Image Encryption Using Modified Perturbed Logistic Map

Subhashini Kumaran
Department of Electronics and Communication Engineering
Sri Sai Ram Engineering College
India
subhashini.ece@sairam.edu.in

Amutha Ramachandran
Department of Electronics and Communication Engineering
Sri Sivasubramaniya Nadar College of Engineering
India
amuthar@ssn.edu.in

Abstract: *The unique properties of chaotic maps have achieved more importance in data protection. The role of the initial parameters in generating the chaotic sequences is vital. In the proposed approach, a new modified perturbed logistic map is suggested. Image encryption is carried out in four steps. The chaotic sequences are attained using the proposed map at first. Secondly, permutation on the pixels of the image is handled by a modified zig zag transformation. Thirdly, permutation effect is further improved by dynamic quaternary DeoxyriboNucleic Acid (DNA) encoding and diffusion using DNA operations. Finally encryption is achieved by exercising DNA decoding. It can be observed from the experiment results of statistical analysis, differential attack analysis, Encryption Quality (EQ) analysis and performance analysis that the proposed scheme surpass existing schemes and can resist different attacks.*

Keywords: *Improved zig zag transformation, image encryption, decoding, DNA operations, modified perturbed logistic map, quaternary DNA encoding.*

Received June 6, 2024; accepted August 26, 2024
https://doi.org/10.34028/iajit/21/5/8

1. Introduction

The increase in use of multimedia images has created a demand for secure transmission. Multimedia has become a vital part of our daily lives in today's rapidly evolving information technology landscape. Digital images have grown in popularity as a form of multimedia data due to their intuitive nature, low level of abstraction and ease of interpretation. With the arrival of the 5G era, image utilization has increased significantly across a diverse range of industries, including the military, medical, and transportation systems. However, as the widespread use of images grow, ensuring their security during transmission and storage has become a top priority.

To address the growing demand for image security, investigators have demonstrated a variety of methods, with image encryption emerging as the most widely used. Chaos-based encryption schemes appear to be a promising approach for safeguarding sensitive image data. One-dimension chaotic map have gained popularity among the various types of chaotic systems due to their simple structure, implementation and low computation. Numerous 1D chaotic maps have been introduced in earlier studies, which examined how well they performed at image encryption. The sine, logistic, sine-tangent map and hybrid maps that combine cubic and exponential functions have all been studied by researchers.

The contribution of this work is:

- A new modified perturbed logistic map is presented which is utilized to originate the chaotic sequences.

The paper is aligned as: In section 2, related work is reviewed. The proposed encryption algorithm is addressed in section 3. The proposed methods' outcome is examined and comparison with already existing methods is also provided in section 4 and the results of the work is concluded in section 5.

2. Related Works

Multi-media data security is a challenging task for the traditional encryption schemes like data encryption standard and advanced encryption standard to encrypt large amount of data at a reasonable speed and provide security. So researchers started exploring different encryption techniques based on chaos, transforms, hybrid chaotic map and Deoxyribonucleic Acid (DNA) coding. Several work on chaotic map has been done using single dimension and multi-dimension.

Ponuma and Amutha [13] performed compression and encryption using a new 1D chaotic map formed by sine, tent and logistic map. Ponuma and Amutha [12] implemented sparse coding method and compressive sensing. The authors attempted a two steps process called scrambling and substitution using the sequences generated by a 1D logistic map. Ponuma *et al*. [15] applied Discrete Wavelet Transform (DWT) and zigzag scan for diffusion. The chaotic sequence is used to build the measurement matrix. Ponuma and Amutha [14] executed compression and encryption using compressive sensing. A combination of chebyshev and tent map was utilized. Ye *et al*. [31] used DWT for sparsification of plain image. The matrix was scrambled by row and then by column using the tent-sine map. DWT and Singular Value Decomposition (SVD) are implemented on the secret image to provide improved security. Ye *et al*. [32] provided initial condition to quantum logistic map using Rivest-Shamir-Adleman (RSA) and scrambling was done by Arnold map. Exclusive-OR (XOR) diffusion and modulo diffusion was applied to each row and column independently. Wang and Zhang [20] determined the scrambling direction of image by using the multiple chaotic systems' coordinates. Two matrices were employed in the diffusion step. Wang *et al*. [22] divided binary image into sub images and permutation by a zigzag method. One dimensional logistic self-embedding map was used for sequence generation. Xian and Wang [27] performed scrambling using a new fractal sorting matrix. Chen's system generated two chaotic sequences for global pixel diffusion using XOR operation. Demirtas [3] constructed a 3D bit plane array and used a Henon map. Alawida [1], developed a new perturbed logistic map, the chaotic range of this map is very large and its behaviour is suitable for confusion and diffusion process.

