Journal of Asian Scientific Research
ISSN(e): 2223-1331
ISSN(p): 2226-5724
DOI: 10.55493/5003.v16i2.5934
Vol. 16, No. 2, 91-112.

URL: www.aessweb.com

Efficient image cryptosystem using low-dimensional chaos, SHA-256, and random permutation

Joshua C. Dagadu

Department of Information Technology Education, Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development, Kumasi, Ghana.
Email: jcdagadu@aamusted.edu.gh

Article History
Received: 30 October 2025
Revised: 27 January 2026
Accepted: 17 February 2026
Published: 10 March 2026

Keywords
Chaos-based image encryption
Diffusion
Low-dimensional chaos
Permutation
Random permutation
SHA-256.

ABSTRACT

Securing digital images has become increasingly challenging due to the vast volume of images produced and transmitted across various platforms, coupled with the rising incidence of cybersecurity threats. Although attention has shifted toward utilizing hyperchaotic systems for image encryption, many cryptosystems that incorporate high-dimensional chaos along with other computationally intensive techniques face drawbacks, including speed limitations. Therefore, there is a pressing need for encryption schemes that are simple, fast, and sufficiently robust to meet the requirements of lightweight systems. This paper proposes a simple and efficient image encryption algorithm based on chaotic diffusion and random permutation. In this scheme, two low-dimensional chaotic systems are used to generate encryption keys, with one system being influenced by a SHA-256 hash value. The encryption process involves two rounds of chaotic diffusion interleaved with a random permutation to secure the images. The two distinct chaotic keystreams are applied at different stages of the encryption process to enhance randomness. Hashing is incorporated into keystream generation to ensure that the encryption key has a partial dependence on the image being encrypted. Pixel scrambling is performed mid-way by the permutation function to enhance randomness in the process. Experimental analyses, including key, differential, and statistical analyses, demonstrate that the proposed algorithm is fast, robust, and resistant to various types of security attacks on digital images.

Contribution/ Originality: This study proposes a novel image encryption algorithm that operates using a 'diffusion-permutation-diffusion' technique. It employs low-dimensional chaos, random permutation, and SHA-256 to provide state-of-the-art encryption, thereby highlighting the efficiency of low-dimensional chaos when strategically combined with key primitives. This demonstrates that these primitives remain relevant for lightweight image encryption.

1. INTRODUCTION

In recent times, there have been tremendous advancements in technologies for communication, storage, and retrieval of multimedia; not excluding digital images. As such, efficiently securing these images is a matter of great concern. Cryptography is one sure way of ensuring the security of data. The very strong relationship that exists between chaotic dynamical systems and cryptography [1, 2] has paved the way for many chaotic image encryption schemes. Indeed, chaotic systems are endowed with properties such as ergodicity (mixing), unpredictability, pseudo-

randomness, and high responsiveness to slight changes in initial conditions. Due to these properties, they have been considered attractive candidates for image cryptosystems [3, 4]. Traditional data encryption algorithms such as the Advanced Encryption Standard (AES), Rivest-Shamir-Adleman (RSA), and Data Encryption Standard (DES) often suffer from speed limitations when applied to encrypt images, owing to some innate properties of images that make them less suitable for these algorithms [5, 6]. Therefore, the use of chaotic systems offers a promising alternative for efficient image encryption, leveraging their inherent properties to overcome the limitations of traditional methods.

The security of numerous chaotic encryption algorithms is, however, not justified because most of them have small key spaces and require a considerable number of iterations to achieve adequate security. Some also have inappropriate key stream generation functions, which make them easy to break [7-9]. Others apply the chaotic sequences directly from the chaotic maps [10], which affects their performance and makes them fragile [11, 12]. For over a decade now, the pursuit for more robust cryptosystems for digital images has led to the integration of chaos maps and other techniques, not excepting cellular automata [4, 13-17], DNA coding [18-22], more currently neural networks [23-27] and a myriad of other techniques [28-32] for designing image encryption algorithms. Systems that use chaotic systems with high dimensions [31, 33-37] or two rounds of diffusion have also been proposed [38] for a long time now. However, the chaotic systems with high dimensions call for more resources, time, and computational power [39]. Even with low-dimensional chaotic schemes combined with techniques such as DNA, two rounds of diffusion increase execution time, as seen with Wang and Liu [38]. Moreover, the mere merger of chaos with other techniques does not guarantee robustness against some forms of attacks [40]. For instance, merely combining DNA with chaos does not make the encryption schemes resistant to plaintext attacks [40, 41]. Besides, some have fixed encoding and decoding rules [42-44], which still make them easy to break. Using DNA has the main advantage of providing good diffusion and making the algorithm more complex to break if the rules for encoding and decoding, as well as the operations used for diffusion, are randomly and dynamically selected.

Some significant security flaws in most chaotic image cryptosystems are the lack of relationships between plain images and the encryption processes, as well as the use of the same chaotic sequences for encrypting different plain images [12, 41, 45]. The ability to resist differential cryptanalysis and to increase key sensitivity can be enhanced by making the encryption key dependent on the plain image. This dependency ensures that alterations in the plain image result in entirely different encryption keys, thereby improving security [41, 45].

Due to their discreteness, simple structure, sensitivity to initial conditions, unpredictability, and the relatively low arithmetic complexity of low-dimensional chaotic systems, these systems are the preferred choice for encryption schemes, despite some of the deficiencies previously mentioned [39]. To address the associated challenges and provide efficient encryption that has high execution speed, good statistical properties, and robustness against various forms of attacks (at the level of using high-dimensional chaos), combining low-dimensional chaos strategically with grounded primitives could be promising. Hence, this paper presents an encryption scheme that uses two low-dimensional chaotic systems and two primitives (SHA-256 and random permutation).

In this proposed scheme, two low-dimensional chaotic systems namely, the pseudo-randomly enhanced logistic map (PELM) [39] and the Bernoulli shift map are employed to generate encryption keys. One chaotic system is driven by the hash value derived from the SHA-256 function applied to the permuted semi-cipher image. A two-round diffusion technique, combined with random pixel permutation, which introduces variability in the hash value produced, leading to new initial values and key sequences at each encryption instance is utilized to produce the encrypted image. Experimental results demonstrate that the proposed scheme provides adequate resistance to various attack types and maintains efficient execution speed.

The paper aims to propose a simple, fast, and efficient image encryption algorithm suitable for applications requiring lightweight encryption. It demonstrates that low-dimensional chaotic systems, when used in strategically designed encryption algorithms, can be highly effective.

The remainder of the paper is organized as follows: Section 2 provides an overview of the Bernoulli shift map, PELM, SHA-256 hash function, and random permutation. Section 3 describes the proposed encryption process. Section 4 presents the experimentation and discusses the results. Finally, Section 5 offers concluding remarks.

2. PRELIMINARIES

The PELM, Bernoulli shift map, random permutation, and SHA-256 hash function are briefly introduced in this section as the preliminaries.

2.1. Pseudo-Randomly Enhanced Logistic Map (PELM)

The original (direct) logistic map is a simple system that exhibits a transition from one state of order to another state of complete disorder and confusion. It possesses many characteristics required of a pseudorandom-number generator (PRNG) [46]. This map is expressed mathematically as.

$$\alpha_{i+1} = p\alpha_i(1-\alpha_i) \quad (1)$$

In Equation 1, control parameter $p \in (0,4)$, initial value $\alpha_0 \in (0,1)$, i represents the number of iterations. This system is in a state of chaos when its control parameter $p \in (3.57,4)$. Despite the fact that the direct logistic map has witnessed widespread use in the design of chaos-based image encryption algorithms, some drawbacks have been identified with it when applied in cryptography. Non-uniform distributions, chaotic discontinuous ranges, periodicity in chaotic ranges, and small key spaces have been seen in its application [39, 47]. However, this map, similar to other low-dimensional chaotic systems (as previously indicated), possesses certain qualities that make it a preferred option over many high-dimensional chaotic systems. To improve its pseudorandom attributes and enhance security in cryptosystems, it was modified into PELM. This improvement was achieved by including one multiplication in each iteration and applying modulo 1. Compared to the basic logistic map, PELM produces higher and more rapid divergence [39]. PELM is expressed mathematically as

$$\alpha_{i+1} = \Big(\big(p\alpha_i(1-\alpha_i)\big)100000\Big)\, mod\ 1 \quad (2)$$

In Equation 2, *mod* represents modulo 1 operation, $\alpha_0 \in (0,1)$ represents the initial value, $p \in (0,4)$ represents the control parameter, i represents the number of iterations [48]. Figures 1 and 2 are the results of plotting 3000 p values of the original logistic map and the PELM at 10000 iterations. The two figures demonstrate the distribution and the chaotic advantages of the PELM over the direct logistic map.

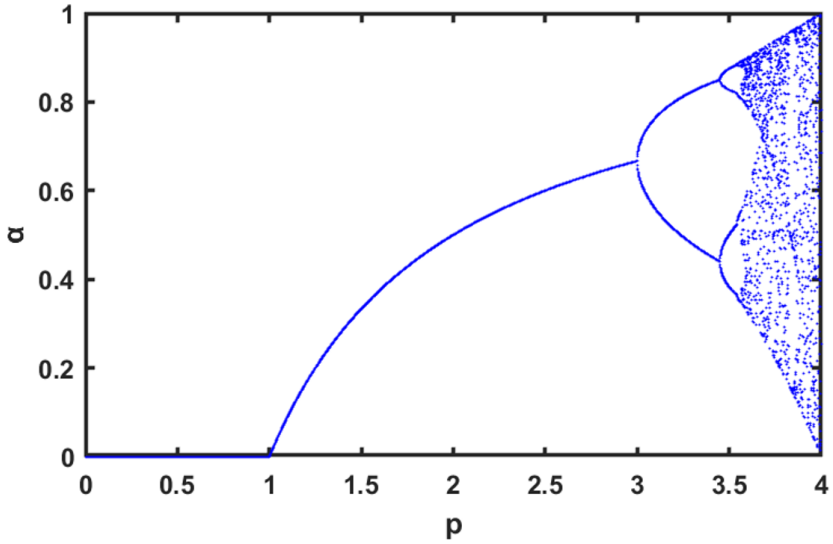

Figure 1. Bifurcation plot of original (direct) Logistic Map.

