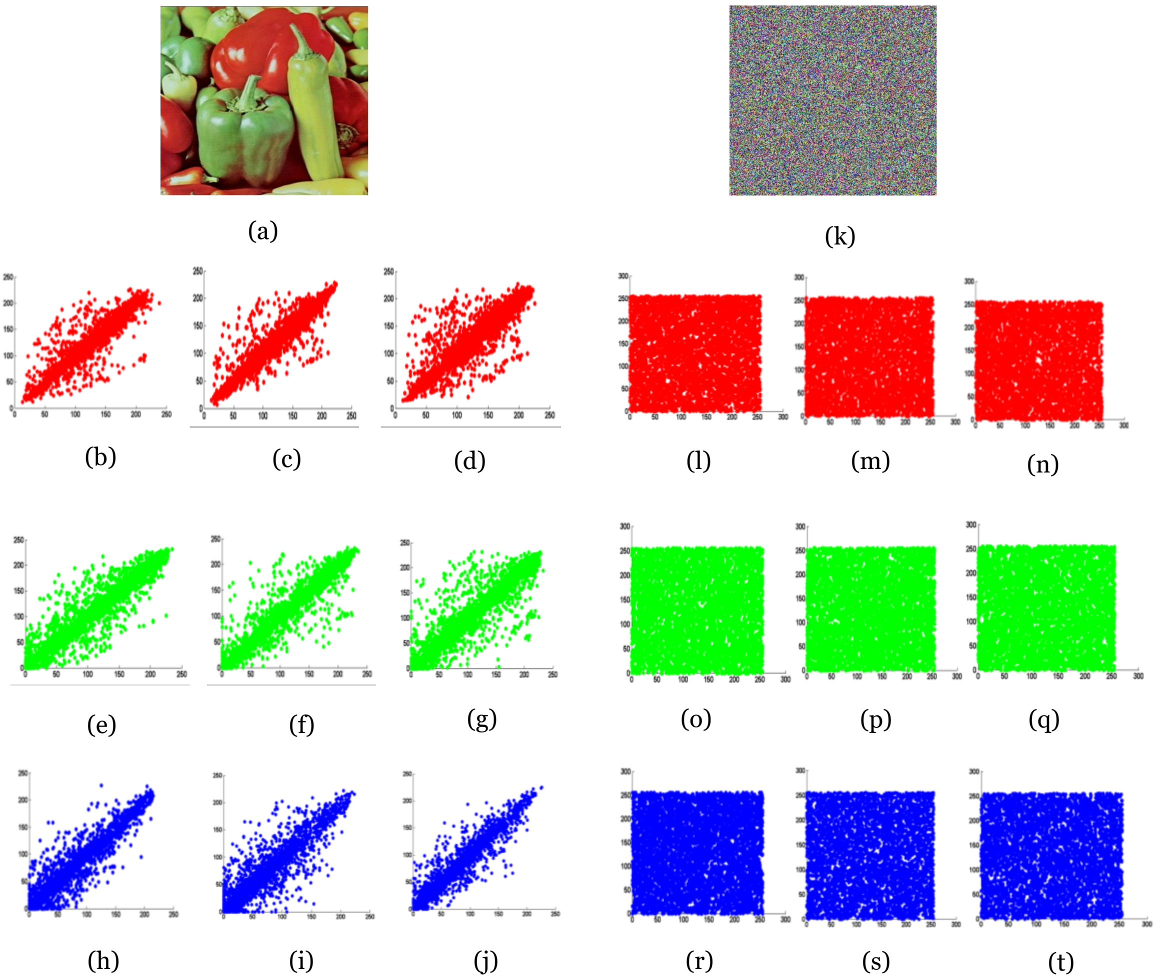

(a)
(k)
(b) (c) (d) (l) (m) (n)
(e) (f) (g) (o) (p) (q)
(h) (i) (j) (r) (s) (t)

Fig. 6 Correlation Coefficient Examination (a) Input Image, (b-d) Correlation Allocation for R Channel of Plain Image in Horizontal, Vertical, and Diagonal Directions, (e-g) Correlation Distribution for G Channel of Plain Image in Horizontal, Vertical, and Diagonal Directions, (h-j) Correlation Distribution for B Channel of Plain Image in Horizontal, Vertical, and Diagonal Directions, (k) Encrypted image, (l-n) Correlation Distribution for R channel of Encrypted Image in Horizontal, Vertical, and Diagonal Directions, (o-q) Correlation Allocation for G channel of Encrypted Image in Horizontal, Vertical, and Diagonal Directions, and (r-t) Correlation Allocation for B Channel of Enciphered Image in Horizontal, Vertical, and Diagonal Directions.