One dimensional chaotic maps possess limited chaotic range which is a drawback. Many works were proposed using hybrid chaotic maps. Khanzadi *et al*. [7] designed a random generator using logistic and tent chaotic maps. Wang *et al*. [23] demonstrated multiple chaotic maps and a randomized growth approach. Gayathri and Subashini [4] permuted the image pixels by employing circular-shift operations by fetching the key dynamically. Liu *et al*. [8] created a complex chaotic system using Henon map and Chebyshev map. Bit plane substitution and pixel scrambling was achieved by genetic recombination and genetic mutation respectively. Patro *et al*. [11] employed the sorted iterated sequences of the cross-coupled piecewise linear chaotic map for diffusion. Shakiba [16] randomized the permutation and diffusion by a hyper chaotic map. Xu *et al*. [28] implemented a random walk method based on hyper chaotic Lorenz system. The random walk matrix was used for image pixel scrambling and Chen's chaotic system for diffusion operation. Yang *et al*. [30] divided the input image into blocks and normalized. The back propagation network compressed the image and then encrypted using zig zag transformation and fractional order memristive system. Hua *et al*. [5] implemented an adaptive thresholding sparsification and SWT was applied. 2D Cat map was used for confusion process. Man *et al*. [9] executed 5D Hamiltonial hyper chaotic system. This chaotic sequence is the convolution kernel of CNN. Wang and Si [19] used 2D Henon map to achieve scrambling by dynamic L shape and confusion by Arnold map. Diffusion is applied via non-linear operation. Sheela *et al*. [17] performed confusion operation using 5D hyper chaotic system. Wei *et al*. [25] utilized a hyper chaotic system and 2D XOR operation to diffuse the image pixels. Xu *et al*. [29] developed an AI based encryption where DWT was applied on the image, followed by bit plane decomposition. The sequences were generated using Rabinovich hyper chaotic system. Zig zag transformation and then forward and reverse diffusion was performed. Jiang *et al*. [6] proposed a 2D map which improved confusion and diffusion in a random way.

Multi-dimensional chaotic maps are complex in nature and shows some difficulty in execution and their computational time is large. DNA based encryption techniques are being developed to overcome these difficulties. Zhang *et al*. [33] achieved image encryption by applying chaotic system and DNA coding. Wang *et al*. [24] manipulated the DNA sequences and chaotic systems. Bitwise XOR operation was applied using coupled map lattice. Encoding of the confused image was done using a specific DNA encode rule and converted into a DNA array. Patel *et al*. [10] combined 3D chaotic logistic map and DNA encoding. The algorithm employed three keys viz. ASCII key of length 32 bit, a chaotic key using Chebyshev polynomial and a prime key. Wang and Zhao [21] performed scrambling in blocks and DNA coding. The Chen hyper chaotic map was used to provoke the chaotic sequence. Scrambling and then diffusion is done through different DNA coding rules. The cipher was achieved by the process of DNA matrix decoding. Zhang *et al*. [34] combined image signature, bit level analysis and dynamic DNA coding. The Hashing algorithm was used for initial value of chaotic mapping and a 3D chaotic map was used for bit level partitioning. Numerical substitution was followed by dynamic DNA coding. Wang *et al*. [18] illustrated a hybrid chaotic map with extensive range of chaotic behaviour better than Logistic and Sine map. The time taken to generate the chaotic sequence was more. The encryption algorithm used an improved zig zag transformation and a quaternary DNA coding.