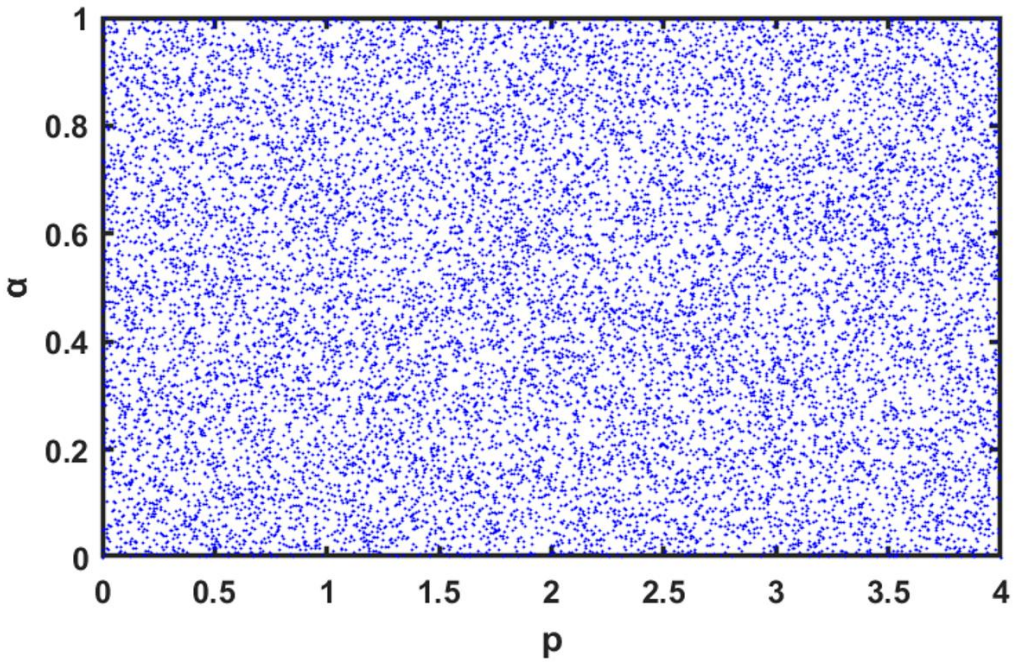

Figure 2. Bifurcation plot of Pseudo-randomly Enhanced Logistic Map (PELM).

2.2. The Bernoulli Shift Map

The two linear functions in Equation 3 represent the Bernoulli Shift map.

$$x_{i+1} = \begin{cases} \mu x_i - q, & if\ x_i \geq 0 \\ \mu x_i + q, & if\ x_i < 0 \end{cases} \quad (3)$$

Equation 3 could be rewritten as

$$x_{i+1} = \mu x_i - q\ sign(x_i) \quad (4)$$

Where x_i represents the initial value, μ represents the control parameter of the random attributes of the chaos map, q represents a scale factor that makes the product μx_i increase or decrease, and then restricts the values produced within the range $[-q,\ q]$.

The map's bifurcation diagram for $q = 1$, plotting 3000 μ values at 10000 iterations are shown in Figure 3.

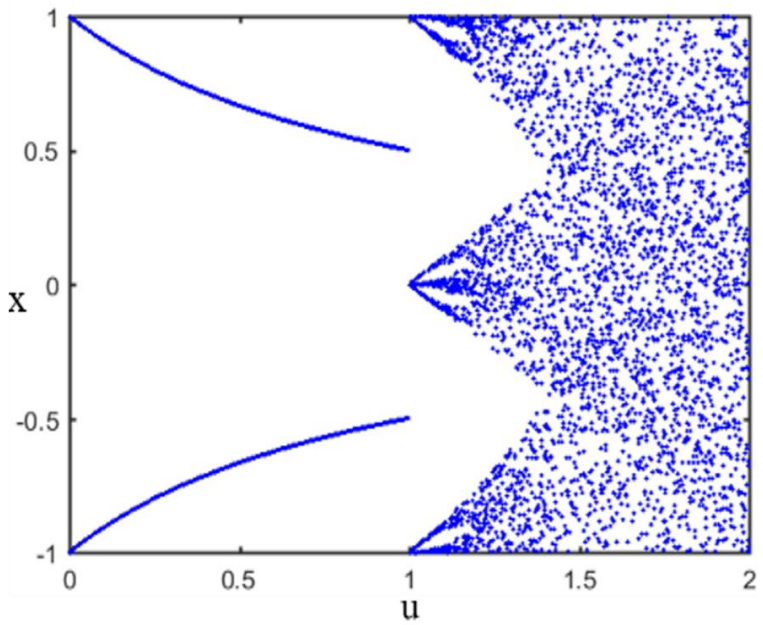

Figure 3. Bifurcation plot of Bernoulli shift map.

From the diagram, it is clear that during states when $\mu \in [0,1)$, an oscillation that occurs between two stable points is produced. At a time $\mu = 1$ instability is produced, and in states where $\mu \in (1,1.4]$, the system's output values are nonuniformly distributed. When μ moves towards 1.4, the distribution improves, and eventually, dispersion at its peak is produced in the output values when $\mu \in [1.4, 2.0)$. The output values spread across the whole range of $[-1, 1]$ for $\mu \in [1.4, 2)$. The chaotic system exhibits instability when $\mu \geq 2$, and for large i values, its output tends to infinity [49, 50].

2.3. Random Permutation

Permutation techniques are necessary components when integrated with random generators for image encryption purposes. An n degree permutation procedure describes the operation of substituting one configuration $\{x_i | i = 1,2,\ldots,m, x_i \in Q\}$ with a different configuration $\{y_i | i = 1,2,\ldots,m, y_i \in Q\}$, mathematically depicted as [51].

$$\varphi = \begin{pmatrix} x_1, x_2 \ldots x_m \\ y_1, y_2, \ldots y_m \end{pmatrix} \quad (5)$$

Where $m!$ permutations of that kind are capable of happening, and any non-empty set is denoted by Q. Equation 6 represents the reverse (i.e., the opposite arrangement) of this permutation process.

$$\varphi^{-1} = \begin{pmatrix} y_1, y_2 \ldots y_m \\ x_1, x_2, \ldots x_m \end{pmatrix} \quad (6)$$

Permutation is an ordered arrangement of elements (with the pairings leaving no remainder) of any Q populated set onto Q. The set containing all of such ordered arrangements is depicted by Q_m with $m!$ members, given that Q has m elements. Premised on this interpretation, we can define a permutation-assisted cryptographic procedure as: Given any data matrix X, If it is converted to a cipher-matrix $\psi_z = \varphi_z(X)$ with φ_z being any permutation operation, then from the inverse operation of ψ_z on it, the original data matrix X can be derived. That is, $\varphi_z^{-1}(\psi_z) = \varphi_z^{-1}(\varphi_z(X)) = X$, since $\varphi_z^{-1}\varphi_z$ forms an identity [51]. Every permutation is a result of transpositions that might be either *odd* or *even*.

In the majority of permutations, not every element is removed from its original position; as such, there are quite a few residual intelligences that attackers can leverage. For this reason, instead of considering every permutation, only certain patterns of permutation that enhance the security level are regarded as valid keys for permutation. Per Prasanna et al. [52], good keys that are used for permutation possess some features that assist in reducing intelligible knowledge. There are three (3) fundamental permutation methods that exist for image matrices. These are *permutations of bits*, *permutations of pixels*, and *permutations of blocks* [48]. Random permutation of pixels is utilized in the proposed algorithm. This method involves arranging the pixels of the original image randomly based on a specific permutation order.

2.4. SHA-256 Hash Function

Hash functions are mathematical functions performed on digital data, primarily used to ensure data integrity. A hash function takes an input of arbitrary length and generates a fixed-length 'fingerprint' string [53]. The SHA-256 is a well-known and extensively used [54-58] cryptographic function. It typically generates a 64-character hexadecimal hash value composed of 256 bits. A single bit change in a message can result in a significant variation in the resulting hash value due to the avalanche effect associated with it. Consequently, when used on images, a one-bit difference in a pixel between two images can lead to entirely different hash values for the images. This security property is exploited in the proposed algorithm to ensure that each encryption key generated for every plain image has a relationship with, or is not independent of, the plain image.

3. METHODOLOGY (PROPOSED ALGORITHM)

The proposed algorithm employs a diffusion-permutation-diffusion technique. It generates a pair of distinct key matrices using two chaotic systems: the PELM and the Bernoulli shift map. These key matrices are utilized in two rounds of diffusion with the plain image matrix and the randomly permuted semi-cipher image matrix through bitwise XOR operations, resulting in the encrypted image. The algorithm accepts the plain image along with a user-provided random 16-character ASCII string as inputs. This ASCII string is used to generate the initial value and control parameters for the PELM. The PELM generates the first key matrix, which is used in the initial diffusion round with the plain image matrix to produce a semi-cipher image matrix. This semi-cipher image matrix is then randomly permuted using a permutation order to create a scrambled semi-cipher image matrix. A SHA-256 hash is computed on the scrambled semi-cipher image matrix to generate a 64-character hexadecimal value, which is used to derive the initial value and control parameters for the Bernoulli shift map. The second key matrix, generated by the Bernoulli shift map, is used in the second diffusion round with the scrambled semi-cipher image matrix, producing the fully encrypted image. During decryption, the process is reversed to recover the original plain image. The decryption process involves inputting the encrypted image, along with the permutation order, the ASCII string, and the SHA-256 hash value as the decryption key set.

The role of random permutation is to introduce variability in the second key generation and in the second round of diffusion. Whenever the random permutation function is performed, there is a high probability of a different permutation order. This causes changes in the image matrix. Since SHA-256 produces a completely different hash value upon a slight change in the message, it is evident that a new hash value will be generated each time the image matrix is altered via random permutation. Therefore, even with the same initial 16-character ASCII seed and the same input plain image, each encryption instance is likely to produce a different hash value. Consequently, this results in different initial values and control parameters for the Bernoulli shift map. Moreover, since the Bernoulli shift map, like other chaotic systems, exhibits high sensitivity to minor variations in initial conditions, a different chaotic sequence and key are generated at every encryption. This explains why the permutation order and hash value at each encryption instance are essential keys for accurately decrypting the cipher image.

The process of generating chaotic keys, as well as the steps involved in image encryption and decryption, are detailed in the following subsections.

