Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082352

ARTICLE

SHA-512 Based Key Generation and Two-Dimensional Logistic Permutation with a Median Filter for Enhanced Grayscale Image Encryption

Ibtisam A. Taqi*

Computer Science Department, University of Baghdad, Baghdad, Iraq

*Corresponding Author: Ibtisam A. Taqi. Email: ibtisam.taqi@sc.uobaghdad.edu.iq

Received: 14 March 2026; Accepted: 23 July 2026; Published: 15 September 2026

ABSTRACT: Multimedia technology and social media platforms like Facebook, Telegram, Viber, and WhatsApp, as well as numerous industries like communications, banking, and the military, depend heavily on images. Therefore, the biggest issue these days is safeguarding the image from theft or hacking during storage or transmission over the internet. This study suggests a novel approach of grayscale image encoding that uses a two-dimensional (2D) logistic map and the Secure Hash Algorithm (SHA). First, convert a color image to grayscale. Second, use the recently proposed equations to calculate the initial states of the Two-Dimensional Logistic Map (2DLM). Then, permute the gray image using the new coordinates produced by a 2DLM. SHA-512 is used to construct an extremely complex random key according to the plain image size. The encoded image is then produced by applying two-directional diffusion stages between the generated key and the shuffled image. Lastly, to improve the effectiveness of the proposed method in recovering the decrypted image, a median filter is applied to the decrypted image as a post-processing step. The proposed method surpasses other works by encrypting large sizes of images equal to 2048 × 2048 and achieves a high entropy approximately equal to 7.99996, a high Number of Pixel Change Rate (NPCR)of 99.86, a high Unified Average Changing Intensity (UACI) of 39.31%, a low Peak Signal to Noise Ratio (PSNR) of 4.9279, and a high Mean Square Error (MSE) of 20,907. Furthermore, a high key precision of 10^{-19}, a high keyspace of 2^{1140}, a very low average correlation coefficient of 0.0006, a flat histogram, and a high speed are attained. According to the evaluation metrics, the proposed method performs better than the other tasks that were done. The system outperforms the other works in terms of security against noise attacks, in which approximately 61% of the image with 50% salt and pepper noise and up to 70% of the image with 0.1 Gaussian noise were recovered.

KEYWORDS: Gray image; SHA-512; chaos; 2D logistic map; median filter

1 Introduction

With the rapid advancement of multimedia and network technologies, there is now a high demand for image transfer. Large data sets, robust correlations, and significant redundancy are all features of image data. To safeguard photos while they are being stored and transmitted over the Internet, image encryption has become a common necessity. Although conventional cryptographic algorithms such as DES and AES can be employed for image encryption, their general-purpose design does not specifically exploit the statistical and structural characteristics of digital images, such as high inter-pixel correlation and large data size. Consequently, chaotic maps have attracted considerable attention in image encryption due to their desirable properties, including high sensitivity to initial conditions and control parameters, complex and deterministic dynamics, and the ability to generate pseudo-random sequences. These properties can be effectively integrated into permutation and diffusion mechanisms to reduce statistical dependencies and

enhance the security of digital images during transmission and storage [1–3]. In order to improve the system's performance against noise attacks when retrieving the decrypted image with no data loss or an excessively low loss of information, this paper suggests a new encoding technique for grayscale images based on SHA-512 combined chaos using 2D logistic maps. The median filter is used as a digital image post-processing method. The following is the arrangement of the research sections: The additional works are explained in Section 2. The SHA-hash function, 2D logistic maps, and median filter processing are explained in Section 3. Section 4 provides a detailed explanation of the proposed system. The criteria for assessing the proposed system efficiency, results and the attack analysis were demonstrated in Section 5. Comparision with the other works are introduced in Section 6, and Conclusions are finally shown in Section 7.

This paper developed a highly secure encryption system that combines the iterative SHA-512 hash function with an optimal permutation method based on initial control states generated by the proposed highly efficient equations. The paper's novelty is listed as:

1. The encryption has a huge image size of 2048 × 2048 pixels without any loss of data (lossless method).
2. Achieves a high key pace of 2^{1140}, indicating high resistance to brute force attacks.
3. High resistance to statistical attacks, demonstrated by a high entropy of approximately 7.99996, and low correlation between neighbors, including VCC (0.001278–0.000894), HCC (0.001330–0.000087), and DCC (0.001780–0.000046).
4. High resistance to two types of noise attacks that recovers up to 70% of the image after 0.1 Gaussian noise. It recovers 56%–61% of the image after 50% salt-and-pepper noise.
5. High resistance to 25% cropping attacks, which retrieves 62% of the image.
6. Excellent key sensitivity, achieving an NPCR of 99.61%, and a UACI of 32.54%–41.37%, indicating significant cipher variations in response to minor key changes.
7. Strong resistance to differential attack by achieving an MAE of 85%–100%, an NPCR of 99.60%–99.86%, and a UACI of 33.46%–39.31%, while a good avalanche effect reached 53%.
8. An efficient method that encrypts a huge-sized image within approximately 3–4 s only.

2 The Other Works

Many articles on grayscale and color image encoding have been published, including the following:

In 2018, Hameed and Taqi [1] introduced a new schema for color image encryption utilizing deoxyribonucleic acid (DNA) and multi-chaotic maps such as the Beta, Sine, and Chen Hyper map. The outcomes show resistance to a statistical, plaintext, brute force, and differential attacks. In 2023, Taqi and Abdul-Haleem [2] presented a novel method for encrypting grayscale images using 1D and 2D logistic chaotic maps. The results demonstrate strong security and resistance to noise attacks. In 2023, Akraam et al. [3] presented a new technique that used Tent, Henon, 1D logistic Map, and Piecewise Linear Map. The findings show appropriateness for an online system and a high degree of security. In 2015, Guesmi et al. [4] suggested a new encryption algorithm based on Secure Hash Algorithm SHA-2, the Lorenz system, and a (DNA). The main benefits are to enhancee the randomness and the information entropy, which can withstand a variety of common attacks and produce positive experimental outcomes. In 2017, Ben Slimane et al. [5] presented a quick, safe, and reliable method for encrypting digital images utilizing the Secure Hash Algorithm SHA-1, the Lorenz chaotic system, and the 4D hyper-chaotic system. The outcomes of security analysis include entropy information, statistical testing, key sensitivity, key space, and differential attacks. In 2018, Ahmad et al. [6] assessed a new image encryption technique based SHA-512 and the characteristics of the 4D hyper-chaotic system which was created mainly to help the network systems maintain privacy. The study show that the enhanced image encryption system is resistant to potential cryptanalytic attacks, and achieve the key sensitivity. In 2018, Zhu et al. [7] proposed a new image encryption technique based on

SHA-256, a Chebyshev map and hyper-chaotic system to address the challenge of key management in "one time pad" encryption schemes and also withstand the attack of selected plaintext. The suggested approach is useful for the safe transmission applications and images storage. In 2020, Xu et al. [8] suggested a new image encryption technique that makes use of 6D hyper-chaotic systems and bit-plane matrix rotation. The technique efficiently uses Psedu Random Numbers (PRNS) produced by the hyper-chaotic systems to jumble bits in the bit-plane matrix. Additionally, the Message-digest Algorithm 5 (MD5) hash value used to obtain some encryption process parameters. The results improving the correlation between the encryption process and the plain-image. In 2020, Shah et al. [9] presented an effective image encryption technique for real-time images. The suggested method reduces the amount of the chaotic value vector needed to permute a real-time image by combining encryption with an effective permutation methodology based on a modular logistic map. As a result, there is a reduction in both computational and temporal complexity, addressing the significant issue of slow speeds brought on by intricate processes in real-time images. In 2021, Firdous et al. [10] created a novel encryption method based SHA-256, a logistic map and the water wave. Stones were thrown onto the regular image to scatter the pixels, which acts as a water pond. The proposed technique uses a huge pool of secret keys to meet the necessary security standards. In 2021, Wu et al. [11] suggested an image encryption technique that uses SHA-256 controlled chaotic systems in conjunction with adversarial neural cryptography (ANC) which is considered extremely non-linear since neural network (NNs) are inherently non-linear. A logistic-sine map is used to generate a pseudorandom masking matrix. The results demonstrate that the technique can withstand a number of typical assaults, has a large key space and high key sensitivity. In 2022, Dua et al. [12] presented a secure and effective non-linear chaotic based image encryption method that uses a logistic map and SHA-256 key sequence in order to encrypt several images at once. Afterward, divide the color image into three-color components. The second step involves rotating and rearranging the image's pixels (the rotational angle might be any random angle). The results demonstrated by the entropy and CC values of the encrypted image to withstand statistical attack. In 2024, Niu et al. [13] presented an enhanced four-dimensional 4D chaotic system and incorporates evolutionary operators to suggest an image encryption scheme that addresses the shortcomings of current chaotic system-based image encryption schemes, specifically with regard to resistance to differential attacks and the unstable performance of chaotic systems, and improves the security of image data transmission. In 2024, Kumaran et al. [14] proposed a modified logistic map is used to extract random sequence. A modified zigzag transform is used to swap pixel positions. Then, dynamic tetrameric DNA encryption and diffusion, using DNA decryption procedures, significantly amplify the swapping effect. Finally, DNA decryption completes the encryption process. The experimental results showed resistance to noise attacks of 15%, cropping attacks of 25%, and it also resist brute force and statistical attacks. In 2024, Ernastuti et al. [15] proposed a schema based on logistic chaotic map for shuffling and substitution. The pixel positions in the original image swapped using perfect permutation based on the ascending key stream sequence. The results withstand brute force, statistical, and differential attack. In 2025, Singh et al. [16] presented a three-stage new image encryption method based on chaotic maps. Prior to encryption, a 3-D image transformed into a format comparable to that of 2-D images. The SHA-256 function is used to the plaintext's coordinate matrix to create the chaotic system's beginning conditions. Three sequences produced by the logistic-dynamic coupled logistic map lattice (LDCML) model are employed to confuse and diffuse the coordinate values. A tent map between the coordinate points used to create the confusion. The results capable of withstanding a variety of traditional attacks. In 2025, Anujaa et al. [17] presented a novel architecture of 5D hyper chaotic map for diffusion step with developed SHA-512 to generate secret key. The outcomes show a high level of security and resist a statistical attack. In 2025, Ammar and Abdullah [18] proposed a method using 3D logistic map for creating chaotic keys, scrambling the image pixels, and then Xored the results with the chaotic keys. The outcomes demonstrated high security against differential attacks and high key sensitivity to the initial conditions. In 2026, Zhang and Hu [19]