Al-Hazaimeh [2] used two different chaotic maps for encryption and decryption called compound chaotic system. New chaotic maps that address these issues is needed to overcome the existing drawbacks. Additionally, encryption algorithms depend on the image pixel permutation. One of the permutation methods, the standard zig zag transformation, faces few drawbacks, including periodicity issues, could not alter the positions of specific pixel values in matrix as a result could not be used for non-square matrices. Researchers have suggested improved zig zag transformation that provide greater flexibility and superior permutation effects to overcome these limitations. DNA coding, an innovative strategy, makes use of the similarities between DNA chains and cypher sequences. However, challenges remain, notably in improving the efficiency of DNA coding-based encryption techniques.

3. Proposed Encryption Algorithm

The proposed scheme carries out encryption by generating the key using proposed modified perturbed logistic map, pixel transformation by modified zig zag method, dynamic quaternary DNA coding and operations followed by DNA decoding.

3.1. Key Generation

The proposed modified perturbed logistic map is given by the Equation (1).

$$Y_{j+1} = abs\left(\left(\mu * \left(1-\frac{\mu}{Y_j}\right)(1-\mu^2)\right) mod\ 1\right) \qquad (1)$$

where μ is the control parameter and Y_j is the initial value and takes values between 0 and 1. The bifurcation diagram of the proposed map is shown in Figure 1. The parameter μ is varied from 0 to 100. The chaotic region of the proposed map is very large when compared to logistic map, as a result the key space can be improved.

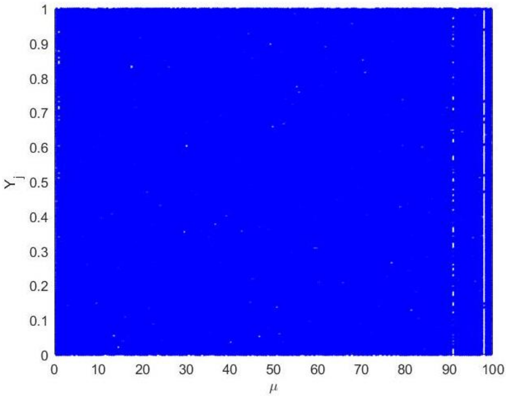


Figure 1. Bifurcation plot of proposed map.

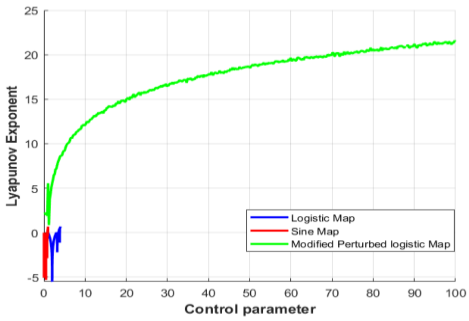


Figure 2. Lyapunov plot of logistic, sine and modified perturbed logistic map.

The Lyapunov exponent must be positive for all control parameter values of a chaotic map. The proposed map gives a positive lyapunov exponent for control parameter values in the range 0 to 100. This exhibits the sensitivity to initial condition. The lyapunov exponent of logistic and sine map is compared with the proposed map as shown in Figure 2.

128 bits are randomly generated. Out of 128 bits 52 bits are used to generate the control parameter and significand bits are only considered and Equation (2) is used to find the control parameter using the bits 1 to 52.

$$\mu = \sum_{t=1}^{52} \frac{Rand_{bit}}{2^t} \qquad (2)$$

The initial value is generated by utilizing the bits 53 to 102 as follows:

$$y_0 = \frac{\sum_{t=53}^{102} \frac{Rand_{bit}}{2^t} + \mu}{2} \qquad (3)$$

where $Rand_{bit}$ is the value of position t. Similarly, another four keys (128 bits) are randomly generated. Five chaotic sequences Y_1, Y_2, Y_3, Y_4, and Y_5 of size 4pq are generated using the same procedure. pxq is the size of the image. For 256x256 p=256; q=256.