3.1. Generation of First Key

Step 1: Input an arbitrary 16-character ASCII string A and plain image dimensions M and N.

$$A = \{a_1, a_2, a_3, \dots, a_{16}\} \quad (7)$$

Step 2: Convert the first four (4) characters $\{a_1, a_2, a_3, a_4\} \in A$ into their hexadecimal forms to get 8 digits $H = \{h_1, h_2, h_3, \dots, h_8\}$ which are then used to produce a value α_1 as

$$\alpha_1 = (\sum_{i=1}^{8}(h_i)_{10})/128 \quad (8)$$

Step 3: Convert the 5th to the 8th characters of A into their binary forms to get 32 bits $B = \{b_1, b_2, b_3, \dots, b_{32}\}$ used to derive another value α_2 as

$$\alpha_2 = (\sum_{i=1}^{32}(b_i \times 2^i))/2^{32} \quad (9)$$

Step 4: Add the two values α_1 and α_2 to obtain the initial condition α_0 of the PELM as

$$\alpha_0 = (\alpha_1 + \alpha_2) mod\ 1 \quad (10)$$

Step 5: Convert each of the last eight (8) characters of A into their binary forms to obtain 64 bits $D = \{d_1, d_2, d_3, \dots, d_{64}\}$ which are put into two (2) blocks of 32 and used to derive two values α_3 and α_4 using Equation 11.

$$\alpha_n = (\sum_{i=1}^{32}(d_i \times 2^i))/2^{32} + 1 \quad (11)$$

Where the value $n \in \{3,4\}$.

Step 6: Using Equation 12, derive the control parameter of PELM.

$$p = 3.999 + \left((\alpha_3 + \alpha_4) mod 1\right) \times 0.01 \quad (12)$$

Step 7: Iterate Equation 2 NM times using the initial value α_0 and control parameter p to obtain the chaotic sequence $S = \{s_1, s_2, s_3, \dots, s_{MN}\}$ which is then converted into the integer sequence using Equation 13 to produce an image matrix of the first key $T = \{t_1, t_2, t_3, \dots, t_{NM}\}$ as.

$$t_i = \lfloor (s_i \times 10^{14}) \rfloor mod\ 256 \quad (13)$$

Where t_i is a pixel and $t_i \in T$. Equation 13 scales each element of the chaotic sequence to represent an image pixel, which is then floored to eliminate fractional parts. The modulo operation restricts the pixel value between 0 and 255.

3.2. Generation of Second Key

Step 1: Obtain the SHA-256 hash value R (i.e., 64-character hexadecimal string) of the permuted semi-cipher image, and image dimensions M and N.

$$R = \{r_1, r_2, r_3, \dots, r_{64}\} \quad (14)$$

Step 2: Convert the first 32 characters $\{r_1, r_2, r_3, \dots, r_{32}\} \in R$ into their binary forms to derive a stream of 128 bits.

$$\beta = \{b_1, b_2, b_3, \dots, b_{128}\} \quad (15)$$

Step 3: Divide β into four (4) separate 32-bit blocks, and preprocess each block to obtain four values $\{\beta_1, \beta_2, \beta_3, \beta_4\}$ using Equation 16, where $i \in \{1,2,3,4\}$ and s is a bit in a block β_i.

$$\beta_i = \sum_{s=1}^{32} (b_s \times 2^s) \quad (16)$$

Step 4: Use the four values to obtain x_0 which is the initial value of the Bernoulli shift map using Equation 17.

$$x_0 = \left(\left(((\beta_1 \oplus \beta_2) \oplus \beta_3) \oplus \beta_4\right) mod\ 256\right) / 255 \quad (17)$$

Step 5: Convert the last 32 characters $\{r_{33}, r_{34}, r_{35}, \dots, r_{64}\} \in R$ into their binary forms to obtain a stream of 128 bits.

$$B = \{s_1, s_2, s_3, \dots, s_{128}\} \quad (18)$$

Step 6: Process the bits in B to derive β_5 using Equation 19, which is then used to derive the control parameter μ, of the Bernoulli shift map using Equation 20 as follows:

$$\beta_5 = \left(\sum_{i=1}^{128} (s_i \times 2^i)\right) / 2^{128} \quad (19)$$

$$\mu = 0.3001 + (\beta_5\ mod\ 2) \quad (20)$$

Step 7: Using x_0 (initial value) and μ (control parameter), Equation 3 is iterated MN times to produce a chaotic sequence $X = \{x_1, x_2, x_3, \dots, x_{MN}\}$ which is then converted into a sequence of integers in order to obtain the second key matrix $\Upsilon = \{\upsilon_1, \upsilon_2, \upsilon_3, \dots, \upsilon_{MN}\}$ using Equation 21, where a pixel $\upsilon_i \in \Upsilon$.

$$\upsilon_i = \lfloor (x_i \times 10^{14}) \rfloor mod\ 256 \quad (21)$$

3.3. Encryption Process

Step 1: Matrix I (of the plain image), and a 16-character ASCII string A are inputted; image dimensions M and N are obtained.

Step 2: Obtain T (key 1) as in section 3.1 using A, M and N.

Step 3: A bitwise XOR is operated on the plain image matrix and key 1 to get a semi-cipher image matrix I'.

$$I' = I \oplus T \quad (22)$$

Step 4: Obtain a random permutation order $\theta = \{o_1, o_2, o_3, \dots, o_{MN}\}$ using a random permutation function f.

$$\theta = f(N \times M) \quad (23)$$

Step 5: Using order θ, permute matrix I' randomly to obtain a scrambled semi-cipher image matrix Q' as

$$Q' = reshape(I'(\theta), M, N) \quad (24)$$

Step 6: Perform a SHA-256 hash operation on Q' to get a hexadecimal string R having 64 characters, which is then used to generate the key 2 matrix Υ using the steps in section 3.2.

Step 7: Operate a bitwise XOR function on Q' and Υ to produce the fully encrypted image Q as

$$Q = Q' \oplus \Upsilon \tag{25}$$

The encryption process is illustrated in Figure 4.

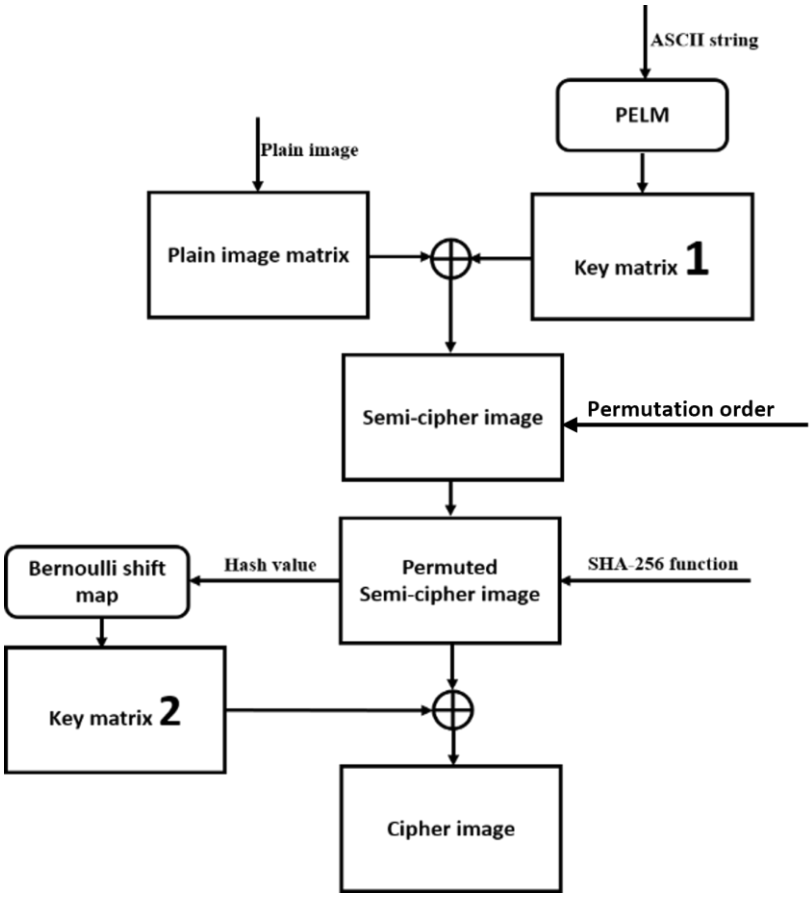

Figure 4. A block diagram showing the process of encryption.

3.4. Decryption Process

Step 1: Input the cipher image matrix Q, the ASCII string A made up of 16-characters, the 64-character SHA-256 hash value string R, the permutation order θ and obtain dimensions M and N of the input image.

Step 2: Obtain the matrices T and Υ of key 1 and key 2 as in sections 3.1 and 3.2.

Step 3: Engage key 2 and Q in a bitwise XOR function to regain the scrambled semi-decrypted image Q'.

$$Q' = Q \oplus \Upsilon \tag{26}$$

Step 4: Inversely permute Q' to regain the semi-decrypted image I', which is unscrambled.

$$I' = reshape(Q'(\theta'), M, N) \tag{27}$$

Step 5: Carry out a bitwise XOR operation on I' and key 1 matrix T to regain the fully decrypted (plain) image I.

$$I = I' \oplus T \tag{28}$$

Figure 5 is a block diagram showing the procedure for decryption.

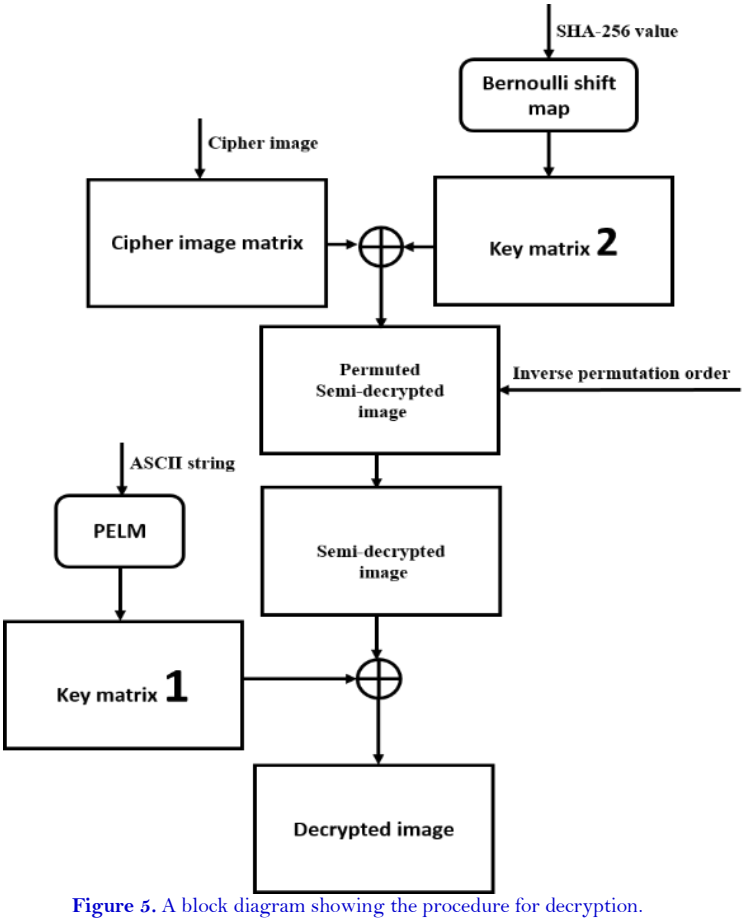

Figure 5. A block diagram showing the procedure for decryption.