proposed a novel 2D sine cubic modular map (2D-SCMM) and a rotating dial model. A diagonal cyclic-shift transformation is used to dynamically adjust the distribution of pixel positions during the scrambling phase. The revolving dial model is used to perform pixel updates during the diffusion phase, modeled after a dial phone. The results resisted 20% of noise attacks and 50% of cropping attacks on 512 × 512 images. In 2026, Amutha and Phamila [20] suggested approach make use of a cosine expanded logistic chaotic map with increased randomness and a broader chaotic range. Using chaotic pixel permutation, controlled flipping, modullo arithmatic, MSB/LSB separation, and cross quadrant bitwise operations, the encryption approach combine two confusion and two diffusion phases to provide light yet reliable security appropriate for systems with limited resources. The experimental results withstand differential, statistical, and brute force attacks. In 2026, Kumar and Jaishree [21] presented a light-weight image encryption technique for grayscale and color images of size (256 × 256) and (512 × 512) that combine the chaos-driven confusion diffusion with the walsh-hadamard transform (WHT). In addition to an explicitly defined composite key spacefrom logistic and tent maps with optional Arnold iterations, uniqness is achieved through direct binary chaotic masking in the WHT domain. The results showed high randomness, differential robustness, and suitability for real time and resource-constraint deployments.

3 SHA-Hash Function, 2D Logistic Map and Median Filter

This section illustrates the fundamental of SHA, the chaotic map used for image encryption, and the median filter as a post-processing technique. This work uses a 2D logistic chaotic map for random permutation, SHA-512 to generate the secret key.

3.1 SHA-512

A cryptographic hash algorithm that belongs to the SHA-2 family is called SHA-512. It creates a fixed 512-bit (uses eight 64-bit words) hash value from any input, including files, messages, and more. Digital signatures, data integrity checks, block chain, and password hashing (with additional key stretching or salting) all make extensive use of it. The main steps of the SHA function include message padding, message scheduling, iterative round processing, and final hash generation. The message to be hashed is first parsed into 1024-bit message sub-blocks after being padded with its length till the result is a multiple of 1024 bits. Each of the sub-blocks is handled iteratively each time. After processing all message sub-blocks, returning the final 512-bit hash digest with a set initial hash value as shown in Eq. (1) [4]

$$H = H_1, H_2, \ldots\ldots\ldots, H_{64} \tag{1}$$

where, each $H_i = \{h_{i1}, h_{i2}, \ldots, h_{i8}\}$ is i-th byte in hash H.

The proposed method utilizes the SHA-512 hash function as a pseudo-random generator to derive a large key matrix from an initial seed value. Since SHA-512 produces a fixed-length output of 512 bits, it is insufficient to directly construct a large-scale key (e.g., 1024 × 1024 or more). Therefore, an iterative expansion approach is employed. Initially, a secret seed is provided as input to the SHA-512 function to generate the first hash output. Subsequently, the output of each iteration is recursively feedback as the input to the hash function to produce a sequence of hash values. This process is repeated until a sufficient number of bits are generated to meet the required key size. The concatenated hash outputs form a long pseudo-random bitstream.

3.2 2D Logistic Map

The 2D logistic map is a chaotic system that randomly generates new positions for the diffusion step. Two coupled nonlinear equations define it as in Eq. (2) [2]

$$\begin{aligned} x_{n+1} &= r_1 x_n \left(1 - x_n\right) + s_1 y_n^2 \\ y_{n+1} &= r_2 y_n \left(1 - y_n\right) + s_2 x_n^2 + x_n y_n \end{aligned} \tag{2}$$

where $r_1 = 0.98$, $r_2 = 0.66$, $s_1 = 0.18$, $s_2 = 0.15$ are 2D logistic control parameters

3.3 Median Filter

The median filter approach, an ordered non-linear statistical numerical filtering technology, improves the performance of the proposed system by removing noise from image. It is a crucial post-processing step in the image processing industry as shown in Eq. (3) [2,22]

$$B = midfilt3(A, [mn]) \tag{3}$$

where A is the decrypted noisy image and B is the smoothed image. m and n are the block mask width and height, respectively.

4 The Methodology of the Proposed Method

In this part, shuffling step, key generation strategy, encryption proposed system, and the smoothing steps as the post-processing step are clarified. Fig. 1 shows the proposed encryption and decryption system

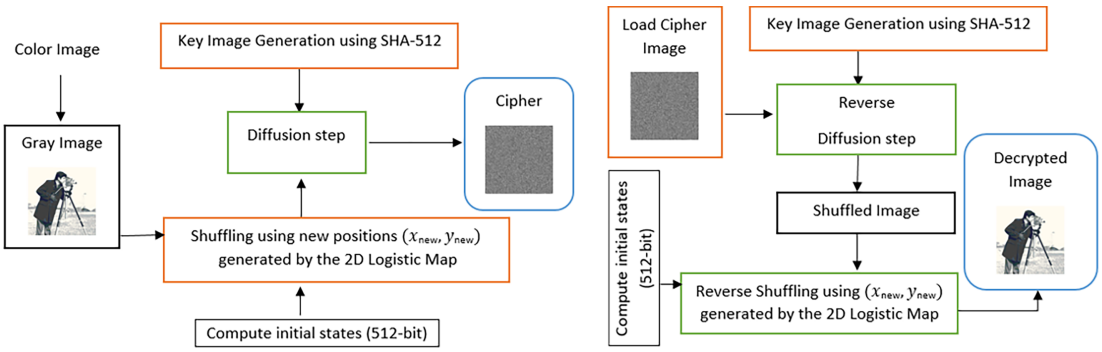


Figure 1: Proposed image encryption and decryption system.

4.1 Shuffling Step Using 2D Logistic Map

The new coordinates are generated using a 2D logistic chaotic map, whose behavior is influenced by the initial states and control parameters. A 512-bit key sequence is divided into 64 bytes, denoted by $(k_1 - k_{64})$. Each k_i is an 8-bit byte represented by two hexadecimal digits. The key values were experimentally determined to obtain the initial states for the 2D chaotic map, which are computed using the proposed Eqs. (4)–(6).

$$x_0 = mod\left((k_1 + k_2 + .. + k_{20})/2^{20}, 1\right) \tag{4}$$

$$x_1 = mod\left((k_{21} + k_{22} + .. + k_{44})/2^{24}, 1\right) \tag{5}$$

$$x_2 = mod\left((k_{45} + k_{46} + .. + k_{64})/2^{20}, 1\right) \tag{6}$$

Additionally, Eq. (7) yields the initial value of (y_0)

$$y_0 = mod\left((x_0 + x_1 + x_2), 1\right) \tag{7}$$

Using Eq. (2) and applying the new proposed Eqs. (8) and (9) to generate x_k and y_k as intermediate values before generateing the new positions

$$x_k = int\left(floor\left(mod\left(\left(x_{n+1} \times 10^{19}\right), w\right)\right)\right) \tag{8}$$

$$y_k = int\left(floor\left(mod\left(\left(y_{n+1} \times 10^{16}\right), w\right)\right)\right) \tag{9}$$

where $\forall n, 0 \leq n \leq (w \text{ or } h) - 1$, $\forall n, 1 \leq k \leq (w \text{ or } h)$, h and w are the image height and width, respectively, with $(w = h)$.

Fig. 2a illustrates the Lyapunov exponent (λ) of the 2D logistic map as the control parameter r1 varies from 2.5 to 4. The Lyapunov exponent is positive over most of the parameter range, confirming the existence of chaotic behavior with high sensitivity to initial conditions. Several local decreases are observed, corresponding to periodic windows where the system temporarily exhibits periodic dynamics. As r1 approaches 4, (λ) reaches higher positive values, indicating stronger chaos and enhanced randomness, which are desirable characteristics for secure image encryption.