$$\mathbf{NPCR} = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%, \qquad D(i,j) = \begin{cases} 0 & \text{if } C_1(i,j) = C_2(i,j) \\ 1 & \text{if } C_1(i,j) \neq C_2(i,j) \end{cases} \qquad \mathbf{(11)}$$

$$\mathbf{UACI} = \frac{1}{M \times N}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\% \qquad \mathbf{(12)}$$

where (C_1, C_2) represent the ciphered images, whose input images differ only by a single pixel, M and N symbolize the height/width of C_1 and C_2, and D denotes the difference matrix between C_1 and C_2. The typical scores of NPCR and UACI in all cryptographic algorithms are 100% and 33.33%, respectively [4, 24]. The computation of the NPCR and UACI generated by the application of the suggested scheme upon the plain test images is given in Table 3. The outcomes in this table demonstrate that the scores of NPCR and UACI are quite close to the theoretical scores. Hence, the described cryptographic system possesses strong immunity against differential attack analysis.

3.7.MSE, PSNR, and SSIM Indicators

Mean square error (MSE) and peak signal-to-noise ratio (PSNR) are two famous tools to quantify the dissimilarity between the plain and cipher images. In contrast, the structural similarity index measure (SSIM) is employed to compute the likeness degree and devastation rate between the plain image and the encrypted image. MSE, PSNR, and SSIM are calculated by Eqs. (13), (14), and (15), respectively.

$$\mathbf{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} [I_1\,(I, j) - I_2\,(I, j)]^2 \quad (13)$$

$$\mathbf{PSNR} = 10 \times \log_{10} \left[\frac{255 \times 255}{MSE}\right] \quad (14)$$

$$\mathbf{SSIM}\,(I_1, I_2) = \frac{(2\mu_{I_1}\mu_{I_2} + \alpha)(2\sigma_{I_1 I_2} + \beta)}{(\mu_{I_1}^2 + \mu_{I_2}^2 + \alpha)(\sigma_{I_1}^2 + \sigma_{I_2}^2 + \beta)} \quad (15)$$

where I_1 and I_2 are the input and encrypted images, respectively, μ_{I_1} and μ_{I_2} denote the averages of I_1 and I_2, respectively, $\sigma_{I_1}^2$ and $\sigma_{I_2}^2$ represent the variance of I_1 and I_2, respectively, $\sigma_{I_1 I_2}$ refers to the I_1 and I_2 covariance, while α and β represent the variables. The MSE value should be large among the input and cipher images; in addition, the PSNR and SSIM values between the input and enciphered images must be low to get a better ciphering effect [3, 26]. The MSE, PSNR, and SSIM scores between the original images and their encrypted versions generated by the introduced mechanism are depicted in Table 3. According to Table 3, the MSE outcomes are so high. Inversely, PSNR and SSIM outcomes are quite low for the entire test image. This result manifests that the presented cryptosystem is robust, and it can demolish the likeness between the input and its output enciphered versions in the ciphering process.

Table 3 The Results of Information Entropy, NPCR, UACI, MSE, PSNR, and SSIM for the Test Images.