4. EXPERIMENT AND RESULTS ANALYSIS

4.1. Experimental Configuration

Several grayscale images are used in experimenting with the proposed algorithm. The images are of dimensions 1024×1024, 512×512, and 256×256. Images were obtained from USC SIPI image database (http://sipi.usc.edu/database) and processed into different sizes where necessary. The experiment is conducted on a personal computer equipped with the following specifications: an Intel(R) Core (TM) i5 processor running at 2.6 GHz and 4 GB of RAM. An ASCII string, 'aS4x5yu5H699fdgb', is used as the seed key for encrypting all images. The results obtained from statistical analysis, differential analysis, key analysis, and robustness testing against specific security attacks are discussed. These findings are also compared with several other well-known published algorithms.

4.2. Histogram Analysis

Displayed in Figure 6 are the histogram plots of the original plain images of the baboon, boat, and peppers. It can be observed from the histograms that there are patterns in the plain images that need to be concealed in their encrypted versions to make it difficult for attackers to extract intelligible information. This is achieved by evenly distributing pixels in the encrypted versions of these images. As demonstrated clearly in Figure 7, the proposed algorithm effectively produces these results, as evidenced by the uniform histograms of the respective cipher images. Moreover, it is apparent from the figure that the cipher images are highly noisy, which demonstrates the effectiveness of the proposed algorithm's encryption process. Consequently, attackers are unable to obtain intelligible information by analyzing the histograms of the cipher images; thus, the proposed algorithm demonstrates robustness against statistical attacks.

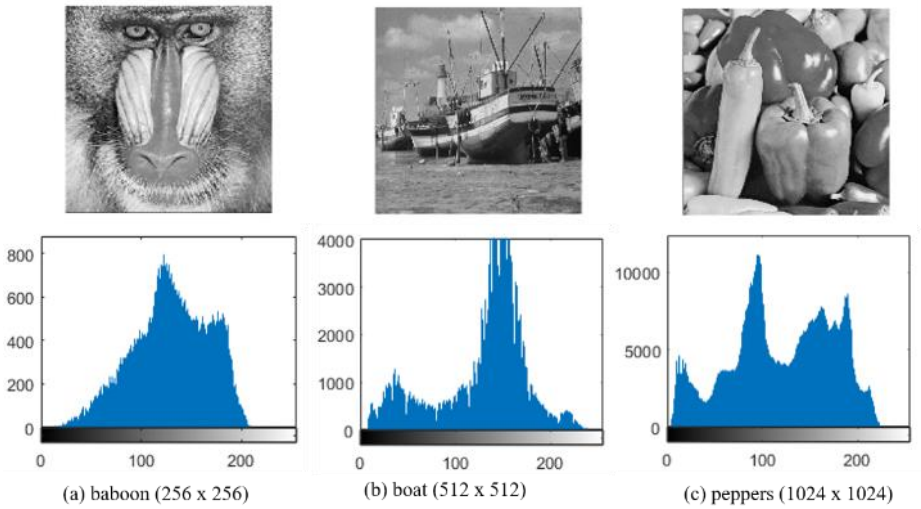

Figure 6. Histogram plots of original plain images.

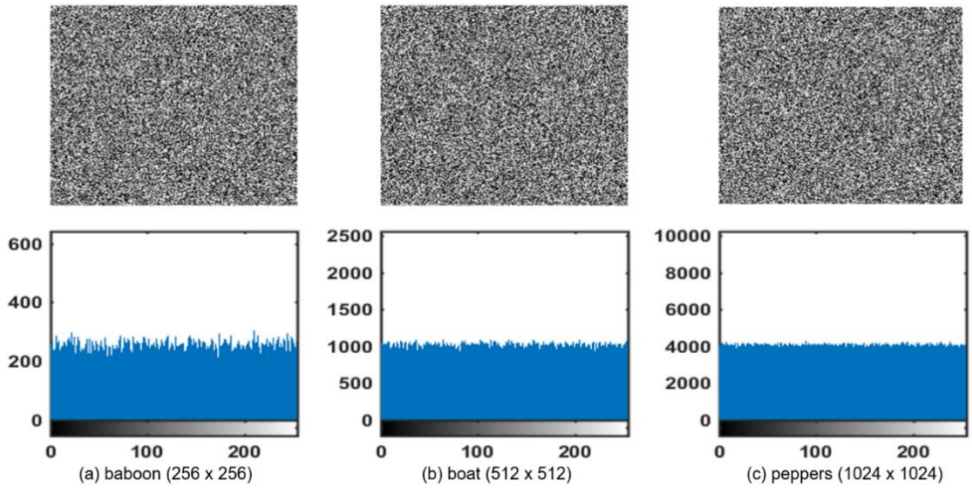

Figure 7. Histogram plots of cipher images.

4.3. Analysis of Correlation

To enhance resistance against statistical attacks, the high correlation coefficients among neighboring pixels in plain images prior to encryption should be significantly reduced or entirely eliminated in their encrypted versions. Correlation coefficients between neighboring pixels are computed using Equations 29, 30, 31, and 32 [59].

$$M(a) = \frac{1}{N}\sum_{i=1}^{N} a_i \tag{29}$$

$$V(a) = \frac{1}{N}\sum_{i=1}^{N}\left(a_i - M(a)\right)^2 \tag{30}$$

$$CV(a,b) = \frac{1}{N}\sum_{i=1}^{N}\left(a_i - M(a)\right)\left(b_i - M(b)\right) \tag{31}$$

$$r_{ab} = \frac{CV(a,b)}{\sqrt{V(a)\times\sqrt{V(b)}}} \tag{32}$$

Where the gray scale values of two neighboring pixels are represented by a and b, variance is represented by $V(a)$, covariance is denoted by $CV(a,b)$, and the mean is denoted by $M(a)$.

Two thousand pairs of adjacent pixels are selected from both plain and cipher images, and their correlations in the vertical (V), diagonal (D), and horizontal (H) directions are computed. Figure 8 displays plots of the correlation between the plain baboon image and its corresponding cipher image, both with dimensions 256×256. Additionally,

the correlation coefficients for all test images are listed in Table 1, compared with other previously proposed algorithms. From the values shown in the table and the plot, it is evident that the proposed algorithm effectively reduces the high correlation among neighboring pixels in the cipher images. As observed, the correlations in the plain images prior to encryption are close to 1, while those in the corresponding cipher images are near 0. This demonstrates that the proposed algorithm can resist statistical attacks, and its efficiency is comparable to modern schemes that utilize high-dimensional chaos and other techniques. Although the algorithm employs low-dimensional chaos, its two rounds of diffusion, interleaved with random permutation, ensure the effective spreading of each pixel's information across multiple pixels in the cipher image. This process also induces significant confusion and permutation by highly randomizing pixel values, thereby breaking the correlation among adjacent pixels.

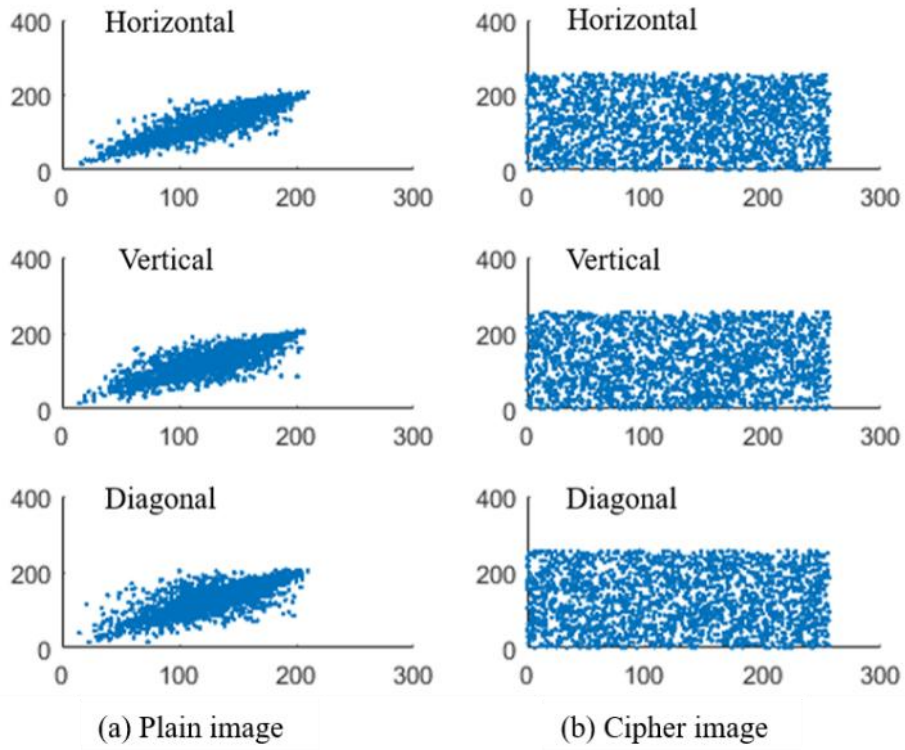

Figure 8. Correlation plots of plain and encrypted baboon images of dimension (256×256).

Table 1. Results of correlation analysis.