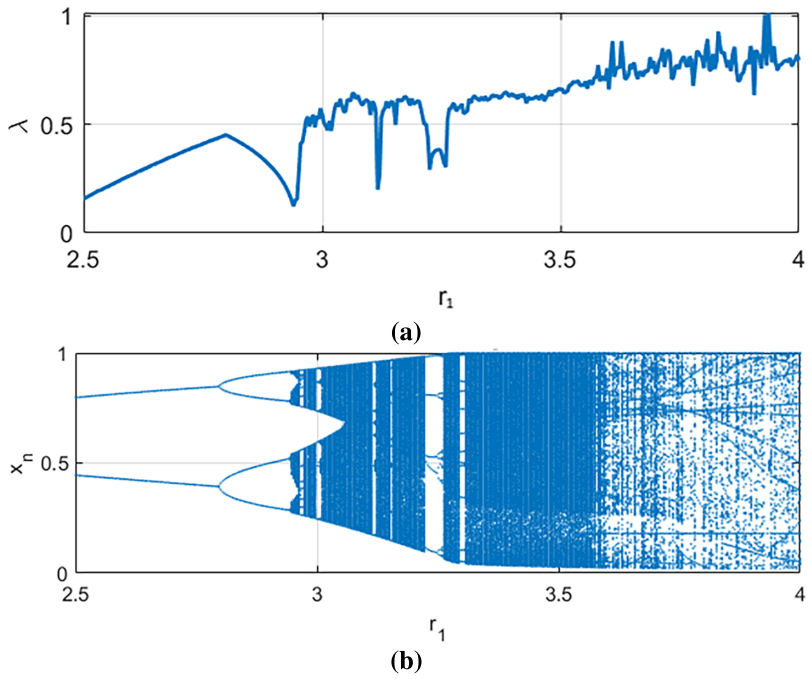


Figure 2: (**a**) Lyapunov exponent and (**b**) Bifurcation diagram of the 2D logistic map with its control parameters and initial states of the proposed method.

Fig. 2b presents the bifurcation diagram of the 2D logistic map. As the control parameter r1 increases, the system evolves from a stable fixed point to successive period-doubling bifurcations before entering a fully chaotic regime. The dense distribution of state values in the higher parameter range demonstrates complex and non-periodic behavior. The chaotic region agrees well with the positive Lyapunov exponent observed in Fig. 2a, confirming the suitability of the proposed map for generating highly random sequences in image encryption applications.

4.2 Key Image Generation Using SHA-512 Function

Suppose the size of the desired key image Key_image is $width \times height$ and $seed$ is an initial secret string input = "SuperSandra81". The steps to generate the secret key image are detailed as follows:

Step 1: Produce a 512-bit (64-byte) hash output for each hashing operation by importing and configuring the Java Message Digest library to utilize the SHA-512 cryptographic hash function as shown in Eq. (10)

$$md = MessageDigest.getInstance('SHA - 512') \tag{10}$$

Step 2: Compute the total size of the output key matrix (bytes required) as shown in Eq. (11)

$$total\ values = prod(imgSize) \tag{11}$$

Step 3: Initialize a one-dimensional buffer (key stream) of length total values to store the generated pseudo-random byte sequence.

Step 4: Convert the initial seed string into an array of uint8 values to form the first input data block for SHA-512 processing.

Step 5: An iterative hashing procedure is performed as follows until the key stream is completely filled according to the size of the desired key image:

- Apply SHA-512 to the initial secret seed and extract the resulting 64-byte digest
- Append the digest bytes sequentially into the key stream
- Replace the input data (next seed) with the newly generated hash value
- Continue the repetition

Step 6: Serve each output block as input for the next to create a cryptographically strong pseudo-random expansion of the original seed as a feedback mechanism.

Step 7: Reshape the one-dimensional key stream into a two-dimensional matrix of size $[width, height]$.

Step 8: Generate key matrix $key\_image[width, height]$. The resulting key_image forms a secure key matrix in which the distribution of values is fully determined by the secret seed and SHA-512 hashing, providing randomness suitable for image encryption.

4.3 Encryption Steps

The two crucial cryptography properties of confusion and diffusion must be addressed by the proposed method. First, the color image is converted to grayscale. Next, as a confusion phase, use a 2D logistic map to shuffle the gray image. The cipher image is then created using Eq. (13) as a diffusion phase by applying the XOR operation between the key created using the SHA-function and the shuffled image.

The steps of the proposed image encryption technique are presented as follows:

Step 1: Load the Plain Image $I\,(width, height)$, where $width, height$ Image dimensions.

Step 2: Convert the RGB image I to grayscale $Gray\,(width, height)$.

Step 3: Calculate the 2D logistic map's initial states using Eqs. (4)–(7) respecivilye.

Step 4: As a confusion step using 2D logistic map Eq. (2), where x_0 = 2.2888e − 04 and y_0 = 3.7730e − 04. Then compute Eqs. (8) and (9) to create new positions x_{new} and y_{new}. To avoid repetition, check if x_k is not found in x_{new}, and add. Otherwise, increase by four steps and add. Likewise with regard to y_k. The new positions are indicated by $[x_{new}, y_{new}]$.

Step 5: Use the new places created in the previous step to shuffle the gray image as shown in Eq. (12)

$$Sh_G(i,j) = Gray\ (x_{new}, y_{new}) \tag{12}$$

$\forall i, 1 \le i \le w$ and $\forall j, 1 \le j \le h$

Step 6: Use SHA-512 depending on the plain image size to create a random complicated key, $Key\_Image(width, height)$, as clarified in Section 4.2.

Step 7: Apply two direction diffusion stages: a forward diffusion to create the intermediate encrypted image $C\_temp\,(width, height)$ as in Eq. (13), and then the backward diffusion to create cipher as in Eq. (14). Finally convert the 1D cipher image $C(i)$ to final 2D cipher image $C(w, h)$.

$$\begin{aligned} &C\_temp\,(1) = Sh_G\,(1) \oplus Key\_image\,(1) \\ &C\_temp\,(i) = mod\,(Sh_G\,(i) \oplus Key\_image\,(i) + C(i-1), 256) \\ &\forall i, 2 \le i \le w \times h \end{aligned} \tag{13}$$

$$\begin{aligned} &C\,(i) = mod\,(C\_temp\,(i) + C\_temp\,(i-1)\,, 256) \\ &\forall i, w \times h - 1 \le i \le 1 \end{aligned} \tag{14}$$

Step 8: Calculate the metrics between the encrypted and the plain images.

Step 9: Obtain the decrypted image; the decryption procedure goes backwards in the encryption steps.

Step 10: Before adding any attacks, calculate the metrics between the decrypted and plain images.

4.4 Smoothing Step using Median Filter

Evaluate the proposed method's efficacy by using two different kinds of noise (Salt and pepper, Gaussian) and Cropping Attack as in steps listed below:

Step 1: Load the Cipher Image $C(width, height)$ and add noise with different densities or crop parts of data.

Step 2: Decrypt the noisy image $noisy\_image(width, height)$, or the cropping image.

Step 3: Calculate the metrics used to compare the plain image with the decrypted image.

Step 4: Apply the following formula to enhance the decrypted noisy image or the decrypted cropping image using the 5 × 5 median filter as in Eq. (15)

$$Smoothy\_image = Midfilt3(Denoisy\_image, [5\ 5]) \tag{15}$$

Step 5: Identify the metrics between the original image and the $smoothly\_image$.

Step 6: Compare the results with those of earlier studies.

5 Experimental Results and Analysis

The data set used to assess the performance of the proposed method was obtained from USC-SIPI Image Database and Hlevkin Dataset, (accessed on 1 July 2025) [1,17]. MATLAB R2020a and an HP computer Core i7-10510U CPU running at 1.80–2.30 GHz with 16 GB of RAM were utilized. Couple, mandrill, airplaneU2, and airport encrypted images were displayed in several sizes in Fig. 3. Because the cipher images are undetectable, the figure makes it clear that the proposed method offers strong encryption. The decrypted images are 100% the same as the plain images with different sizes of MSE = 0, PSNR = ∞, NPCR = 0, UACI = 0, and SSIM = 1.