3.2. Modified Zig Zag Transformation

The modified zig zag transformation utilizes scanning in two directions as shown in Figure 3. In the first process, the scanning starts from first pixel value (34 in the Figure 3) of the matrix. It is continued till half the value of the size of the matrix. The first value is kept as it is, the previous value is XORed with the present value to change the pixels. The second process starts from the last matrix value and eXclusive-NOR (XNOR) operation is performed. The third process is shuffling the vector 1 & 2 obtained from process one and two by placing them in even and odd position respectively. Three iterations are performed to permute the pixels completely.

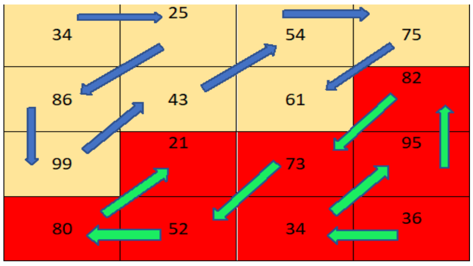


Figure 3. Modified zig zag transformation.

3.3. Quaternary DNA Coding

In traditional DNA coding, the complementarity is defined by the Watson-Crick base pairing rules, which state the combination as adenine (a)-thymine (t) and cytosine (c)-guanine (g). In the formation and function of DNA the complementary base pairing rule is fundamental. a, g, c, t are represented as 0,1,2,3 respectively [34]. A dynamic rule is provoked to encode the quaternary number and decoded using another rule. Table 1 provides the 8 possible DNA coding and decoding rules [18].

Table 1. DNA coding and decoding rules.

Quaternary DNA	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7	Rule8
0	0-a	0-a	0-g	0-g	0-c	0-c	0-t	0-t
1	1-g	1-c	1-t	1-a	1-t	1-a	1-g	1-c
2	2-c	2-g	2-a	2-t	2-a	2-t	2-c	2-g
3	3-t	3-t	3-c	3-c	3-g	3-g	3-a	3-a