Image	Dimension	Direction	Plain image	Cipher images			
				Proposed	Li, et al. [10]	Wang and Liu [38]	Zhan, et al. [60]
Baboon	256×256	H	0.87367	0.00700	0.00012	0.00239	0.00296
		V	0.82610	-0.00125	-0.19528	-0.00262	0.00184
		D	0.78433	-0.00287	-0.00074	0.00016	-0.00250
	512×512	H	0.86654	-0.00136	-0.00218	-0.00166	0.00097
		V	0.75873	0.00049	-0.18652	-0.00151	0.00143
		D	0.72618	-0.00344	-0.00179	-0.00151	-0.00149
	1024×1024	H	0.96817	-0.00059	0.00074	-0.00034	0.00015
		V	0.93885	-0.00050	-0.25845	0.00183	-0.00084
		D	0.91541	-0.00094	0.00039	-8.87419	-0.00094
Boat	256×256	H	0.92684	-0.00298	0.00391	-0.00397	0.00638
		V	0.94518	0.00498	-0.26236	0.00151	-0.00149
		D	0.88334	-0.00137	-0.00133	0.00495	0.00187
	512×512	H	0.93811	0.00065	-0.00091	-0.00153	-0.00218
		V	0.97131	-0.00057	-0.28033	-0.00237	0.00088

Image	Dimension	Direction	Plain image	Cipher images			
				Proposed	Li, et al. [10]	Wang and Liu [38]	Zhan, et al. [60]
		D	0.92216	0.00016	0.00304	-0.00128	0.00057
	1024×1024	H	0.98450	0.00014	0.00028	0.00077	-0.00018
		V	0.99320	-0.00103	-0.30791	0.00025	0.00084
		D	0.97820	0.00012	0.00096	0.00042	-0.00039
Peppers	256×256	H	0.96347	-0.00498	0.00477	0.00255	-0.00161
		V	0.97051	0.00051	-0.29483	-0.00037	-0.00976
		D	0.93652	-0.00063	-0.00556	-0.00189	-0.00304
	512×512	H	0.97677	-0.00211	0.00024	0.00079	-0.00100
		V	0.97920	-0.00059	-0.29697	-0.00063	0.00118
		D	0.96393	0.00094	0.00163	-0.00413	0.00148
	1024×1024	H	0.99453	-0.00096	6.09647	0.00020	-0.00041
		V	0.99514	0.00034	-0.31726	0.00158	-0.00054
		D	0.99000	-0.00045	6.02935	-0.00061	-0.00039

4.4. Information Entropy

This is a mathematical attribute that indicates unpredictability and irregularity or randomness in the source of information [61]. It is expressed by Equation 33.

$$\mathrm{E}(s) = \sum_{i=0}^{2^N-1} p(s_i) log \frac{1}{p(s_i)} \qquad (33)$$

In the equation, N represents the overall number of symbols $s_i \in s$; the probability of a symbol s_i occurring is represented by $p(s_i)$, *log* stands for the base 2 logarithm. Given a message of length 8 bits, it is required that for a good cipher algorithm, $\mathrm{E}(s)$ moves close to 8. In cryptosystems, when the image's entropy value is closer to the ideal condition (i.e., 8), the probability of attackers obtaining meaningful information from it decreases. Information entropies of plain and encrypted images, using the proposed algorithm, in comparison with other earlier proposed algorithms, are displayed in Table 2. It is evident from the displayed values that the entropies of encrypted images are close to the ideal value; thus, the algorithm demonstrates robustness against entropy attacks. The favorable entropy values are attributed to the randomization of pixel values in the cipher image, which results from the random permutation function carried out after the first round of key application. Additionally, the application of both the first and second chaotic key matrices, which are highly randomized and unpredictable in bitwise XOR operations, enhances high diffusion and confusion in the cipher image.

Table 2. Results of entropy analysis.

Image	Dimension	Plain image	Cipher Images			
			Proposed	Li, et al. [10]	Wang and Liu [38]	Zhan, et al. [60]
Baboon	256×256	7.22919	7.99748	7.99675	7.99755	7.99715
	512×512	7.35830	7.99940	7.99925	7.99933	7.99932
	1024×1024	7.32720	7.99982	7.99983	7.99980	7.99979
Boat	256×256	7.15866	7.99723	7.99744	7.99726	7.99663
	512×512	7.19137	7.99937	7.99921	7.99924	7.99889
	1024×1024	7.19785	7.99980	7.99983	7.99983	7.99946
Peppers	256×256	7.57703	7.99668	7.99743	7.99691	7.99692
	512×512	7.59365	7.99924	7.99924	7.99923	7.99914
	1024×1024	7.58854	7.99981	7.99980	7.99980	7.99965

4.5. Analysis of Differential Attacks

Two different metrics are used to measure the capacity of image encryption schemes to resist differential cryptanalysis. These are NPCR (i.e., Number of Pixel Change Rate) and UACI (i.e., Unified Average Changing Intensity). These metrics are mathematically expressed as:

$$NPCR = \frac{1}{W \times H}\left(\sum_{i,j} D(i,j)\right) \times 100\% \quad (34)$$

and

$$UACI = \frac{1}{W \times H}\left(\sum_{i,j} \left|\frac{C_1(i,j) - C_2(i,j)}{255}\right|\right) \times 100\% \quad (35)$$

In the expressions, C_1 and C_2 represent two cipher images that have a difference of just one pixel change in their corresponding plain images. The pixel values are denoted by $C_2(i,j)$, and W and H represent the image dimensions.

Attackers typically make tiny changes in a plain image to obtain two variants of the same image with a slight difference. They then encrypt the image variants with the cryptosystem to discover the connection between their cipher versions. One-pixel changes are made to plain images; both the original and their corresponding altered images are encrypted with the same key to carry out this test. In Table 3, the results of these tests alongside those of other schemes are presented. The values in the table show that changing just one pixel value in an image leads to high values of NPCR and UACI.

This demonstrates that the proposed algorithm is sufficiently strong to resist differential attacks and competes favorably with other algorithms. The good NPCR and UACI values that the proposed algorithm produces are attributed to the key's partial dependence on the plain image via the SHA-256 value obtained from the randomly permuted semi-cipher image, which is used to generate the initial condition and control parameter of the chaotic system (Bernoulli Shift Map), used to derive the second key matrix. This is also evident from the low values obtained by Li et al. [10], whose key does not depend on the plain image.

Table 3. Differential analysis.

Image	Dimension	Metric	Proposed	Li, et al. [10]	Wang and Liu [38]	Zhan, et al. [60]
Baboon	256×256	NPCR	0.99591	1.52587e-5	0.99597	0.94224
		UACI	0.33209	5.98383e-8	0.33534	0.47230
	512×512	NPCR	0.99618	3.81469e-6	0.99627	0.32831
		UACI	0.33537	1.49595e-8	0.33502	0.08235
	1024×1024	NPCR	0.99624	9.53674e-7	0.99603	0.94139
		UACI	0.33468	3.73989e-9	0.33413	0.39016
Boat	256×256	NPCR	0.99589	1.52587e-5	0.99578	0.94224
		UACI	0.33501	5.98383e-8	0.33394	0.47304
	512×512	NPCR	0.99609	3.81469e-6	0.99613	0.94166
		UACI	0.33470	1.49595e-8	0.33415	0.45676
	1024×1024	NPCR	0.99609	9.53674e-7	0.99610	0.44529
		UACI	0.33476	3.73989e-9	0.33469	0.19306
Peppers	256×256	NPCR	0.99603	1.52587e-5	0.99610	0.94224
		UACI	0.33397	5.98383e-8	0.33461	0.47206
	512×512	NPCR	0.99595	3.81469e-6	0.99625	0.94166
		UACI	0.33441	1.49595e-8	0.33392	0.23631
	1024×1024	NPCR	0.99609	9.53674e-7	0.99606	0.94139
		UACI	0.33462	3.73989e-9	0.33464	0.39012

4.6. Key Space

A good cryptosystem is expected to possess a sufficiently large key space to provide strong resistance against exhaustive key search attacks. In the proposed algorithm, the ASCII string of 16 characters constitutes 128 bits, the 64-hexadecimal character SHA-256 hash value comprises 256 bits, and the random permutation order together form the key set. If the security of SHA-256 hash function to resist the best-known hacking attempt is 2^{128} [12] then with the 64-bit double precision number of about 10^{-15} (according to the IEEE floating-point standard [62])with an assumed precision of 10^{-16}, the proposed algorithm has a key space larger than 2^{256}, rendering it sufficient enough to provide resistance to brute-force attacks.

4.7. Key Sensitivity Analysis

Key sensitivity ensures that attackers cannot succeed when making partial guesses of the decryption key. When incorrect keys are used in decryption attempts, the wrongly decrypted image should not reveal any pattern of information. With the proposed scheme, small pixel changes in images lead to entirely different hash values used in generating initial values and control variables of the chaotic system employed to produce the second keystream. Therefore, if a slight change is made to the seed ASCII string used to derive the initial value and control parameter of the PELM that generates key 1, a ripple effect occurs throughout the entire key set, resulting in incorrect decryption of the cipher image. Figure 9 displays images and histograms when part of the decryption key (ASCII string) is slightly altered from 'aS4x5yu5H699fdgb' to 'AS4x5yu5H699fdgb'. It is evident from the figure that even if an incorrect key though very close to the correct key is used to decrypt an encrypted image, no visible pattern of information will be revealed in the resulting image.

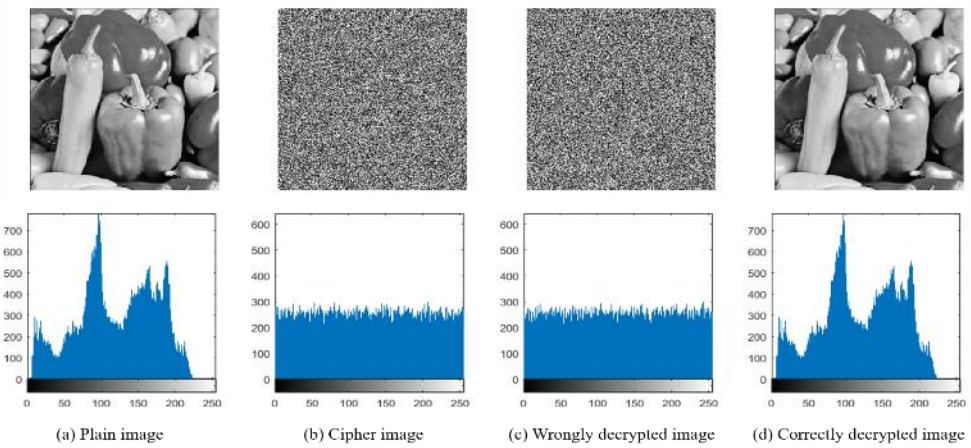

Figure 9. Histogram plots of key sensitivity test.

4.8. Robustness Against Plaintext Attacks

Some image encryption schemes have been successfully broken using known-plaintext and chosen-plaintext attacks. In the algorithm proposed herein, the initial value and control parameter of the chaos map used to derive one of the encryption keys are obtained from a hash value generated when the SHA-256 function is applied to the randomly permuted semi-cipher image. As a result, the encryption key is directly dependent on the image being encrypted. Consequently, each time a new image is encrypted, a completely different initial value and control parameter are obtained, leading to an entirely different encryption key. Even with the same image, the hash value is highly likely to change with each execution due to the random permutation function. Therefore, the encryption key heavily depends on the plain image matrix, which enhances robustness against known-plaintext and chosen-plaintext attacks. To demonstrate this, two plain images (all-white and all-black) are used to evaluate the algorithm's robustness against these attacks. Figure 10 shows the results of the test, while Table 4 provides entropy and

correlation coefficient values of the assessment. It is evident from the figure and the table that no meaningful patterns are present in the encrypted images. They exhibit evenly distributed histograms, entropy values close to 8 (the ideal value), and all correlation coefficients near 0. Therefore, it can be concluded that the algorithm demonstrates strong resistance against known-plaintext and chosen-plaintext attacks.