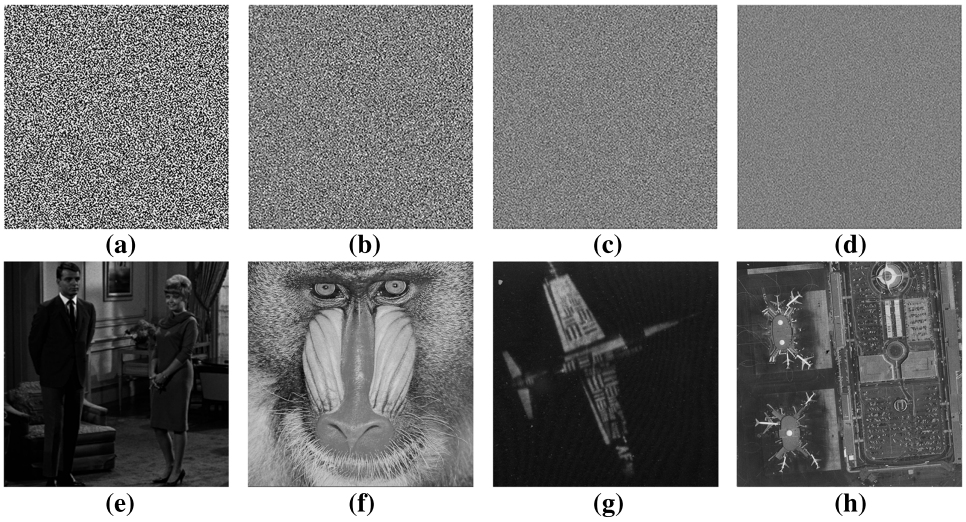


Figure 3: (**a**) Couple image, (**b**) Mandrill image, (**c**) AirplaneU2 and (**d**) Airport encrypted images of sizes (256 × 256), (512 × 512), (1024 × 1024), and (2048 × 2048), respectively, and (**e**–**h**) the decrypted images.

5.1 Keyspace Test

The system's ability to withstand a brute-force attack is demonstrated by the size of the key space. The proposed method employs two independent 512-bit secret keys. The first key is created from SHA-512 to generate the secret key image, providing a key space of 2^{512}. The second key is used to generate the initial states x_0, x_1, x_2 and y_0 of the 2D logistic map, also providing a key space of 2^{512}. The x sequence has a precision of 10^{-19}, while the $y$sequence has a precision of 10^{-16}. Thus, the overall key space is approximately equal to 2^{1140} $((10^{19} > 2^{63}) \times (10^{16} > 2^{53}) \times 2^{512} \times 2^{512})$. The computational precision reported in the literature is 10^{-14} for [1], 10^{-16} for [2], 10^{-15} for [3], 10^{-14} for [4], 10^{-12} for [5], 10^{-15} for [6,7], 10^{-16} for [8], 10^{-15} for [9], 10^{-14} for [10], 10^{-15} for [11], 10^{-14} for [13], 10^{-15} for [15], 10^{-12} for [17], and 10^{-15} for [21]. In comparison, the proposed method achieves higher computational precision and a larger key space than the reported methods, thereby providing stronger resistance against brute-force attacks and enhancing the overall security of the encryption system.

5.2 MSE, PSNR, NPCR, UACI, and SSIM (Structural Similarity Index Measure) Tests

The effectiveness of the proposed encryption method is evaluated using MSE, PSNR, and NPCR metrics, which compute the discrepancies between the original and encrypted images. MSE is defined in Eq. (16) [1,9,12]

$$MSE = \frac{1}{wh} \sum_{i=0,j=0}^{w,h} (I(i,j) - C(i,j))^2 \tag{16}$$

where $I(i,j)$ is the original image, $C(i,j)$ is an encrypted or decrypted image, and h and w are the image's dimensions.

Eq. (17) defines PSNR, a logarithmic number that indicates the ratio of the signal's maximum strength to the corrupted noise that affects how accurately it is represented on a decibel scale (dB) [1,3,9,12]

$$PSNR = 20\ log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \tag{17}$$

Eq. (18) computes the pixel difference ratio between two distinct images [1,3,5–9,11–13,16]

$$NCPR\left(Img1, Img2\right) = \frac{\sum_{i=1}^{w}\sum_{j=1}^{h} D\left(i,j\right)}{w \times h} \times 100 \tag{18}$$

$$D\left(i,j\right) = \begin{cases} 0, if\ Img1\left(i,j\right) = Img2(i,j) \\ 1,\ if\ Img1 \neq Img2(i,j) \end{cases}$$

*Img*1 and *Img*2 are plain, encrypted or decrypted images.

The UACI quantifies the average pixel intensity change between images. It indicates the effectiveness of visual information concealment when calculated between the plaintext and cipher images, and measures diffusion strength and resistance to differential attacks when calculated between two cipher images, as defined in Eq. (19) [1,2,17–21].

$$UACI\left(Img1, Img2\right) = \frac{\sum_{i=1}^{w}\sum_{j=1}^{h} \mid img1\left(i,j\right) - img1\left(i,j\right) \mid /255}{w \times h} \times 100 \tag{19}$$

Table 1 shows that the proposed scheme supports larger image sizes while achieving a high MSE of 20,890 and a low PSNR of 4.9314 between the plain and cipher images. Moreover, NPCR values around 99.62% and UACI values ranging from 27% to 48% indicate significant pixel and intensity changes after encryption. The low SSIM value of 0.0044, which is close to zero, confirms that the encrypted image retains almost no structural information from the original image. These results demonstrate the effectiveness of the proposed scheme in producing highly distorted cipher images and concealing visual information.

Table 1: MSE, PSNR, NPCR, and SSIM of the proposed method between the plain and the encrypted image.

Size	Image File	MSE	PSNR	NPCR	UACI	SSIM
2048 × 2048	AirplaneU2	15,142	6.3289	99.6182	39.9267	0.0115
	Airport	8664.9	8.7532	99.6099	30.0059	0.0246
	Female	8182.6	9.0019	99.6126	29.2667	**0.0044**
1024 × 1024	AirplaneU2	15,153	6.3258	99.6082	39.9513	0.0114
	Airport	8707.6	8.7318	99.6280	30.0702	0.0410
	Man	10,280	8.0109	99.6000	32.4734	0.0289
	Pentagon	9370	8.4134	99.6115	31.0811	0.0632
512 × 512	Boat	7636.2	9.3020	99.5975	28.4240	0.0388
	Cameraman	16,152	6.087	99.6197	41.4854	0.0489
	Gray21	11,383	7.5683	99.6216	34.1909	0.0025
	House	8907.7	8.6331	99.6063	30.3835	0.0398
	Mandrill	7268.7	9.5162	99.6098	27.8633	0.0685
256 × 256	Cameraman	11,700	7.4488	99.5987	34.6344	0.0404
	Couple	15,362	6.2663	99.6109	40.2696	0.0169
	Female	8164.7	9.0114	99.6124	29.2220	0.0277
	House	7709.4	9.2606	99.6201	28.5549	0.0303
	Resolution Chart	**20,907**	**4.9279**	**99.6292**	**48.7757**	0.0338

Note: Bold values indicate the best results obtained for each metric among the tested images.

5.3 *Entropy, Correlation Coefficient, Histogram Tests*

Information randomness degrees such as entropy (e), correlation coefficient, and histogram analysis are used to assess how well the proposed approach can withstand statistical attacks. Information entropy, defined in Eq. (20), measures the probability distribution of pixel intensity values in the encrypted image. An effective image encryption algorithm should produce an entropy value as close to the ideal value of 8 as possible [1,4–8,16].

$$e = entropy(C) \tag{20}$$

where C is the cipher image.

Eq. (21) counts the correlation which determines the association of the closely adjacent pixels in different directions [9].

$$r_{xy} = Corrcoef(X, Y) \tag{21}$$

where X and Y are neighboring pixels sets, and 1000 adjacent pixels are chosen in the image.

The histograms of various cipher images with varying sizes are shown in Fig. 4. There exhibit an almost uniform distribution across the gray-level range, indicating high randomness and strong resistance to statistical attacks.