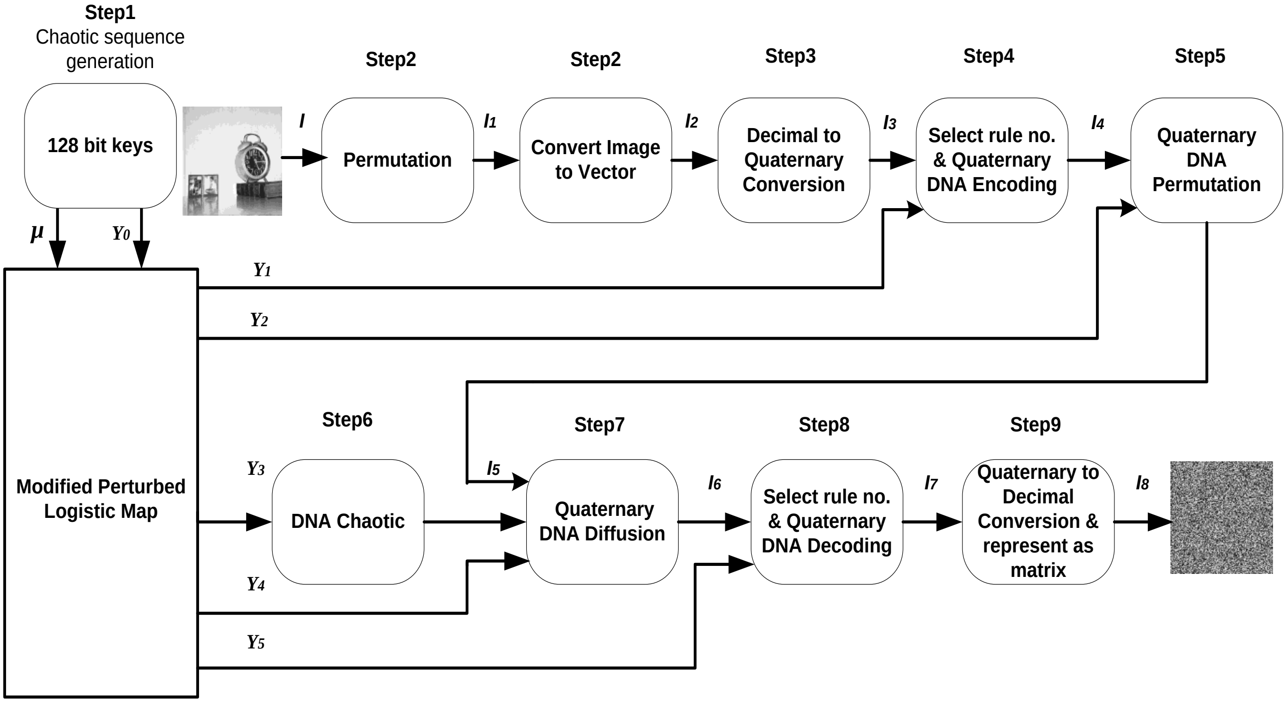

Step1
Chaotic sequence generation
128 bit keys
μ
Y0
Modified Perturbed Logistic Map
Y1
Y2
Y3
Y4
Y5
I
Step2
Permutation
I1
Step2
Convert Image to Vector
I2
Step3
Decimal to Quaternary Conversion
I3
Step4
Select rule no. & Quaternary DNA Encoding
I4
Step5
Quaternary DNA Permutation
I5
Step6
DNA Chaotic
Step7
Quaternary DNA Diffusion
I6
Step8
Select rule no. & Quaternary DNA Decoding
I7
Step9
Quaternary to Decimal Conversion & represent as matrix
I8

Figure 4. Proposed encryption technique.

The quaternary DNA operations are performed as found in the Tables 2, 3, 4, and 5.

Quaternary DNA addition operation using rule 7 is shown in Table 2. For example in rule 7, 0⟷*t*; 1⟷*g*; 2⟷*c*; 3⟷a fourth column of Table 2 is obtained by adding first column values with 2 that is 0+2=2, and the result 2 is represented as '*c*' according to rule 7. 1+2=3 and the result 3 is represented as '*a*' according to rule 7.

Table 2. Quaternary DNA addition using rule 7.

Add	0-t	1-g	2-c	3-a
0-t	t	g	c	a
1-g	g	c	a	t
2-c	c	a	t	g
3-a	a	t	g	c

Results of quaternary DNA subtraction using rule 7 is listed in Table 3. For example, second column of Table 3 is obtained by subtracting 0 from first column values, 0-0=0 and result is represented as '*t*' according to rule 7. 1-0=1 and result is represented as '*g*' according to rule 7.

Table 3. Quaternary DNA subtraction using rule7.

Sub	0-t	1-g	2-c	3-a
0-t	t	a	c	g
1-g	g	t	a	c
2-c	c	g	t	a
3-a	a	c	g	t

Table 4. Quaternary DNA XOR using rule 7.

XOR	0-t	1-g	2-c	3-a
0-t	t	g	c	a
1-g	g	t	a	c
2-c	c	a	t	g
3-a	a	c	g	t

Results of quaternary DNA-XOR using rule 7 is listed in Table 4. For example, third column of Table 4 is obtained by XORing first column values with 1, $0\oplus1=1$ and result is represented as '*g*' according to rule 7. $1\oplus1=0$ and result is represented as '*t*' according to rule 7.

Results of quaternary DNA eXclusive-NOR (DNA-XNOR) using rule 7 is listed in Table 5. For example, fifth column of Table 5 is obtained by XNORing first column values with 3, $0\odot3$ and result is represented as '*t*' according to rule 7. $1\odot3=1$ and result is represented as '*g*' according to rule 7.

Table 5. Quaternary DNA-XNOR using rule 7.

XNOR	0-t	1-g	2-c	3-a
0-t	a	c	g	t
1-g	c	a	t	g
2-c	g	t	a	c
3-a	t	g	c	a

3.4. Algorithm

The block diagram of the image encryption is shown in Figure 4.