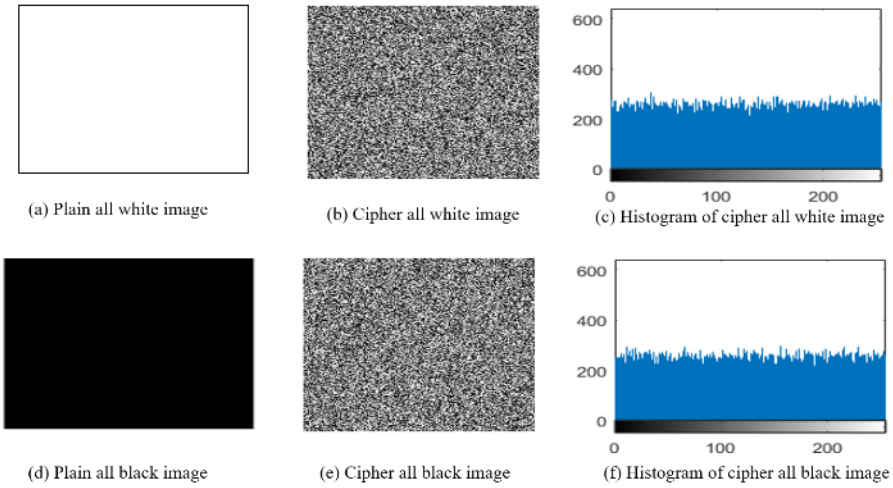

Figure 10. Histogram analysis of robustness against plaintext attacks.

Table 4. Entropies and correlation coefficients of all-white and all-black images.

Test image	Information entropy	Correlation coefficients		
		Horizontal	Vertical	Diagonal
Plain all-white	0	-	-	-
Cipher all-white	7.99702	-0.00246	-0.00844	0.00255
Plain all-black	0	-	-	-
Cipher all-black	7.99762	-0.00253	0.00041	-0.00050

4.9. Robustness Against Noise Attacks

Sometimes, cipher images may be affected by various forms of noise during transmission. This adversely impacts the quality when these images are decrypted. Robust encryption schemes should be capable of decrypting and recovering images to satisfactory levels even when the cipher images are affected by different types and degrees of noise during transmission. The proposed algorithm is tested to evaluate its strength against Gaussian noise and salt-and-pepper noise at various intensities, using the boat image with dimensions 256×256. To assess this, Gaussian noise at different variances is introduced to the cipher image before decryption with the correct keys. Figure 11 shows the results.

Figure 12 gives a Peak Signal-to-Noise Ratio (PSNR) performance comparison between the proposed algorithm and other earlier algorithms in resisting Gaussian noise attacks. Clearly, the proposed scheme outperforms Zhan et al. [60] and Wang and Liu [38] but is comparable with Li et al. [10].

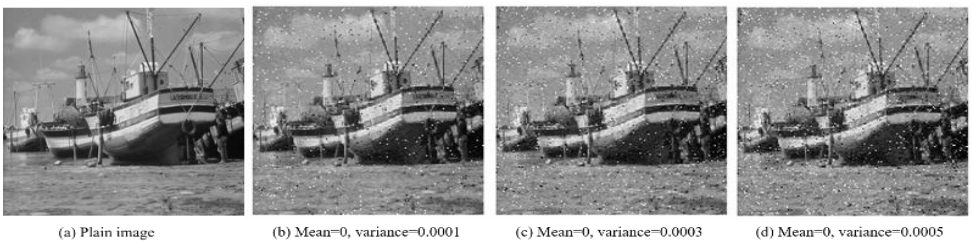

Figure 11. Decrypted images after Gaussian noise attacks on cipher images.

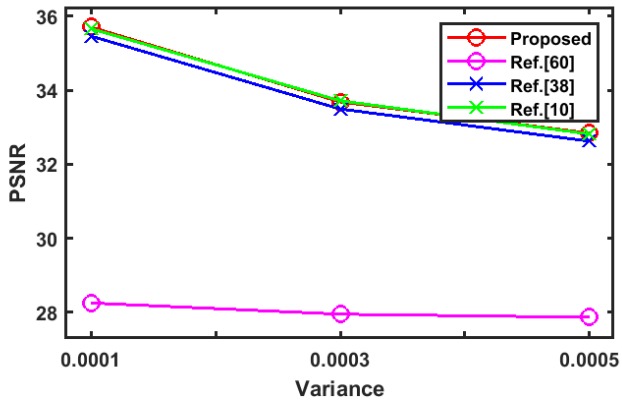

Figure 12. Resistance to Gaussian noise attacks.

Further, salt and pepper noise at different densities is also introduced to the cipher image before decryption using the correct keys. Figure 13 shows the results.

Figure 14 gives a PSNR performance comparison between the proposed algorithm and other earlier algorithms in resisting salt and pepper noise attacks. Clearly, the proposed scheme performs far better than Zhan et al. [60] and is comparable with Wang and Liu [38] and Li et al. [10]. The new algorithm recovers well from noise attacks due to its design, which introduces high redundancy in cipher images through layers of diffusion interleaved with random permutation. This structural design ensures that noise affecting portions of cipher images is evenly distributed across pixels during decryption, resulting in low-concentrated intensities in specific areas of the decrypted images.

Figure 13. Decrypted images after salt and pepper noise attacks on cipher images.

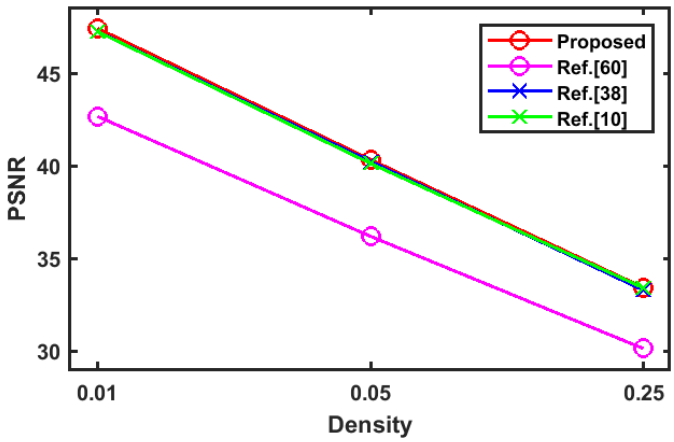

Figure 14. Resistance to salt and pepper noise attacks.

4.10. Robustness Against Cropping Attacks

During transmission, data loss can occur. As a result, it is possible that the content of images transmitted across communication networks might be lost due to congestion or malicious destruction. Algorithms for image encryption should be sufficiently robust to ensure the correct retrieval of images when decrypted, even after data loss in the cipher image during transmission. Portions of cipher images are cropped before decryption to test the proposed scheme's ability to recover the original image effectively. Figure 15 displays the results when the cipher images suffer data losses due to cropping at different degrees.

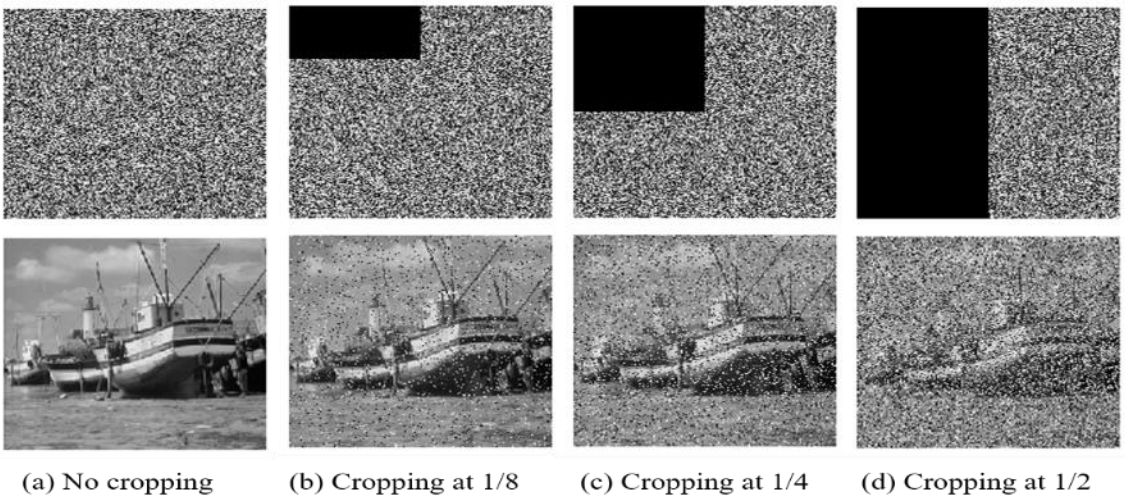

Figure 15. Decrypted images after cropping attacks on cipher images.

Figure 16 gives a PSNR performance comparison between the proposed algorithm and other algorithms in recovering images after cropping attacks. Clearly, the proposed algorithm outperforms those of Zhan et al. [60], Wang and Liu [38], and Li et al. [10]. This quality is also due to the proposed algorithm's design, which ensures high redundancy by spreading pixel values across the entire cipher image through permutation and a second-round chaotic diffusion after permutation. Consequently, data loss resulting from cropping portions of cipher images is evenly distributed throughout the image during decryption, making the recovered image closely resemble the original image.

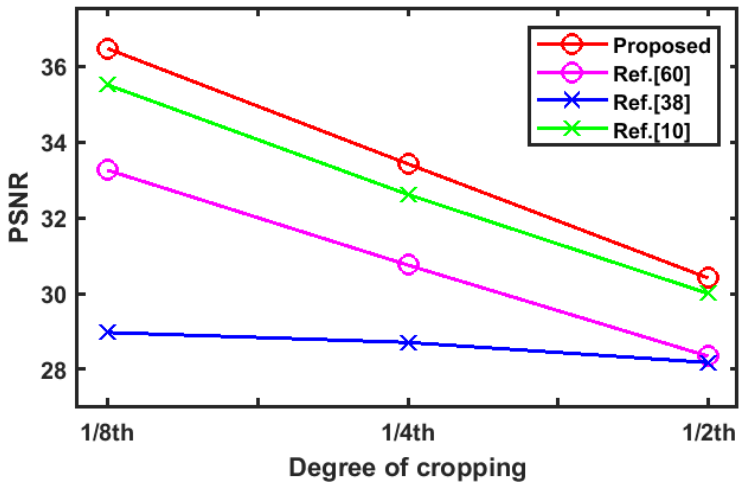

Figure 16. Resistance to cropping attacks.