Image	Entropy (plain)	Entropy (cipher)	NPCR (%)	UACI (%)	MSE	PSNR (dB)	SSIM
Cameraman (256 × 256)	7.0097	7.9971	99.6246	33.2439	9446.9	8.3779	0.0095
Cameraman (512 × 512)	7.0482	7.9993	99.6101	33.3936	9372.6	8.4122	0.0339
Gray Lena (256 × 256)	7.4318	7.9972	99.6247	33.6132	7726.2	9.2512	0.0098
Gray Lena (512 × 512)	7.4455	7.9993	99.6302	33.6626	7781.4	9.2202	0.0368
Gray Baboon (256 × 256)	7.2285	7.9974	99.6296	33.4336	6950.1	9.7109	0.0065
Gray Baboon (512 × 512)	7.3585	7.9994	99.6531	33.8125	7238.2	9.5345	0.0344
Gray Peppers (256 × 256)	7.5647	7.9973	99.6789	33.6521	8418.5	8.8785	0.0086
Gray Peppers (512 × 512)	7.5712	7.9995	99.6945	33.7096	8470.1	8.8519	0.0354
Color Lena (256 × 256)	7.7330	7.9991	99.6625	33.4069	8925.4	8.6245	0.0101
Color Lena (512 × 512)	7.7503	7.9998	99.6490	33.3410	8954.4	8.6104	0.0364
Color Baboon (256 × 256)	7.6781	7.9990	99.6806	33.2719	8257.6	8.9623	0.0068
Color Baboon (512 × 512)	7.7626	7.9998	99.6253	33.7441	8608.4	8.7816	0.0309
Color Peppers (256 × 256)	7.6976	7.9989	99.6963	33.4921	1004.8	8.1098	0.0090
Color Peppers (512 × 512)	7.6612	7.9997	99.6524	33.6382	1010.5	8.0856	0.0311

Table 4 The Outcomes of Correlation Coefficients of the Original Image and its Encrypted Image.

Image	Original image			Encrypted image		
	H	V	D	H	V	D
Cameraman (256 × 256)	0.9352	0.9603	0.9103	-0.0159	-0.0024	0.0032
Cameraman (512 × 512)	0.9844	0.9902	0.9744	0.0132	-0.0172	-0.0171
Gray Lena (256 × 256)	0.9465	0.9691	0.9276	0.0099	-0.0151	-0.0018
Gray Lena (512 × 512)	0.9673	0.9850	0.9581	-0.0126	0.0073	-0.0044
Gray Baboon (256 × 256)	0.8639	0.8267	0.7989	-0.0091	-0.0124	-0.0105
Gray Baboon (512 × 512)	0.8616	0.7409	0.7295	-0.0149	-0.0121	0.0219
Gray Peppers (256 × 256)	0.9600	0.9660	0.9364	-0.0048	-0.0203	-0.0031
Gray Peppers (512 × 512)	0.9705	0.9685	0.9644	-0.0052	0.0110	-0.0023
Color Lena (256 × 256)	0.9536	0.9809	0.9374	-0.0161	-0.0129	-0.0074
Color Lena (512 × 512)	0.9818	0.9894	0.9708	0.0032	-0.0086	-0.0018
Color Baboon (256 × 256)	0.9449	0.9153	0.9040	-0.0087	-0.0115	0.0030
Color Baboon (512 × 512)	0.9248	0.8682	0.8621	0.0095	0.0043	-0.0107
Color Peppers (256 × 256)	0.9454	0.9573	0.9169	-0.0227	-0.0181	0.0020
Color Peppers (512 × 512)	0.9666	0.9529	0.9274	0.0150	-0.0029	-0.0017

3.8. Execution Time Analysis

A good image cryptosystem should have the shortest running time [10]. The proposed approach is executed in Windows 7, MATLAB R2013a, and the computer used is a 4.00 GB RAM, 2.40 GHz CPU, Intel Core i3. Table 5 is presented to show the total time consumed by the presented scheme to encipher different kinds of images of different sizes. Table 5 clarifies that the consuming time to encrypt images increases with the image size. Also, color images take a longer time than gray images of the same size. The total encryption time relies upon the type and size of the input image. As in Table 5, the suggested scheme is fast and effective for encrypting different kinds of images of different sizes. To sum up, the proposed approach is effective and can be used in real-time applications to secure sensitive information in digital images.

Table 5 Enciphering Time (seconds) of the Sample Images in the Proposed Method.

Image	Encryption time (s)
Cameraman (256 × 256)	0.274002
Gray Lena (256 × 256)	0.485300
Gray Baboon (256 × 256)	0.307338
Gray Peppers (256 × 256)	0.456244
Color Lena (256 × 256)	1.698029
Color Baboon (256 × 256)	1.825644
Color Peppers (256 × 256)	1.706693
Cameraman (512 × 512)	0.393160
Gray Lena (512 × 512)	0.568760
Gray Baboon (512 × 512)	0.452804
Gray Peppers (512 × 512)	0.570416
Color Lena (512 × 512)	2.832915
Color Baboon (512 × 512)	2.887692
Color Peppers (512 × 512)	2.785491