- *Step* 1. Chaotic sequence generation. Generate five chaotic sequences of size 4pxq using the proposed modified perturbed logistic map, where pxq is the size of the image.
- *Step* 2. Permutation and convert image to vector. Apply modified zig zag transformation to permute the pixels of image I to get I_1. The resultant matrix is arranged as a vector I_2.
- *Step* 3. Decimal to quaternary conversion. Convert the pixel values in I_2 into its equivalent quaternary representation to get I_3. The vector dimension of I_3 is 4pq.

Conversion of decimal to quaternary.

Consider the permuted pixels of I_2 vector as [12, 5, 8, 22]. The decimal values are converted into quaternary (base 4) as shown in Figure 5.

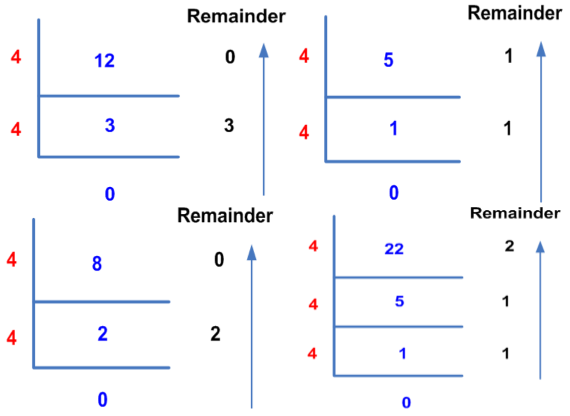


Figure 5. Decimal to quaternary conversion process.

Quaternary equivalent of 12, 5, 8 and 22 is 30, 11, 20 and 112 respectively. The quaternary values are represented as [0030, 0011, 0020, 0112], using 4 digits.

Concatenation of these values gives the vector I_3 as [0 0 3 0 0 0 1 1 0 0 2 0 0 1 1 2].

- *Step* 4. Select rule no. and quaternary DNA encoding. Select anyone of the eight DNA encoding rule dynamically to encode vector I_3using the following equation to get I_4 of dimension 4pq.

$$DNAencoding_{rule}(j) = floor(8 * Y_1(j)) + 1, \quad j = 1,2,\dots,4pq \tag{4}$$

- *Step* 5. Quaternary DNA permutation. Sort Y_2 in increasing order and procure the index number which is used to permute I_4 and get vector I_5 of size 4pq.

$$I_5(j) = I_4(Sorted_{index}(j)), \quad j = 1,2,\dots,4pq \tag{5}$$

- *Step* 6. DNA Chaotic. Convert Y_3 into $Y_{3DNAchaotic}$ using the following equations,

$$\left.\begin{array}{r} if\ 0 \le Y_3 < 0.25, then\ Y_{3DNAchaotic} = 0 \\ if\ 0.25 \le Y_3 < 0.5, then\ Y_{3DNAchaotic} = 1 \\ if\ 0.5 \le Y_3 < 0.75, then\ Y_{3DNAchaotic} = 2 \\ if\ 0.75 \le Y_3 < 1, then\ Y_{3DNAchaotic} = 3 \end{array}\right\} \tag{6}$$

- *Step* 7. Quaternary DNA diffusion. Select one of the four DNA functions dynamically using the following rule to perform diffusion and obtain vector I_6.

$$DNAoperation_{rule}(j) = floor(4 * Y_4(j)) + 1, \tag{7}$$

$$\left.\begin{array}{r} if\ DNAoperation_{rule}(j) = 1, then\ Add(I_5(j), Y_{3DNAchaotic}(j)) \\ if\ DNAoperation_{rule}(j) = 2, then\ Sub(I_5(j), Y_{3DNAchaotic}(j)) \\ if\ DNAoperation_{rule}(j) = 3, then\ XOR(I_5(j), Y_{3DNAchaotic}(j)) \\ if\ DNAoperation_{rule}(j) = 4, then\ XNOR(I_5(j), Y_{3DNAchaotic}(j)) \end{array}\right\} = I_6(j) \tag{8}$$

- *Step* 8. Select rule no. and quaternary DNA decoding: Generate the DNA decoding rule using Equation (9) to decode vector I_6 to I_7