4.11. Encryption Speed

Several grayscale images were encrypted using the proposed algorithm. The average encryption time on the given experimental setup is 0.587 seconds for an image with dimensions 256×256, 2.212 seconds for an image with dimensions 512×512, and 9.031 seconds for an image with dimensions 1024×1024. A comparison of the speed performance between the proposed algorithm and that of other schemes using a boat image is presented in Figure 17.

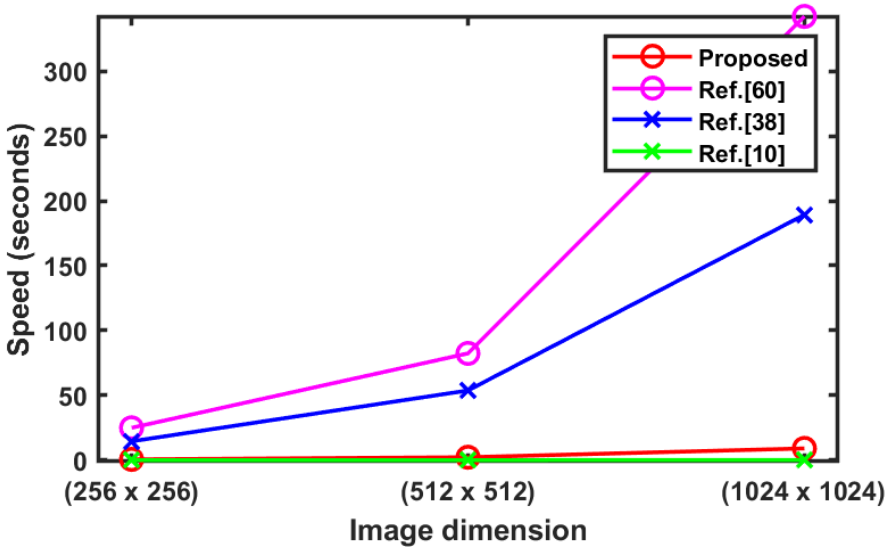

Figure 17. Speed performance.

From the figure, it is clear that the proposed scheme performs better than all others except [10], which is slightly faster. This is evident considering the fact that both algorithms utilize low-dimensional chaos, but the proposed algorithm employs two such systems and two rounds of chaotic diffusion, whereas Li et al. [10] do not. Additionally, Wang and Liu [38] and Zhan et al. [60] require more time to encrypt because the former combines chaos with DNA coding, while the latter employs high-dimensional chaos with DNA. Algorithms that incorporate DNA coding and high-dimensional chaotic operations tend to be more computationally intensive, especially when applied to larger image sizes, compared to algorithms that utilize low-dimensional chaos.

5. CONCLUSION

An efficient image encryption algorithm leveraging low-dimensional chaos, random permutation, and SHA-256 is proposed in this paper. The algorithm employs a 'diffusion-permutation-diffusion' technique. Two chaotic systems are utilized for keystream generation, along with a strategic combination of random permutation and SHA-256 to ensure high randomness and keystream dependence on the image. Through two rounds of bitwise XOR operations interleaved with random permutation, a robust cipher is produced.

The results of extensive experimentation conducted with images of various dimensions demonstrate the efficiency of the algorithm. Furthermore, comparative analyses between the proposed scheme and other algorithms highlight the algorithm's competitiveness. Experimental results indicate that low-dimensional chaos, when combined with other primitives within well-designed encryption structures, can produce outcomes such as high randomness and strong resistance to common attacks. These results are comparable to those achieved by high-dimensional chaos-based systems, while offering improved encryption speed. High-dimensional chaos algorithms and techniques like DNA coding can be computationally intensive, often requiring more time to encrypt larger images. Algorithms lacking structures that ensure keys are partly dependent on the plaintext may be weaker against differential attacks. Based on the experimental findings, it can be concluded that the proposed algorithm is simple, fast, and sufficiently

secure against common attacks. Therefore, it is suitable for use in various systems requiring rapid protection. However, the algorithm was tested only with grayscale images and in an offline environment. Future work includes extending the algorithm to color images, implementing it in real-time streaming environments, and exploring strategies to enhance overall performance.

Funding: This study received no specific financial support.
Institutional Review Board Statement: Not Applicable.
Transparency: The author states that the manuscript is honest, truthful, and transparent. No key aspects of the investigation have been omitted. This study followed all writing ethics.
Competing Interests: The author declares that there are no conflicts of interests regarding the publication of this paper.

REFERENCES

[1] Y. Mao and G. Chen, *Chaos-based image encryption. In E. Bayro Corrochano (Ed.), Handbook of Geometric Computing: Applications in Pattern Recognition, Computer Vision, Neuralcomputing, and Robotics* Berlin, Heidelberg, Germany: Springer-Verlag, 2005.

[2] P. R. Sankpal and P. A. Vijaya, "Image encryption using chaotic maps: A survey," presented at the 2014 Fifth International Conference on Signal and Image Processing (ICSIP). Piscataway, NJ: IEEE, 2014.

[3] X. Yan, Q. Hu, and L. Teng, "A novel color image encryption method based on new three-dimensional chaotic mapping and DNA coding," *Nonlinear Dynamics*, vol. 113, no. 2, pp. 1799-1826, 2025. https://doi.org/10.1007/s11071-024-10277-8

[4] Q. Lai and Y. Liu, "A family of image encryption schemes based on hyperchaotic system and cellular automata neighborhood," *Science China Technological Sciences*, vol. 68, no. 3, p. 1320401, 2025. https://doi.org/10.1007/s11431-024-2678-7

[5] D. Ravichandran, P. Praveenkumar, J. B. B. Rayappan, and R. Amirtharajan, "Chaos based crossover and mutation for securing DICOM image," *Computers in Biology and Medicine*, vol. 72, pp. 170-184, 2016. https://doi.org/10.1016/j.compbiomed.2016.03.020

[6] M. Li, M. Wang, H. Fan, K. An, and G. Liu, "A novel plaintext-related chaotic image encryption scheme with no additional plaintext information," *Chaos, Solitons & Fractals*, vol. 158, p. 111989, 2022. https://doi.org/10.1016/j.chaos.2022.111989

[7] R. Bechikh, H. Hermassi, A. A. Abd El-Latif, R. Rhouma, and S. Belghith, "Breaking an image encryption scheme based on a spatiotemporal chaotic system," *Signal Processing: Image Communication*, vol. 39, pp. 151-158, 2015. https://doi.org/10.1016/j.image.2015.09.006

[8] W.-S. Yap, R. C.-W. Phan, W.-C. Yau, and S.-H. Heng, "Cryptanalysis of a new image alternate encryption algorithm based on chaotic map," *Nonlinear Dynamics*, vol. 80, no. 3, pp. 1483-1491, 2015. https://doi.org/10.1007/s11071-015-1956-x

[9] B. Norouzi, S. Mirzakuchaki, and P. Norouzi, "Breaking an image encryption technique based on neural chaotic generator," *Optik*, vol. 140, pp. 946-952, 2017. https://doi.org/10.1016/j.ijleo.2017.04.103

[10] C. Li, G. Luo, K. Qin, and C. Li, "An image encryption scheme based on chaotic tent map," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 127-133, 2017. https://doi.org/10.1007/s11071-016-3030-8

[11] A. Jolfaei, "Comments on an image encryption scheme based on a chaotic Tent map," *arXiv preprint arXiv:1611.00381*, 2016. https://doi.org/10.48550/arXiv.1611.00381

[12] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Optics and Lasers in engineering*, vol. 88, pp. 197-213, 2017. https://doi.org/10.1016/j.optlaseng.2016.08.009

[13] T. Hu, Y. Liu, L.-H. Gong, and C.-J. Ouyang, "An image encryption scheme combining chaos with cycle operation for DNA sequences," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 51-66, 2017. https://doi.org/10.1007/s11071-016-3024-6

[14] X. Wu, H. Shi, M. Ji'e, S. Duan, and L. Wang, "A novel image compression and encryption scheme based on conservative chaotic system and DNA method," *Chaos, Solitons & Fractals*, vol. 172, p. 113492, 2023. https://doi.org/10.1016/j.chaos.2023.113492

[15] L. F. Ávalos-Ruíz, C. J. Zúñiga-Aguilar, J. F. Gómez-Aguilar, H. M. Cortes-Campos, and J. E. Lavín-Delgado, "A RGB image encryption technique using chaotic maps of fractional variable-order based on DNA encoding," *Chaos, Solitons & Fractals*, vol. 177, p. 114306, 2023. https://doi.org/10.1016/j.chaos.2023.114306

[16] C. Zou and L. Wang, "A visual DNA compilation of Rössler system and its application in color image encryption," *Chaos, Solitons & Fractals*, vol. 174, p. 113886, 2023. https://doi.org/10.1016/j.chaos.2023.113886

[17] C. Senthilkumar, M. Thirumalaisamy, R. K. Dhanaraj, and A. Nayyar, "DNA encoded color image encryption based on chaotic sequence from neural network," *Journal of Signal Processing Systems*, vol. 95, no. 4, pp. 459-474, 2023.

[18] A. Souyah and K. M. Faraoun, "An image encryption scheme combining chaos-memory cellular automata and weighted histogram," *Nonlinear Dynamics*, vol. 86, no. 1, pp. 639-653, 2016. https://doi.org/10.1007/s11071-016-2912-0

[19] W. Srichavengsup and W. San-Um, "Data encryption scheme based on rules of cellular automata and chaotic map function for information security," *International Journal of Network Security*, vol. 18, no. 6, pp. 1130-1142, 2016.