3.9. Resilience Examination

During the image transmission process, the encrypted image can be destroyed by several sorts of attacks. A good enciphering scheme must be capable of restoring important information from an encrypted image under various attacks [15, 24, 27-34]. The immunity of the proposed cryptosystem against common cryptographic attacks on the encrypted test images (512×512) is investigated in this section. These attacks are cropping 50 % of the cipher image, salt and pepper noise (intensity 30%), Gaussian noise (σ^2=0.1), sharpening, blurring (len=10, θ=45), correcting Gamma (γ=0.5), equalizing histogram, and JPEG compression (QF=90). The outcomes of the attacks in terms of correlation, PSNR, and SSIM are given in Table 6. Also, decryption outcomes for the gray Lena image after employing the attacks upon the ciphered image are demonstrated in Fig. 7. According to Table 6 and Fig. 7, the presented approach is capable of recovering the input image under image processing attacks with acceptable visual quality, and the reconstructed image is still recognizable. Hence, it can be affirmed that the described approach has strong immunity to several known image processing attacks. Ultimately, it can be justified that the presented method has strong invulnerability against famous image processing threats and can resist them with sufficient security. The approach strength arises from a combination of several known strong algorithms that have very powerful intrinsic mathematical characteristics to produce efficient and strong image security algorithms.

Table 6 Correlation, PSNR, and SSIM after adding Image Processing Attacks on the Cipher Images.

Image		Attacks							
		Cropping	Salt and pepper noise	Gaussian noise	Sharpening	Blurring	Histogram equalization	Gamma correction	JPEG compression
Cameraman	r_{xy}	0.4477	0.6558	0.9259	0.6610	0.0286	0.9997	0.4234	0.3116
	PSNR	11.4429	13.6175	17.0637	12.9923	8.5252	44.1714	10.8638	8.4091
	SSIM	0.5057	0.2863	0.4993	0.2591	0.0447	0.9917	0.1878	0.0309
Gray Lena	r_{xy}	0.4309	0.5894	0.8835	0.6376	0.0320	0.9995	0.3522	0.4140
	PSNR	12.4458	14.4753	17.0989	13.4826	9.3553	44.1620	11.0844	9.2323
	SSIM	0.5132	0.3336	0.5802	0.2841	0.0479	0.9960	0.1710	0.0338
Gray Baboon	r_{xy}	0.3091	0.5526	0.8575	0.6254	0.0326	0.9993	0.4294	0.4011
	PSNR	12.4018	14.7851	17.0445	13.5722	9.6457	44.1501	12.0316	9.5245
	SSIM	0.5079	0.4821	0.7500	0.4726	0.0570	0.9984	0.3467	0.0295
Gray Peppers	r_{xy}	0.4617	0.6197	0.9055	0.6552	0.0313	0.9992	0.4136	0.4020
	PSNR	12.1244	14.0756	17.0803	13.4871	8.9748	41.7371	11.3217	8.8651
	SSIM	0.5107	0.3167	0.5638	0.2669	0.0424	0.9929	0.1759	0.0351
Color Lena	r_{xy}	0.3594	0.5411	0.8670	0.5759	0.0228	0.9976	0.3236	0.4411
	PSNR	11.6637	13.8460	17.3888	13.2826	8.7366	37.7744	10.6574	8.6179
	SSIM	0.5116	0.3132	0.5894	0.2632	0.0451	0.9844	0.1651	0.0330
Color Baboon	r_{xy}	0.4881	0.6218	0.9046	0.6503	0.0274	0.9967	0.3760	0.4012
	PSNR	11.7529	13.9824	17.3012	13.3854	8.8890	35.0836	10.8978	8.7753
	SSIM	0.5064	0.4856	0.7771	0.4604	0.0488	0.9867	0.2924	0.0272
Color Peppers	r_{xy}	0.4020	0.5904	0.8987	0.5813	0.0207	0.9995	0.2759	0.3014
	PSNR	11.1636	13.3059	17.1472	13.0392	8.1900	44.0977	9.9232	8.0910
	SSIM	0.5080	0.2954	0.5463	0.2391	0.0375	0.9941	0.1327	0.0318

4.COMPARATIVE ANALYSIS

To analyze the effectiveness of the presented scheme, this section exhibits the security comparison between this approach and other similar works. The outcomes of this comparison are shown in Table 7 for the test images of different sizes based on the entropy, NPCR, UACI, and correlation coefficient in 3 directions. In Table 7, the entropy scores are closer to 8 than the other methods. Besides, the correlation outcomes are smaller, and the NPCR and UACI scores are closer to the typical scores than some current techniques. According to the outcomes in Table 7, it can be deduced that the presented technique possesses good security performance compared to other existing algorithms. Hence, the proposed work is very adequate for image enciphering and can be implemented for secure image transmission.