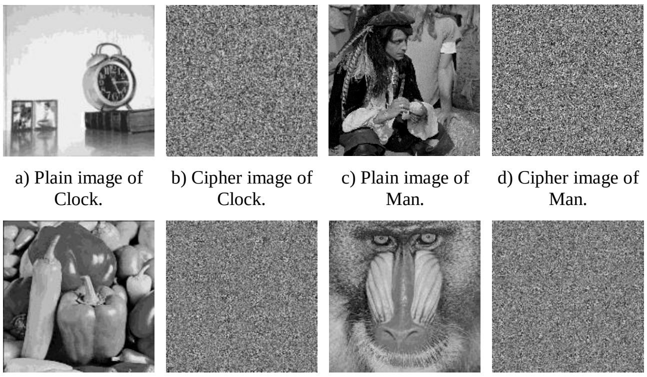
$$DNAdecoding_{rule}(j) = floor(8 * Y_5(j)) + 1, \quad j = 1,2,\dots,4pq \tag{9}$$

For example, after diffusion operation the vector I_6=[*t*, *g*, *a*, *t*, *c*, *c*, *t*, *t*, *c*, *t*] is obtained and the decoding rule vector $DNAdecoding_{rule}(j)$= [2, 4, 6, 7, 1, 3, 5, 8, 2, 4] is generated using Equation (9). The decoded vector is $I_7(j)$=[*a*, *a*, *g*, *t*, *c*, *a*, *c*, *t*, *c*, *g*]. Convert $I_7(j)$ into its quaternary value [3, 3, 1, 0, 2, 3, 2, 0, 2, 1].

- *Step* 9. Quaternary to decimal conversion and represent as matrix. Convert vector I_7 into its equivalent decimal value as I_8. Group $I_7(j)$ into a group of 4 values. For example [3, 3, 1, 0, 2, 3, 2, 0, 2, 1] is divided into [3, 3, 1, 0], [2, 3, 2, 0]. Convert each group into its equivalent decimal value to generate the vector $I_8(j)$=[244, 184, …]. Construct vector I_8 as a matrix which results in the cipher image.

4. Simulation Results and Discussions

Simulation results are obtained using images from University of Southern California-Signal and Image Processing Institute (USC-SIPI) database. Grayscale images of size 256x256, 512x512 and 1024x1024 are used. The images used in the simulations are Clock, Peppers, Baboon and Man. Image encryption is done using three maps, namely proposed new modified perturbed logistic map, Hybrid Logistic Sine Exponential (HLSE) map [18] and perturbed logistic map [1]. The performance of the image encryption algorithm is verified using security analysis, encryption analysis, statistical test and differential attack analysis. The encryption algorithm is executed using a system with AMD 5600H CPU @3.3GHz and 8 GB RAM, PyCharm Integrated Development Environment (IDE) and MATLAB R2022a. The input and the respective cipher obtained by applying the proposed algorithm are shown in Figure 6.



a) Plain image of Clock. b) Cipher image of Clock. c) Plain image of Man. d) Cipher image of Man.

e) Plain image of Peppers. f) Cipher image of Peppers. g) Plain image of Baboon. h) Cipher image of Baboon.

Figure 6. Plain, Cipher images for each one of these: Clock, Man, Peppers and Baboon.