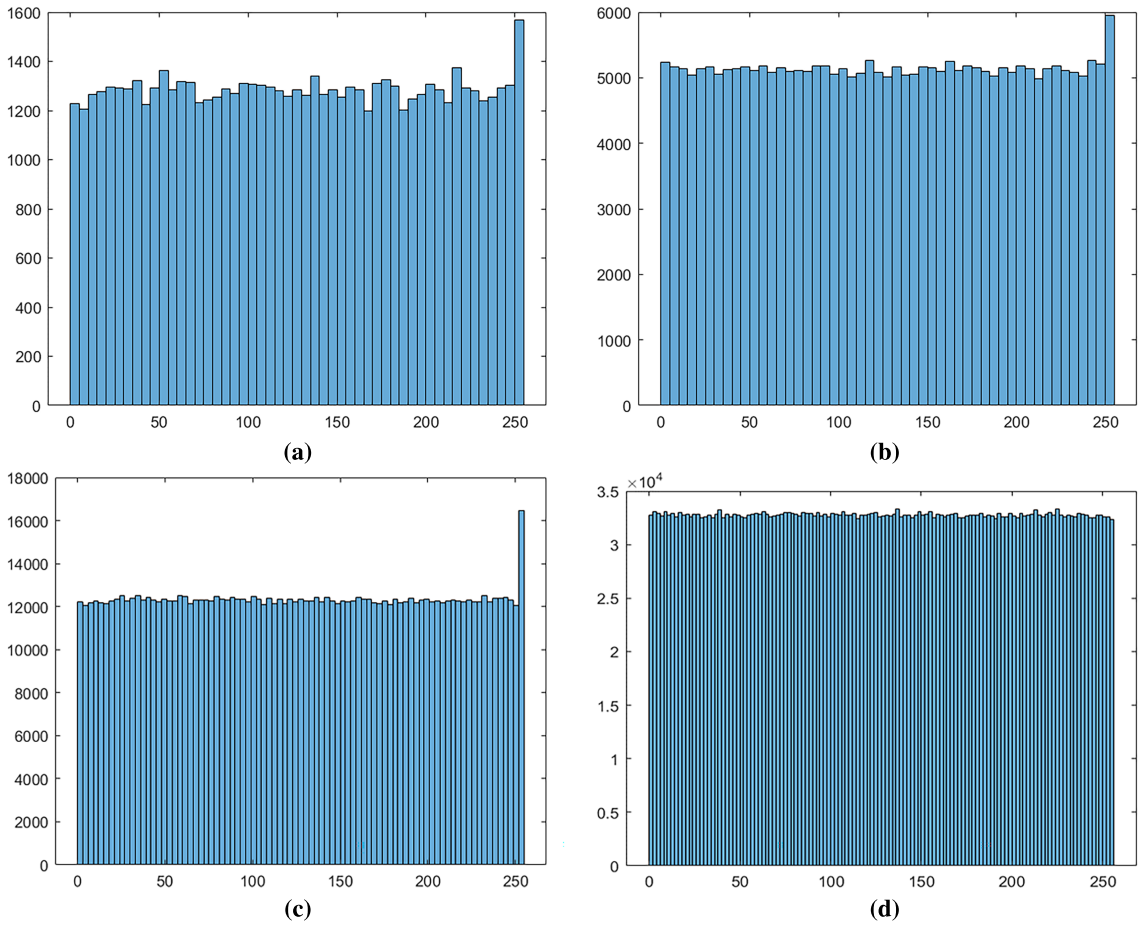


Figure 4: The histogram of the cipher images (**a**) 256 × 256 Couple image, (**b**) 512 × 512 Cameraman image, (**c**) 1024 × 1024 Pentagon image, and (**d**) 2048 × 2048 Airport image.

Fig. 5 shows the correlation distribution of adjacent pixels in the vertical, horizontal, and diagonal directions for the encrypted image. The random and uniformly scattered points indicate negligible correlation between neighboring pixels, confirming that the proposed encryption algorithm effectively removes statistical dependencies and provides strong resistance against statistical attacks.

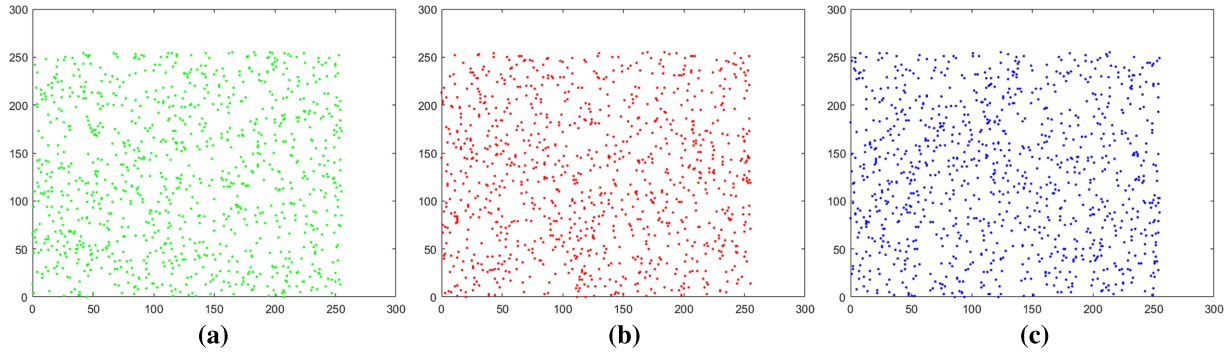


Figure 5: (**a**) Vertical, (**b**) Horizontal, and (**c**) Diagonal correlation coefficient of AirplaneU2 (2048 × 2048) cipher image.

Table 2 demonstrates that the proposed method is resistant to statistical attack and outperforms earlier work [3,5,8,19] by encrypting a larger image size and achieving high entropy, and a tiny correlation coefficient between the neighboring pixels in different directions. The proposed method outperforms technique [3] in terms of correlation coefficients (AVG-CC = 0.003–0.002 vs. 0.015–0.02) and Cameraman entropy (7.9971 vs. 7.9968). The proposed method yields better entropy than the technique in [5], and the average correlation values are nearly identical. Better correlation coefficients (AVG-CC = 0.003255 vs. 0.010566) than the technique [8]. Lastly, better correlation coefficients of mandrill image (AVG-CC = 0.000644 vs. 0.001933) and entropy (7.9993 vs. 7.9992), and boat image (AVG-CC = 0.002782 vs. 0.001396) than the technique [19].

Table 2: The proposed method entropy, vertical, horizontal, diagonal, and average correlation.

M.	Size	Image	VCC	HCC	DCC	Avg-CC	Entropy
Proposed Method	2048 × 2048	Airport	0.000156	**0.000087**	0.000721	0.000321	7.999954
		AirplaneU2	−0.000313	−0.000627	**0.000046**	0.000329	**7.999958**
		Female	**−0.000894**	0.000380	0.000241	0.000505	7.999955
	1024 × 1024	Man	0.002570	−0.000638	−0.000148	0.001119	7.9998
		AirplaneU2	0.000180	−0.002482	0.001780	0.001481	7.9998
		Airport	−0.001278	0.000165	−0.001882	0.001108	7.9998
		Pentagon	0.000430	0.002558	0.000259	0.001082	7.9998
	512 × 512	Boat	−0.001987	−0.002341	0.004019	0.002782	7.9993
		Cameraman	−0.000798	−0.000347	−0.001925	0.001023	7.9994
		Couple	0.003950	0.001330	0.000148	0.001810	7.9994
		Mandrill	−0.000104	−0.001382	0.000446	0.000644	7.9993
	256 × 256	Cameraman	0.004053	−0.003800	0.001813	0.003222	7.9971
		Couple	0.004817	−0.004702	0.006070	0.005196	7.9970
		House	−0.002387	−0.003215	−0.000574	0.002059	7.9971
		Pentagon	−0.001882	0.004770	−0.003375	0.003342	7.9974

(Continued)

Table 2 (continued)

M.	Size	Image	VCC	HCC	DCC	Avg-CC	Entropy
[3]	256 × 256	Cameraman	0.01620	0.01490	−0.01680	0.015967	7.9968
		House	−0.00310	−0.03120	−0.02940	0.021233	7.9972
[5]	512 × 512	Mandrill	−0.000591	0.00031	−0.001123	0.000674	7.9964
[8]	256 × 256	Couple	−0.0280	0.0031	0.0006	0.010566	7.9973
[19]	512 × 512	Mandrill	0.0005	0.0044	−0.0009	0.001933	7.9992
		Boat	0.0013	−0.0024	0.00049	0.001396	7.9993

Note: Bold values indicate the best results obtained for each metric among the tested images.

Overall, the proposed method outperforms all cited approaches and guarantees stronger results by consistently producing near-zero correlation and near-ideal entropy across all test situations.

5.4 Noise Analysis Test

When transmitting an image via a channel, high-security encryption systems must withstand various kinds of noise, including Gaussian and salt-and-pepper. Female (256 × 256) and Mandrill (512 × 512) cipher images are appended with several variances of Gaussian noise v = {0.01 and 0.1} with no mean and different densities of salt and pepper noise d = {20% and 50%}. Table 3 shows the Mean Absolute Error (MAE), which is the difference between the original images and the decrypted images under two different kinds of noise before and after the decrypted image was smoothed with a retrieval ratio.

Table 3: Mean Absolute Error (MAE) of Female (256 × 256) and Mandrill (512 × 512) plain and decrypted images under noise before (BS) and after smoothing (AS) with the retrieval ratio (RR).

Image Size	Noise Type	Ratio	MAE (BS)	RR	MAE (AS)	RR
256 ×256	Salt & Pepper	20%	36.4806	63.5194	7.6320	92.368
		35%	54.4134	45.5866	20.5323	79.4677
	Gaussian	0.01	60.3827	39.6173	27.3812	72.6188
		0.1	60.8536	39.1464	28.1778	71.8222
512 × 512	Salt & Pepper	20%	62.0252	37.9748	31.6940	68.306
		45%	59.3028	40.6972	29.3261	70.6739
	Gaussian	0.01	55.9311	44.0689	27.0593	72.9407
		0.1	56.2964	43.7036	27.3588	72.6412

According to the results, the proposed method achieves a very low MAE between the plain and decrypted images. It also attains a high retrieval ratio of approximately 80% under 35% salt-and-pepper and 72% under Gaussian noise with a variance of 0.1 for a 256 × 256 image. For 512 × 512 images, the proposed method achieves a high retrieval ratio of approximately 71% under 45% salt and pepper noise and 73% under Gaussian noise with a variance of 0.1. In comparison, the methods reported in [19,20] withstand 20% and 2% salt-and-pepper noise attacks, respectively, on 512 × 512 images.