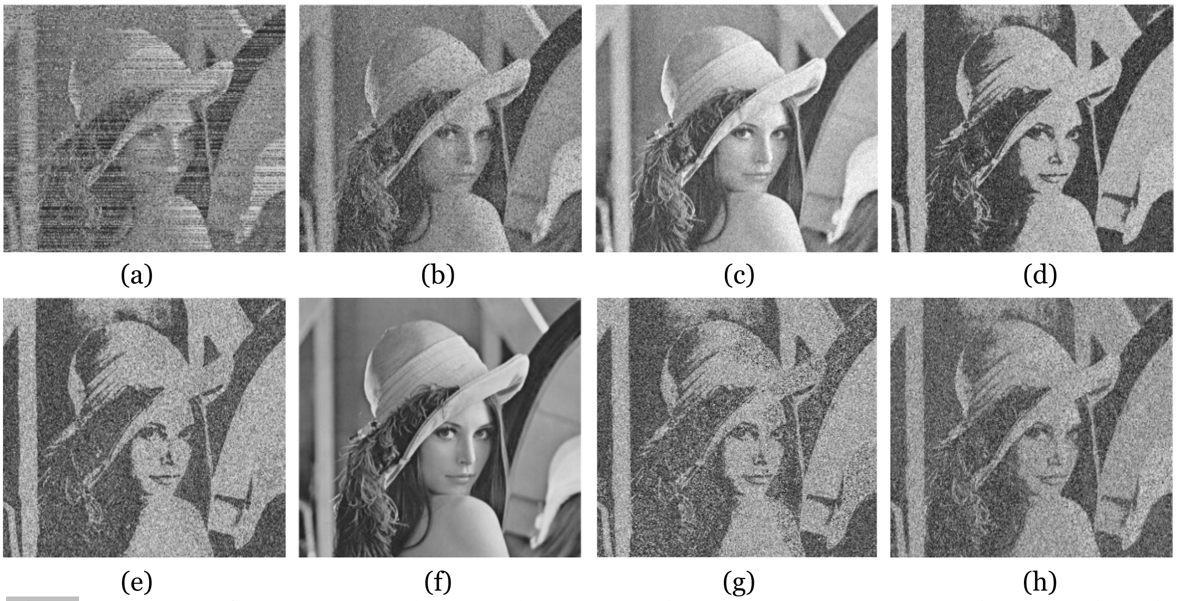


Fig. 7 Outcomes of Immunity Analysis on the Decrypted Image (a) Cropping Attack (50%), (b) Salt and Pepper Noise Attack (30%), (c) Gaussian Noise Attack ($\sigma^2 = 0.1$) (d) Sharpening Attack, (e) Blurring Attack ($\text{len} = 10, \theta = 45$), (f) Histogram Equalization Attack, (g) Gamma Correction Attack ($\gamma = 0.5$), and (h) JPEG Compression Attack (QF=90).

6.CONCLUSIONS

The major conclusions of the presented article are:

- The comparative analysis results show that the average percentage of improvement obtained by the proposed approach is better than or at par with compared similar approaches. For example, the gain in the entropy, NPCR, UACI, horizontal correlation coefficient, vertical correlation coefficient, and diagonal correlation coefficient parameters for gray Lena of size (256 × 256) was 1.18%, 0.86%, 3.5%, 0.99%, 1.555%, and 0.02%, respectively. It is concluded from these results that the proposed approach has better security in terms of randomness, confusion, and diffusion of the protected image.
- Histogram, differential analysis, execution time, MSE, PSNR, SSIM, and robustness analyses are another group of parameters used to check the strength of the proposed method. These tests prove that the proposed cryptosystem possesses a flat histogram, high values of MSE, and low values of PSNR and SSIM.
- The experimental results reveal that the proposed methodology has a fast encryption/decryption time (≈ 1.23 seconds) and high resistance against several famous cryptographic attacks, i.e., cropping, compression, and noise, and minimal computational complexity. Hence, this approach can be regarded as a good competitor for securing digital images in unsafe environments.
- This method is suitable for encrypting/decrypting different types and sizes of digital images. Thus, it can be widely utilized for secure transferring of image information.
- In the future, the proposed digital image cryptosystem will be improved in various ways. First, the scheme is integrated with another technology, such as chaotic maps or a computational genetic approach, to enhance the system's security. Second, image compression techniques can be employed to speed up the encryption process. Third, the suggested work can be implemented to encrypt other types of digital images, such as biometric, remote sensing, or medical images. Eventually, this approach can be accelerated by implementing it on an FPGA kit.