[20] Y. Dong, G. Zhao, Y. Ma, Z. Pan, and R. Wu, "A novel image encryption scheme based on pseudo-random coupled map lattices with hybrid elementary cellular automata," *Information Sciences*, vol. 593, pp. 121-154, 2022. https://doi.org/10.1016/j.ins.2022.01.031

[21] W. Lv, J. Chen, X. Chai, and C. Fu, "A robustness-improved image encryption scheme utilizing life-liked cellular automaton," *Nonlinear Dynamics*, vol. 111, no. 4, pp. 3887-3907, 2023. https://doi.org/10.1007/s11071-022-08021-1

[22] A. Y. Darani, Y. K. Yengejeh, H. Pakmanesh, and G. Navarro, "Image encryption algorithm based on a new 3D chaotic system using cellular automata," *Chaos, Solitons & Fractals*, vol. 179, p. 114396, 2024. https://doi.org/10.1016/j.chaos.2023.114396

[23] F. Yang, J. Mou, K. Sun, and R. Chu, "Lossless image compression-encryption algorithm based on BP neural network and chaotic system," *Multimedia Tools and Applications*, vol. 79, no. 27, pp. 19963-19992, 2020. https://doi.org/10.1007/s11042-020-08821-w

[24] X. Wang *et al.*, "A new V-Net convolutional neural network based on four-dimensional hyperchaotic system for medical image encryption," *Security and Communication Networks*, vol. 2022, no. 1, p. 4260804, 2022. https://doi.org/10.1155/2022/4260804

[25] E. A. A. Hagras, S. Aldosary, H. Khaled, and T. M. Hassan, "Authenticated public key elliptic curve based on deep convolutional neural network for cybersecurity image encryption application," *Sensors*, vol. 23, no. 14, p. 6589, 2023. https://doi.org/10.3390/s23146589

[26] M. Alsafyani, F. Alhomayani, H. Alsuwat, and E. Alsuwat, "Face image encryption based on feature with optimization using secure crypto general adversarial neural network and optical chaotic map," *Sensors*, vol. 23, no. 3, p. 1415, 2023. https://doi.org/10.3390/s23031415

[27] Q. Deng, C. Wang, and H. Lin, "Chaotic dynamical system of Hopfield neural network influenced by neuron activation threshold and its image encryption," *Nonlinear Dynamics*, vol. 112, no. 8, pp. 6629-6646, 2024. https://doi.org/10.1007/s11071-024-09384-3

[28] C. Jin and H. Liu, "A color image encryption scheme based on Arnold scrambling and quantum chaotic," *International Journal of Network Security*, vol. 19, no. 3, pp. 347-357, 2017.

[29] M. Jiang and H. Yang, "Image encryption using a new hybrid chaotic map and spiral transformation," *Entropy*, vol. 25, no. 11, p. 1516, 2023. https://doi.org/10.3390/e25111516

[30] S. Zhu, X. Deng, W. Zhang, and C. Zhu, "Secure image encryption scheme based on a new robust chaotic map and strong S-box," *Mathematics and Computers in Simulation*, vol. 207, pp. 322-346, 2023. https://doi.org/10.1016/j.matcom.2022.12.025

[31] W. Alexan, M. Gabr, E. Mamdouh, R. Elias, and A. Aboshousha, "Color image cryptosystem based on sine chaotic map, 4D chen hyperchaotic map of fractional-order and hybrid DNA coding," *IEEE Access*, vol. 11, pp. 54928-54956, 2023.

[32] Y.-M. Li, Y. Deng, M. Jiang, and D. Wei, "Fast encryption algorithm based on chaotic system and cyclic shift in integer wavelet domain," *Fractal and Fractional*, vol. 8, no. 2, p. 75, 2024. https://doi.org/10.3390/fractalfract8020075

[33] X. Hu, D. Jiang, M. Ahmad, N. Tsafack, L. Zhu, and M. Zheng, "Novel 3-D hyperchaotic map with hidden attractor and its application in meaningful image encryption," *Nonlinear Dynamics*, vol. 111, no. 20, pp. 19487-19512, 2023. https://doi.org/10.1007/s11071-023-08545-0

[34] S. Yan, L. Li, W. Zhao, and B. Gu, "Design of a new four-dimensional chaotic system and its application to color image encryption," *Nonlinear Dynamics*, vol. 111, no. 18, pp. 17519-17545, 2023. https://doi.org/10.1007/s11071-023-08726-x

[35] Q. Li and L. Chen, "An image encryption algorithm based on 6-dimensional hyper chaotic system and DNA encoding," *Multimedia Tools and Applications*, vol. 83, no. 2, pp. 5351-5368, 2024. https://doi.org/10.1007/s11042-023-15550-3

[36] J. Xin, H. Hu, and J. Zheng, "3D variable-structure chaotic system and its application in color image encryption with new Rubik's Cube-like permutation," *Nonlinear Dynamics*, vol. 111, no. 8, pp. 7859-7882, 2023. https://doi.org/10.1007/s11071-023-08230-2

[37] A. Bencherqui *et al.*, "Optimal algorithm for color medical encryption and compression images based on DNA coding and a hyperchaotic system in the moments," *Engineering Science and Technology, an International Journal*, vol. 50, p. 101612, 2024. https://doi.org/10.1016/j.jestch.2023.101612

[38] X. Wang and C. Liu, "A novel and effective image encryption algorithm based on chaos and DNA encoding," *Multimedia Tools and Applications*, vol. 76, no. 5, pp. 6229-6245, 2017. https://doi.org/10.1007/s11042-016-3311-8

[39] M. A. Murillo-Escobar, C. Cruz-Hernández, L. Cardoza-Avendaño, and R. Méndez-Ramírez, "A novel pseudorandom number generator based on pseudorandomly enhanced logistic map," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 407-425, 2017. https://doi.org/10.1007/s11071-016-3051-3

[40] Y. Liu, J. Tang, and T. Xie, "Cryptanalyzing a RGB image encryption algorithm based on DNA encoding and chaos map," *Optics & Laser Technology*, vol. 60, pp. 111-115, 2014. https://doi.org/10.1016/j.optlastec.2014.01.015

[41] H. Wen and Y. Lin, "Cryptanalyzing an image cipher using multiple chaos and DNA operations," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 7, p. 101612, 2023. https://doi.org/10.1016/j.jksuci.2023.101612

[42] Q. Zhang, L. Guo, and X. Wei, "Image encryption using DNA addition combining with chaotic maps," *Mathematical and Computer Modelling*, vol. 52, no. 11-12, pp. 2028-2035, 2010. https://doi.org/10.1016/j.mcm.2010.06.005

[43] X. Huang and G. Ye, "An image encryption algorithm based on hyper-chaos and DNA sequence," *Multimedia Tools and Applications*, vol. 72, no. 1, pp. 57-70, 2014. https://doi.org/10.1007/s11042-012-1331-6

[44] R. Guesmi, M. A. B. Farah, A. Kachouri, and M. Samet, "A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2," *Nonlinear Dynamics*, vol. 83, no. 3, pp. 1123-1136, 2016. https://doi.org/10.1007/s11071-015-2392-7

[45] H. Wen and Y. Lin, "Cryptanalysis of an image encryption algorithm using quantum chaotic map and DNA coding," *Expert Systems with Applications*, vol. 237, p. 121514, 2024. https://doi.org/10.1016/j.eswa.2023.121514

[46] S. C. Phatak and S. S. Rao, "Logistic map: A possible random-number generator," *Physical Review E*, vol. 51, no. 4, pp. 3670–3678, 1995. https://doi.org/10.1103/PhysRevE.51.3670

[47] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," *Optics and Lasers in Engineering*, vol. 78, pp. 17-25, 2016. https://doi.org/10.1016/j.optlaseng.2015.09.007

[48] J. C. Dagadu, J.-P. Li, and P. C. Addo, "An image cryptosystem based on pseudorandomly enhanced chaotic DNA and random permutation," *Multimedia Tools and Applications*, vol. 78, no. 17, pp. 24979-25000, 2019. https://doi.org/10.1007/s11042-019-7693-2

[49] L. G. De La Fraga, E. Torres-Pérez, E. Tlelo-Cuautle, and C. Mancillas-López, "Hardware implementation of pseudo-random number generators based on chaotic maps," *Nonlinear Dynamics*, vol. 90, no. 3, pp. 1661-1670, 2017. https://doi.org/10.1007/s11071-017-3755-z

[50] J. C. Dagadu, J.-P. Li, and E. O. Aboagye, "Medical image encryption based on hybrid chaotic DNA diffusion," *Wireless Personal Communications*, vol. 108, no. 1, pp. 591-612, 2019. https://doi.org/10.1007/s11277-019-06420-z

[51] A. Mitra, Y. V. S. Rao, and S. R. M. Prasanna, "A new image encryption approach using combinational permutation techniques," *International Journal of Computer Science*, vol. 1, no. 2, pp. 127-131, 2006.

[52] S. R. M. Prasanna, M. Ashalatha, S. Nirmala, and K. Haribhat, "Study of permutations in the context of speech privacy," presented at the International Conference on Evolutionary Computing for Computer Communications, Control and Power (ECCAP 2000), 2000.

[53] X. Sun and Z. Chen, "A novel chaotic image encryption algorithm based on coordinate descent and SHA-256," *IEEE Access*, vol. 10, pp. 114597-114611, 2022. https://doi.org/10.1109/ACCESS.2022.3217520

[54] J. Wu, J. Zhang, D. Liu, and X. Wang, "A multiple-medical-image encryption method based on SHA-256 and DNA encoding," *Entropy*, vol. 25, no. 6, p. 898, 2023. https://doi.org/10.3390/e25060898

[55] B. Rahul, K. Kuppusamy, and A. Senthilrajan, "Chaos-based audio encryption algorithm using biometric image and SHA-256 hash algorithm," *Multimedia Tools and Applications*, vol. 82, no. 28, pp. 43729-43758, 2023. https://doi.org/10.1007/s11042-023-15289-x

[56] K. Sattaiah and K. Chinnaiah, "Providing security in genesis and other blocks of blockchain technology using SHA-256 algorithm," presented at the 2024 3rd International Conference for Innovation in Technology (INOCON), IEEE, 2024.

[57] D. Singh, H. Kaur, C. Verma, N. Kumar, and Z. Illés, "A novel 3-D image encryption algorithm based on SHA-256 and chaos theory," *Alexandria Engineering Journal*, vol. 122, pp. 564-577, 2025. https://doi.org/10.1016/j.aej.2025.03.026

[58] K. Al-Fatlawi and J. Kazemitabar, "A comprehensive security framework for wireless sensor networks using SHA-256 and CNNs," *International Journal of Engineering, Transactions A: Basics*, vol. 38, no. 1, pp. 205-222, 2025. https://doi.org/10.5829/ije.2025.38.01a.19

[59] E. S. M. El-Alfy, S. M. Thampi, H. Takagi, S. Piramuthu, and T. Hanne, *Advances in intelligent informatics*. Cham: Springer International Publishing, 2015.

[60] K. Zhan, D. Wei, J. Shi, and J. Yu, "Cross-utilizing hyperchaotic and DNA sequences for image encryption," *Journal of Electronic Imaging*, vol. 26, no. 1, pp. 013021, 2017. https://doi.org/10.1117/1.JEI.26.1.013021

[61] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656-715, 1949. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x

[62] IEEE Computer Society Standards Committee Working Group of the Microprocessor Standards Subcommittee & American National Standards Institute, *IEEE standard for binary floating-point arithmetic (ANSI/IEEE Std 754-1985)*. New York: IEEE, 1985.