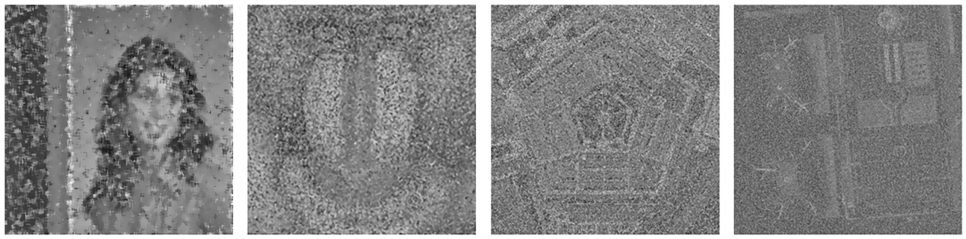
The proposed method is more secure against noise attacks on Airport image at a larger size (2048 × 2048), as evidenced by the MAE of 29.4216, which recovers approximately 71% under 0.1% Gaussian noise and recovers around 61% of the image under 50% salt-and-pepper noise, which can withstand a larger noise ratio as shown in Table 4. It recovers 62% under 0.1 Gaussian noise and recovers around 56% of the image under 50% salt-and-pepper noise on pentagon (1024 × 1024), so the results perform very well against the two types of noise on different image sizes.

Table 4: MAE of Pentagon (1024 × 1024) and Airport (2048 × 2048) plain and decrypted images under noise before (BS) and after smoothing with retrieval ratio (RR).

Image Size	Noise Type	Ratio	MAE (BS)	RR	MAE (AS)	RR
1024 × 1024	Salt & Pepper	20%	38.7322	61.2678	19.8615	80.1385
		50%	69.5330	30.4670	43.8648	56.1352
	Gaussian	0.01	64.2331	35.7669	37.4172	62.5828
		0.1	64.3535	35.6465	37.6752	62.3248
2048 × 2048	Salt & Pepper	20%	37.3547	62.6453	6.9837	93.0163
		50%	66.9431	33.0569	38.9211	61.0789
	Gaussian	0.01	61.7119	38.2881	29.0733	70.9267
		0.1	61.9588	38.0412	29.4216	70.5784

Figs. 6 and 7 present the recovered images with varying sizes after applying Salt-and-Pepper noise with a density of 50% and Gaussian noise with a variance of 0.1. The recovered images maintain high visual quality and preserve most image details across all tested noise levels, demonstrating the strong robustness and effective recovery capability of the proposed method against severe noise attacks.

Furthermore, it works better than [3,14], which can withstand 5% and 15% noise intensity, respectively. Noise removal is a critical post-processing step in image reconstruction, particularly in sensitive domains such as medical and military imaging. The presence of noise can significantly degrade visual quality and obscure essential details, leading to inaccurate analysis and decision-making. In medical applications, denoising enhances the visibility of fine anatomical structures, thereby improving diagnostic accuracy. Similarly, in military and surveillance systems, noise reduction facilitates reliable object detection and scene interpretation under challenging conditions. Therefore, effective denoising techniques are essential to restore image fidelity and ensure the integrity of extracted information when transmitted over the Internet after exposure to noise.



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

Figure 6: (**a**) Female (256 × 256), (**b**) Mandrilla (512 × 512), (**c**) Pentagon (1024 × 1024), and (**d**) Airport (2048 × 2048) decrypted results after 50% salt and pepper noise after smoothing, respectively.

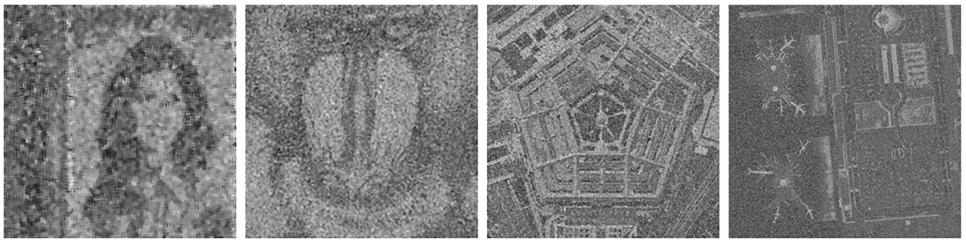


Figure 7: (**a**) Female (256 × 256), (**b**) Mandrill (512 × 512), (**c**) Pentagon (1024 × 1024), and (**d**) Airport (2048 × 2048) decrypted images after 0.1 Gaussian noise after smoothing, respectively.

5.5 Data Loss Analysis Test

The proposed algorithm or method is strong in data encryption when it resists crop attacks. This means that the data distribution is highly random, which is required to retrieve the data with the least possible loss. Table 5 demonstrates the robustness and unpredictability of the proposed method by showing that it passes a 25% cropping ratio by retrieving approximately 62% of the important information after the smoothing process as a post-processing step, which outperforms [3,9,14].

Table 5: MAE between the plain and the decrypted images, and the retrieval ratio (RR) after 25% cropping.

Image	Image Size	MAE	RR (BS)	MAE	RR (AS)
Pentagon	1024 × 1024	73.5725	26.4275	46.5615	53.4385
Female	2048 × 2048	69.3843	30.6157	37.4768	62.5232

Fig. 8 illustrates the recovered images after a 25% cropping attack. Despite the removal of 25% of the image content, the recovered images remain visually clear, indicating the robustness of the proposed method and its capability to recover the original image with minimal visual degradation.

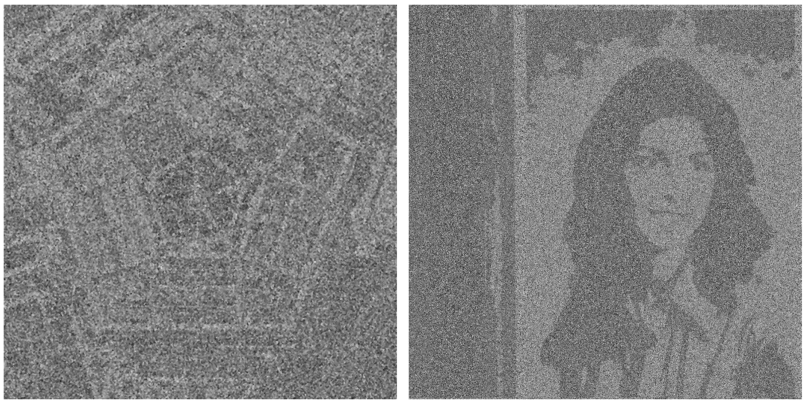


Figure 8: (**a**) Encrypted Pentagon (1024 × 1024), (**b**) Female (2048 × 2048) with cropping ratio 25%, respectively.

5.6 Speed Analysis Test

As shown in Table 6, the proposed scheme provides efficient encryption and decryption performance. Nevertheless, the methods presented in [5,20,21] achieve lower execution times. Hence, improving the computational efficiency of the proposed scheme represents a potential direction for future research.

Table 6: The execution time of the proposed method.

Image Size	Image	Encrypt Time (s)	Decrypt Time (s)
2048 × 2048	AirplaneU2	3.711071	3.254925
	Airport	3.619292	3.300334
	Female	3.666379	3.277668
1024 × 1024	AirplaneU2	0.879603	0.869465
	Man	0.886301	0.849703
512 × 512	Cameraman	0.250271	0.257769
	Mandrill	0.248026	0.243275
256 × 256	Couple	0.076727	0.062108
	Cameraman	0.078598	0.069389
	Pentagon	0.078939	0.068221

5.7 Computational Complexity Analysis

Let an input grayscale image have dimensions M × N, where L = M × N denotes the total number of pixels. The proposed encryption scheme consists of four main stages: key generation, chaotic permutation, forward and backward diffusion. In the key generation stage, the SHA-512-based mechanism produces a key stream proportional to the image size, resulting in a computational complexity of O(L). The permutation stage employs direct index mapping driven by chaotic sequences generated from the 2D logistic map without requiring any sorting operations, thereby maintaining a linear complexity of O(L). Subsequently, the diffusion performs forward in O(L) time and backward in O(L) time. Therefore, the overall computational complexity of the proposed method is $T(L) = O(L) + O(L) + O(L) + O(L) = O(4L) = O(MN)$ indicating good scalability for high-resolution image.

5.8 Key Sensitivity Analysis Test

The proposed method is more secure after adjusting each parameter separately, as evidenced by its sensitivity to a minor change in a 2D logistic initial state, only one at a time.

The control parameters of 2D logistic are set to x0 = 2.2888e − 04 and x1 = 6.2585e − 05, and SHA initial seed set to "SuperSandra81". Fig. 9 shows that the decrypted image after a slight change in one control parameter at a time namely $x0' = 2.2889e - 04$, $x1' = 6.2586e - 05$, or the SHA initial seed "SuperSandra80".

The NPCR up to 99.60% and UACI (32.5422–41.3760) results confirm the high key sensitivity of the proposed method, where a one-bit change in the initial seed ("SuperSandra81" to "SuperSandra80") resulted in substantial changes in the decrypted image, as shown in Table 7.