Table 7 The Results of the Presented Cryptosystem Performance in contrast to Existing Methods in Terms of Entropy, Correlation, NPCR, and UACI Measures.

Method	Image	Entropy	Correlation coefficient			NPCR (%)	UACI (%)
			H	V	D		
Proposed	Gray Lena (256 × 256)	7.9972	0.0099	-0.0151	-0.0018	99.6247	33.6132
Ref. [10]		7.9973	−0.0002	−0.0015	−0.0008	99.6152	28.6180
Ref. [14]		7.9854	0.0002	0.0024	−0.0032	99.7017	28.2970
Ref. [17]		-	-	-	-	99.6170	33.4199
Proposed	Gray Baboon (256 × 256)	7.9974	-0.0091	-0.0124	-0.0105	99.6296	33.4336
Ref. [16]		7.99710	−0.0022	−0.0064	−0.0023	99.607	33.362
Ref. [17]		-	-	-	-	99.6399	33.3027
Proposed	Gray Peppers (256 × 256)	7.9973	-0.0048	-0.0203	-0.0031	99.6789	33.6521
Ref. [16]		7.99707	−0.0020	−0.0046	−0.0070	-	-
Ref. [17]		-	-	-	-	99.6185	33.4211
Proposed	Gray Peppers (512 × 512)	7.9995	-0.0052	0.0110	-0.0023	99.6945	33.7096
Ref. [24]		7.9992	0.0023	−0.0002	−0.0008	99.60	33.35
Proposed	Color Lena (256 × 256)	7.9991	-0.0161	-0.0129	-0.0074	99.6625	33.4069
Ref. [9]		7.9968	-0.0123	0.0048	-0.0062	99.5998	33.3848
Ref. [12]		7.9971	− 0.0034	0.0021	-0.0003	99.6093	33.4684
Ref. [18]		-	-0.00005	-0.000169	0.000171	-	-
Ref. [19]		7.9971	0.0016	0.0046	−0.0006	99.63	33.44
Proposed	Color Lena (512 × 512)	7.9998	0.0032	-0.0086	-0.0018	99.6490	33.3410
Ref. [20]		7.99988	−0.0003	−0.006	0.00014	99.69561	33.81015
Proposed	Color Baboon (256 × 256)	7.9990	-0.0087	-0.0115	0.0030	99.6806	33.2719
Ref. [12]		7.9973	-0.0031	0.0048	0.0005	99.6133	33.4617
Ref. [19]		7.9973	−0.0010	0.0026	0.0003	99.6399	33.5197
Proposed	Color Baboon (512 × 512)	7.9998	0.0095	0.0043	-0.0107	99.6253	33.7441
Ref. [21]		7.9826	-0.0042	0.0099	0.005	99.5937	33.3803
Proposed	Color Peppers (256 × 256)	7.9989	-0.0227	-0.0181	0.0020	99.6963	33.4921
Ref. [12]		7.9973	0.0004	0.0035	− 0.0021	99.6135	33.4532
Ref. [19]		7.9974	−0.0007	0.0068	0.0004	99.619	33.5037
Proposed	Color Peppers (512 × 512)	7.9997	0.0150	-0.0029	-0.0017	99.6524	33.6382
Ref. [21]		7.9908	0.008	0.0062	0.0070	99.6138	33.4151

ACKNOWLEDGEMENTS

The authors thank the Engineering College, Diyala University, for encouragement and support to finish the study.

NOMENCLATURE

α	Euler's totient function
e	Public exponent
d	Private exponent
b , c	Prime random numbers
m	Modulus
M	Plaintext
C	Cipher text
XOR	Exclusive-OR
Mod	Modulo operation (reminder of division)
~	NOT operation
H(x)	Information entropy
P (x_i)	Probability of x_i
L	Gray values number
r_{xy}	Correlation coefficient
N	Number of neighboring pixels chosen from the input and output images
x_i, y_i	Two adjacent pixels of the input and enciphered images
C_1, C_2	Encrypted images that are equivalent to plain images differ (only) by a single pixel
M, N	Height and width of $C_1$and C_2
D	Difference matrix between $C_1$and C_2
I_1 , I_2	Original and encrypted images
μ_{I_1}, μ_{I_2}	Averages of I_1 and I_2
$\sigma^2_{I_1}$, $\sigma^2_{I_2}$	Variance of I_1 and I_2
$\sigma_{I_1 I_2}$	Covariance of I_1 and I_2
α, β	Variables

REFERENCES

[1] Abboud AJ. **Multifactor Authentication for Software Protection**. *Diyala Journal of Engineering Sciences* 2015; **8**(4): 479-492.

[2] Shehab JN, Radhi HY, Alhayali RAI. **Multimedia Cryptography Based on Liu and Chen Systems**. *Diyala Journal of Engineering Sciences* 2016; **9**(4): 24-35.

[3] Yousif SF, Abboud AJ, Alhumaima RS. **A New Image Encryption Based on Bit Replacing, Chaos and DNA Coding Techniques**. *Multimedia Tools and Applications* 2022; **81**: 27453-27493.

[4] Yousif SF, Abboud AJ, Radhi HY. **Robust Image Encryption with Scanning Technology, the El-Gamal Algorithm and Chaos Theory**. *IEEE Access* 2020; **8**: 155184-155209.

[5] Choudhary R, Abrol P. **Genetic Algorithm Based Image Cryptography to Enhance Security**. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 2017; **6**(6): 873-878.

[6] Arunpandian S, Mahesh K. **Image Encipherment Using Genetic Algorithm**. *International Journal of Engineering and Technology (IJET)* 2017; **9**(5): 3570- 3574.

[7] Ray A, Potnis A, Dwivedy P, Soofi S. **An Advance Approach of Image Encryption Using AES, Genetic Algorithm and RSA Algorithm**. *International Journal of Engineering and*

Applied Computer Science (IJEACS) 2017; **2**(8): 245- 249.

[8] Alsaffar DM, Almutiri AS, Alqahtani B, Alamri RM, Alqahtani HF, Alqahtani NN, Alshammari GM, Ali AA. **Image Encryption Based on AES and RSA Algorithms**. *3rd International Conference on Computer Applications & Information Security (ICCAIS)* 2020: 1-5.

[9] Rehman A, Liao X, Ashraf R, Ullahc S, Wang H. **A Color Image Encryption Technique Using Exclusive-OR with DNA Complementary Rules Based on Chaos Theory and SHA-2**. *Optik* 2018; **159**: 348-367.

[10] Rehmana A, Liao X, Hahsmi MA, Haider R. **An Efficient Mixed Inter-Intra Pixels Substitution at 2bits-Level for Image Encryption Technique Using DNA and Chaos**. *Optik* 2018; **153**: 117–134.

[11] Liu Y, Zhang J. **A Multidimensional Chaotic Image Encryption Algorithm based on DNA Coding**. *Multimedia Tools and Applications* 2020; **79**(29): 21579-21601.

[12] Huang L, Wang S, Xiang J, Sun Y. **Chaotic Color Image Encryption Scheme Using Deoxyribonucleic Acid (DNA) Coding Calculations and Arithmetic over the Galois Field**. *Mathematical Problems in Engineering* 2020: 1-22.

[13] Jithin K C, Sankar S. **Colour Image Encryption Algorithm Combining, Arnold Map, DNA Sequence Operation, and a Mandelbrot Set**. *Journal of Information Security and Applications* 2020; **50**: 102428.

[14] Zhang J, Fang D, Ren H. **Image Encryption Algorithm Based on DNA Encoding and Chaotic Maps**. *Mathematical Problems in Engineering* 2014: 1-10.

[15] Wu X, Kan H, Kurths J. **A New Color Image Encryption Scheme Based on DNA Sequences and Multiple Improved 1D Chaotic Maps**. *Applied Soft Computing* 2015; **37**: 24-39.