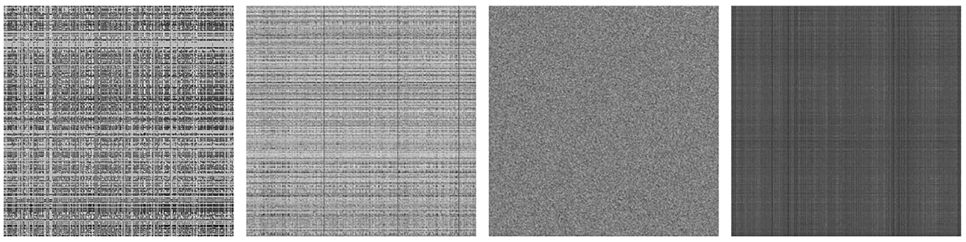


Figure 9: (**a**,**b**) the decrypted House (256 × 256), (512 × 512) using $x1'$, $x0'$, (**c**) the decrypted Man (1024 × 1024), after slight change in an initial seed of the key, respectively, and (**d**) the decrypted Airport (2048 × 2048) using $x0'$.

Table 7: NPCR and UACI between the plain and decrypted images after changing one bit in an initial seed.

Method	Image Size	Image	NPCR	UACI
Proposed M.	2048 × 2048	AirplainU2	99.6066	39.9122
	1024 × 1024	Man	99.6125	32.5422
	512 × 512	Cameraman	99.6128	41.3760
	256 × 256	Pentagon	99.6170	35.4982

5.9 Differential Attack Test

The resistance of the proposed method against differential attacks is evaluated using MAE, NPCR, UACI, and Avalanche Effect. The obtained results show an MAE (85%–100%), NPCR (99.60–99.86), and UACI (33.46–39.3147), while a good avalanche effect was achieved at 53%. These values indicate that even a slight change in the plain image (only one bit) leads to a significant and unpredictable variation in the encrypted output image. As shown in Table 8, it can be observed that the proposed method outperforms all other methods in terms of differential attack. The symbol (–) denotes that the MAE and avalanche effect are not applicable, whereas the pixel position was not reported.

Table 8: MAE, NPCR, UACI and avalanche effect between two encrypted images after changing one bit at a randomly selected pixel position in the plain image.

Method	Image Size	Image	MAE	NPCR	UACI	Avalanche Effect	Pixel Position
ProposedM.	2048 × 2048	AirplaneU2	85.2975	99.6095	33.4500	50.0016	(1907, 2048)
		Female	100%	99.8610	39.3147	53.6519	(1126, 1879)
	1024 × 1024	Airport	94.0658	99.6920	36.8885	52.6168	(293, 776)
		Man	88.7042	99.6493	34.7860	49.8790	(156, 440)
	512 × 512	Mandrill	97.0495	99.7601	38.0586	55.5424	(115, 385)
		cameraman	87.0700	99.6296	34.1451	50.6794	(469, 406)
		Boat	85.3405	99.6098	33.4669	49.9925	(372, 512)

(Continued)

Table 8 (continued)

Method	Image Size	Image	MAE	NPCR	UACI	Avalanche Effect	Pixel Position
	256 × 256	Aerial	85.3193	99.6094	33.4586	49.9773	(256, 256)
		Cameraman	90.0989	99.7070	35.3329	51.8978	(194, 191)
		Couple	94.0177	99.6979	36.8697	53.0680	(1, 13)
		House	88.7515	99.6994	34.8045	52.3109	(141, 246)
		Pentagon	89.5169	99.6948	35.1047	51.6287	(194, 71)
[3]	256 × 256	Cameraman	–	99.6521	33.4538	–	(217, 17)
		House	–	99.6353	33.3731	–	(128, 128)
[5]	512 × 512	Mandrill	–	99.59	33.4100	–	–
[7]	256 × 256	Cameraman	–	99.5692	33.3478	–	–
[8]	256 × 256	Couple	–	99.62	33.4500	–	–
[10]	512 × 512	Cameraman	–	99.6056	33.4591	–	–
	512 × 512	Mandrill	–	99.6064	33.4822	–	–
[11]	256 × 256	Cameraman	–	99.6338	33.5452	–	–
[13]	256 × 256	Pentagon	–	99.6033	33.4884	–	–
		House	–	99.5926	33.3774	–	–
		boat	–	99.5851	33.4002	–	–
[14]	512 × 512	Mandrill	–	99.6178	33.5468	–	–
	1024 × 1024	Man	–	99.6069	33.4806	–	–
[15]	512 × 512	Mandrill	75.0869	99.6073	33.2845	–	–
[17]	256 × 256	Pentagon	–	99.6307	33.4453	–	–
[18]	512 × 512	Mandrill	–	99.6132	33.5388	–	–
[19]	512 × 512	Mandrill	–	99.5922	33.4687	–	–
		Boat	–	99.5941	33.4169	–	–
[20]	256 × 256	Aerial	–	99.6017	33.4556	–	–
[21]	256 × 256	Cameraman	–	99.54	33.0500	–	–

5.10 Security Strength

The proposed method combines the advantages of chaotic systems and cryptographic hashing by employing two independent 512-bit keys: a SHA-512-based key for key-image generation and a chaotic key for permutation. This complete separation between the key generation and permutation stages enhances security and expands the key space. Moreover, all 64 bytes of the chaotic key are utilized to derive the initial states of the 2D logistic map, ensuring high sensitivity to initial conditions and generating highly random permutations for images of 2048 × 2048 pixels. Consequently, the proposed method achieves high entropy, low pixel correlation, histogram uniformity, and strong resistance against statistical, brute-force,

noise, cropping attacks, as well as high sensitivity to key variations. The proposed key separation strategy increases the difficulty of known-plaintext and differential attacks by preventing direct relationships between the permutation and diffusion stages.

5.11 NIST Test

The randomness of encrypted data is assessed using the NIST statistical test suite. The uniform randomness of the cipher image's pixel sequences is checked for image encryption. Strong encryption and defense against attacks are indicated by passing NIST tests. As shown in Table 9, the proposed method successfully passed all NIST statistical tests with all p-values exceeding 0.01. These results demonstrate strong cryptographic security, high randomness, and robustness regardless of image contents.

Table 9: NIST test of female and airport (2048 × 2048) cipher images.

Test Name	Female (p-Value)	Airport (p-Value)	Pass or Not
Frequency (Monobit) Test	0.732350	0.877610	Pass
Frequency Test within a block	0.999772	0.089707	Pass
Runs Test	0.671779	0.437274	Pass
Test For The Longest Run of ones in a block	0.301992	0.653881	Pass
Binary Matrix Rank Test	0.835327	0.309042	Pass
Discrete Fourier Transform Test (DFT/FFT)	0.602458	0.487554	Pass
Non-Overlapping Template Matching Test	0.465172	0.839615	Pass
Overlapping Template Matching Test	0.362509	0.728615	Pass
Maurer's "Universal Statistical" Test	0.462235	0.910262	Pass
Linear Complexity Test	0.347286	0.616159	Pass
Serial Test p-value 1	0.481291	0.331051	Pass
Serial Test p-value 2	0.246048	0.781671	Pass
Approximate Entropy Test	0.563965	0.939699	Pass
Cumulative Sums (Forward) Test	0.756123	0.955826	Pass
Cumulative Sums (Reverse) Test	0.455041	0.101661	Pass
Random Excursions Test	0.108176	0.034460	Pass
Random Excursions Variant Test	0.040480	0.877610	Pass

6 Comparisons with Other Works

The proposed method outperforms the other methods in encrypting huge images (2048 × 2048), attaining full entropy (8.0), and having a high key space 2^{1140} (10^{343}), demonstrating high robustness against brute force attack. Table 10 shows the comparisons with other state of arts. The proposed method outperforms methods [2,3] in all evaluated aspects; method [5] in all evaluated aspects except the execution time; method [7] in all evaluated aspects; method [8] in all evaluated aspects except the entropy; method [10] in all evaluated aspects except chi-square; method [11,13] in all evaluated aspects; method [14] in all evaluated aspects while MSE and PSNR values of the man image are very similar and have the same entropy; method [15] in all aspects; method [19] in all aspects while boat entropy is identical; method [20] in terms of image size, key space, entropy, and chi-square of the airplane image while MSE values are very similar; for the plant image, in all evaluated metrics except the chi-square values, which are very similar, and the method achieves a shorter execution time; method [21] in all evaluated aspects except the execution time, which is a constraint I wish to address in future work.

Table 10: Comparisons with the other methods.