[16] Zheng J, Luo Z, Tang Z. **An Image Encryption Algorithm Based on Multichaotic System and DNA Coding**. *Discrete Dynamics in Nature and Society* 2020: 1-16.

[17] Zhu S, Deng X, Zhang W, Zhu C. **Image Encryption Scheme Based on Newly Designed Chaotic Map and Parallel DNA Coding**. *Mathematics* 2023; **11**(1): 231.

[18] Zhao J, Wang S, Zhang L. **Block Image Encryption Algorithm Based on Novel Chaos and DNA Encoding**. *Information* 2023; **14**(3): 150.

[19] Shraida GK, Younis HA, Al-Amiedy T A, Anbar M, Younis H A, Hasbullah I H. **An Efficient Color-Image Encryption Method Using DNA Sequence and Chaos Cipher**. *Cmc-Computers Materials & Continua* 2023; **75**(2): 2641-2654.

[20] Hafsa A, Gafs M, Malek J, Machhout M. **FPGA Implementation of Improved Security Approach for Medical Image Encryption and Decryption**. *Scientific Programming* 2021: 1-20.

[21] Zhang R, Yu L, Jiang D, Ding W, Song J, He K, Ding Q. **A Novel Plaintext-Related Color Image Encryption Scheme Based on Cellular Neural Network and Chen's Chaotic System**. *Symmetry* 2021; **13**(3): 393.

[22] Ye G, Jiao K, Huang X, Goi B M, Yap WS. **An Image Encryption Scheme Based on Public Key Cryptosystem and Quantum Logistic Map**. *Scientific Reports* 2020; **10**(1): 1-19.

[23] Jiao K, Ye G, Dong Y, Huang X, He J. **Image Encryption Scheme Based on a Generalized Arnold Map and RSA Algorithm**. *Security and Communication Networks* 2020: 1-14.

[24] Ghazvini M, Mirzadi M, Parvar N. **A Modified Method for Image Encryption Based on Chaotic Map and Genetic Algorithm**. *Multimedia Tools and Applications* 2020; **79**(37): 26927-26950.

[25] Yousif SF. **Grayscale Image Confusion and Diffusion Based on Multiple Chaotic Maps**. *1st International Scientific Conference of Engineering Sciences - 3rd Scientific Conference of Engineering Science (ISCES);* 2018: 114-119.

[26] Abdullah HN, Yousif SF, Valenzuela AA. **Wavelet Based Image Steganographic System Using Chaotic Signals**. *6th International Conference on Information Communication and Management*; 2016: 130-135.

[27] Zhang X, Wang L, Zhou Z, Niu Y. **A Chaos-Based Image Encryption Technique Utilizing Hilbert Curves and H-Fractals**. *IEEE Access* 2019; **7**: 74734-74746.

[28] Abboud AJ. **Protecting Documents Using Visual Cryptography**. *International Journal of Engineering Research and General Science* 2015; **3**(2): 464-470.

[29] Abdullah HN, Yousif SF, Jafar A. **Design of Efficient Chaos Based Image Steganographic System**. *SYLWAN Journal* 2015; **159** (7): 275- 284.

[30] Abboud AJ, Jassim SA. **Incremental Fusion of Partial Biometric Information**. *Mobile Multimedia/ Image Processing, Security, and Applications* 2012; **8406**: 169-177.

[31] Yousif SF. **Performance Comparison between RSA and El-Gamal Algorithms for Speech Data Encryption and Decryption**. *Diyala Journal of Engineering Sciences 2023*; **16**(1): 123-137.

[32] Shahab MM, Hardan SM, Hammoodi AS. **A New Transmission and Reception Algorithms for Improving the Performance of SISO/MIMO- OFDM Wireless Communication System**. *Tikrit Journal of Engineering Sciences* 2021; **28**(3): 146-158.

[33] Al-Kadhimi A M, Abdulkareem A E, Tsimenidis C C. **Performance Enhancement of LDPC Codes Based on Protograph Construction in 5G-NR Standard**. *Tikrit Journal of Engineering Sciences* 2023; **30**(4): 1-10.

[34] Haque MA, Ahmad S, Abboud AJ, Hossain MA, Kumar K, Haque S, Sonal D, Rahman M, Marisennayya S. **6G wireless Communication Networks: Challenges and Potential Solution**. *International Journal of Business Data Communications and Networking (IJBDCN)* 2024; **19**(1):1-27.