Method	Image	Image Size	Key Space	Entropy	MSE	PSNR	Speed(s)	ChiSquare
[2]	Mandrill	512 × 512	2^{317}	7.99928	7239.39	9.5337	–	–
[3]	Cameraman	256 × 256	9×10^{159}	7.9968	7796	9.2118	3.319	291.64
[5]	Mandrill	512 × 512	$2^{80} \times 10^{132}$	7.9992	–	–	0.0093	–
[7]	Cameraman	256 × 256	2^{339}	7.9989	–	–	0.22	285.31
[8]	Couple	256 × 256	2^{544}	7.9973	–	7.9709	–	–
[10]	Cameraman	512 × 512	10^{161}	7.9972	–	7.73	–	243
[11]	Cameraman	256 × 256	2^{355}	7.9935	–	–	–	–
[13]	Pentagon	256 × 256	10^{60}	7.9972	–	–	–	251.32
	House			7.9970	–	–	–	269.09
[14]	Mandrill	512 × 512	2^{640}	7.9993	7240.4	9.53	11.6921	271.49
	Man	1024 × 1024		7.9998	10,303	8.00	43.1967	282.68
[15]	Pentagon	256 × 256	$2^{512} \times 10^{180}$ ($=2^{1110}$)	7.9969	6737.82	9.8456	0.5324	278.42
[19]	Mandrill	512 × 512	2^{279}	7.9992	–	–	–	–
	Boat			7.9993	–	–	–	–
[20]	Airplane	256 × 256	2^{318}	7.9970	10,919.2	-	0.0576	276.53
	Plant			–	7735.7	–		253.91
[21]	Cameraman	256 × 256	2^{206}	7.9960	–	–	11 ms	–
Proposed M.	Airplane	256 × 256	2^{1140} (10^{343})	7.9976	10,880.0	7.7644	0.07907	216.57
	Plant			7.9972	7797.6	9.2112	0.07837	255.66
	Couple			7.9970	15,362	6.2663	0.07672	252.79
	House			7.9971	7709.4	9.2606	0.08105	220.43
	Pentagon			7.9974	12,360	7.2107	0.07893	241.81
	Cameraman	256 × 256		7.9971	11,700	7.4488	0.07859	220.00
		512 × 512		7.9994	16,152	6.0487	0.25027	268.72
	Mandrill	512 × 512		7.9993	7268.7	9.5162	0.24802	256.68
	Boat			7.9993	7636.2	9.3020	0.25860	258.44
	Man	1024 × 1024		7.9998	10,280	8.0109	0.88630	247.19
	AirplainU2	2048 × 2048		7.99958	15,142	6.3289	3.71107	239.06

Chi-Square is a measure used to ensure that the distribution of pixels in the cipher image is uniform with respect to the image. The Chi-square is computed as $x^2 = \sum_{i=0}^{255}(O_i - E_i)^2/E_i$ where O_i and E_i denote the observed and expected frequencies of the i-th gray level, respectively, with $E_i = (M \times N)/256$. In an 8-bit image with significance level of 0.05, the critical threshold is 293.24. If the results are less than the critical value, the encryption is excellent, and the pixel distribution is uniform. The lower the value, the better the results, indicating that the data is unpredictable [14]. The proposed method passed the chi-square test and achieved good results between (216–268), lower than the other methods, thus outperforming them. The symbol (–) indicates that the corresponding metric is not applicable, except for the plant image in [20], where the entropy value was not reported.

7 Conclusions

In this study, a novel encoding technique for grayscale images is proposed. A complex, highly random, secure key is created using iterative SHA-512. Then, a perfect 2D chaotic map is used to randomly alter the original image's pixel placements, which is very sensitive to the initial control parameters and the initial states. To improve the quality and clarity of the decrypted image, a median filter is applied to reduce noise and smooth the image. According to the results of the comparison, tables, and figures, the proposed method can encode images of varying sizes and withstand a higher ratio of two forms of noise attacks, but the other system is unable to do so. Fig. 8 shows that the proposed method is very strong against cropping attacks, which resists a 25% cropping ratio. As seen in Table 10, the proposed system outperforms the previous works in terms of key space and entropy, and more resilience to many types of attacks, such as statistical and brute force attacks. The separation of permutation and diffusion stages improves security and increases the difficulty of differential attacks.

A future work aim will focus on applying the proposed method to medical [17] and military imaging applications. Second, it needs to accelerate encryption and decryption times using parallel processing. To further enhance security and performance, it is suggested to replace the symmetric key with a public key-based approach, integrate DNA encoding with chaotic maps with a hashing method to increase diffusion and support color image encryption, adopt additional permutation mechanisms, and extend the proposed method to audio and video encryption. In addition, an intelligent enhancement framework based on Machine Learning (ML) and Deep Learning (DL) techniques will be investigated to improve low-resolution images prior to the encryption process, aiming to enhance image quality while maintaining a high level of security.

Acknowledgement: Not applicable.

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

Availability of Data and Materials: The image datasets used in this study are publicly available and can be accessed from the following sources: USC-SIPI Image Database: https://sipi.usc.edu/database/database.php?volume=misc; Hlevkin Test Image Dataset: https://hlevkin.com/hlevkin/06testimages.htm.

Ethics Approval: Not applicable.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Hameed SM, Taqi IA. A new RGB image encryption based on DNA encoding and multi-chaotic maps. In: New trends in information and communications technology applications. Cham, Switzerland: Springer International Publishing; 2018. p. 69–85. doi:10.1007/978-3-030-01653-1_5.
2. Taqi IA, Abdul-Haleem MG. An efficient cryptosystem for image using 1D and 2D logistic chaotic maps. Int J Intell Eng Syst. 2023;16(4):125–36. doi:10.22266/ijies2023.0831.11.
3. Akraam M, Rashid T, Zafar S. A chaos-based image encryption scheme is proposed using multiple chaotic maps. Math Probl Eng. 2023;2023(1):2003724. doi:10.1155/2023/2003724.
4. Guesmi R, Farah MAB, Kachouri A, Samet M. A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2. Nonlinear Dyn. 2016;83(3):1123–36. doi:10.1007/s11071-015-2392-7.
5. Ben Slimane N, Bouallegue K, Machhout M. A novel image encryption scheme using chaos, hyper-chaos systems and the secure Hash algorithm SHA-1. In: Proceedings of the 2017 International Conference on Control, Automation and Diagnosis (ICCAD); 2017 Jan 19–21; Hammamet, Tunisia. p. 141–5. doi:10.1109/cadiag.2017.8075646.

6. Ahmad M, Al Solami E, Wang XY, Doja MN, Beg MMS, Alzaidi AA. Cryptanalysis of an image encryption algorithm based on combined chaos for a BAN system, and improved scheme using SHA-512 and hyperchaos. Symmetry. 2018;10(7):266. doi:10.3390/sym10070266.
7. Zhu S, Zhu C, Wang W. A new image encryption algorithm based on chaos and secure hash SHA-256. Entropy. 2018;20(9):716. doi:10.3390/e20090716.
8. Xu C, Sun J, Wang C. A novel image encryption algorithm based on bit-plane matrix rotation and hyper chaotic systems. Multimed Tools Appl. 2020;79(9):5573–93. doi:10.1007/s11042-019-08273-x.
9. Shah AA, Parah SA, Rashid M, Elhoseny M. Efficient image encryption scheme based on generalized logistic map for real time image processing. J Real Time Image Process. 2020;17(6):2139–51. doi:10.1007/s11554-020-01008-4.
10. Firdous A, Rehman AU, Missen MMS. A gray image encryption technique using the concept of water waves, chaos and hash function. IEEE Access. 2021;9:11675–93. doi:10.1109/access.2021.3049791.
11. Wu J, Xia W, Zhu G, Liu H, Ma L, Xiong J. Image encryption based on adversarial neural cryptography and SHA controlled chaos. J Mod Opt. 2021;68(8):409–18. doi:10.1080/09500340.2021.1900440.
12. Dua M, Kumar A, Garg A, Garg V. Multiple image encryption approach using non linear chaotic map and cosine transformation. Int J Inf Technol. 2022;14(3):1627–41. doi:10.1007/s41870-022-00885-1.
13. Niu Y, Zhou H, Zhang X. Image encryption scheme based on improved four-dimensional chaotic system and evolutionary operators. Sci Rep. 2024;14(1):7033. doi:10.1038/s41598-024-57756-x.
14. Kumaran S, Ramachandran A. Image encryption using modified perturbed logistic map. Int Arab J Inf Technol. 2024;21(5):866–78. doi:10.34028/iajit/21/5/8.
15. Ernastuti E, Salim RA, Puspitodjati S. Pixel permutation using chaotic perfect shuffle technique for image encryption. TEM J. 2024;13(2):940–9. doi:10.18421/tem132-10.
16. Singh D, Kaur H, Verma C, Kumar N, Illés Z. A novel 3-D image encryption algorithm based on SHA-256 and chaos theory. Alex Eng J. 2025;122(10):564–77. doi:10.1016/j.aej.2025.03.026.
17. Anujaa T, Thajudeen Ali Ahamed AF, Baranwal V, Thanikaiselvan V, Subashanthini S, Sivaranjani Devi C, et al. A lightweight multi round confusion-diffusion cryptosystem for securing images using a modified 5D chaotic system. Sci Rep. 2025;15(1):31986. doi:10.1038/s41598-025-13290-y.
18. Ammar R, Abdullah HA. Performance evaluation of chaotic image encryption. Iraqi J Inf Commun Technol. 2025;8(3):36–49. doi:10.31987/ijict.8.3.304.
19. Zhang X, Hu H. Image encryption algorithm based on a new two-dimensional chaotic system and rotating dial model. Entropy. 2026;28(5):530. doi:10.3390/e28050530.
20. Amutha R, Phamila AV. Chaotic cosine and logistic map based robust image encryption with dual-stage confusion–diffusion architecture. Sci Rep. 2026;16(1):16951. doi:10.1038/s41598-026-40337-5.
21. Kumar A, Jaishree. Image encryption technique using Walsh transform and chaotic systems. Discover Electron. 2026;3(1):47. doi:10.1007/s44291-026-00203-0.
22. Gupta G. Algorithm for image processing using improved median filter and comparison of mean, median and improved median filter. Int J Soft Comput Eng. 2011;1(5):304–11.