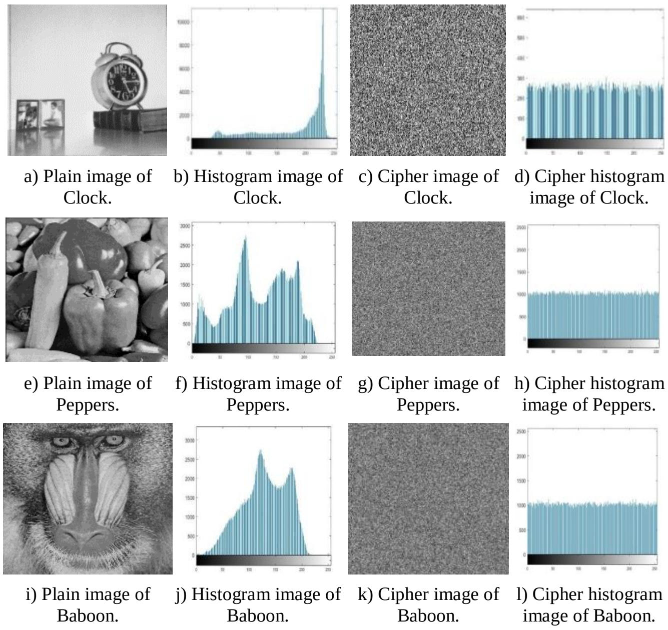
4.1. Statistical Analysis

Statistical attacks happen due to the correlation that exists among the neighboring pixels. The proposed technique is assessed using statistical analysis like histogram, correlation, entropy analysis, chi-square test, National Institute of Standards and Technology (NIST) test and Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE) analysis in the next section.

4.1.1. Histogram Analysis

Histogram illustrates the pixel frequency distribution. The histogram of plain and cipher is presented in Figure 7.

It is observed that there exists an uneven distribution of image pixels in the plain histogram and it is prone to statistical attacks. The encrypted image exhibit even distribution for all test images and hence the proposed technique can withstand such attacks. Moreover, the snooper could not pry input from cipher.



a) Plain image of Clock. b) Histogram image of Clock. c) Cipher image of Clock. d) Cipher histogram image of Clock.

e) Plain image of Peppers. f) Histogram image of Peppers. g) Cipher image of Peppers. h) Cipher histogram image of Peppers.

i) Plain image of Baboon. j) Histogram image of Baboon. k) Cipher image of Baboon. l) Cipher histogram image of Baboon.

Figure 7. Plain, histogram, cipher, cipher histogram images for each one of these: Clock, Peppers, and Baboon.

4.1.2. Correlation Analysis

The correlation between neighbouring pixels of an image can be found using the following equations:

$$R_{xy} = \frac{cov(x,y)}{\sqrt{A(x)A(y)}} \tag{10}$$

where,

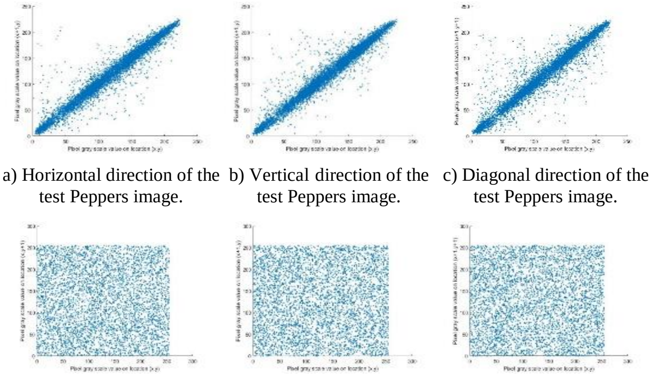
$$cov(x,y) = \frac{1}{N}\sum_{j=1}^{N}(x_j - E(x))(y_j - E(y)) \tag{11}$$

$$A(y) = \frac{1}{N}\sum_{j=1}^{N}(y_j - E(y))^2 \tag{12}$$

The correlation coefficient is measured in horizontal, vertical and diagonal direction by arbitrarily selecting 10000 pixels. Table 6 provides the correlation values of the peppers image for the proposed and existing methods. The values are approximately equal to 1 for plain image and ascertain high correlation. The correlation values of the cipher obtained for the proposed algorithm are less than '0.01' indicating that the correlation is minimum and close to '0'. The value of correlation coefficient of the proposed method in horizontal and vertical direction is '-0.018' and '-0.0109' respectively, which is lesser compared to the results of [16, 18, 28, 32]. Figure 8-a), (b) and (c) depicts the pixel correlation plot of peppers image in three directions. The pixels are grouped in the diagonal direction and infers very strong correlation. The scatter plot of the cipher image displayed in Figure 8-d), (e) and (f) is sporadic and combat attacks based on correlation.

Table 6. Correlation analysis of Peppers image

Algorithm	Horizontal	Vertical	Diagonal
Plain image	0.9721	0.9767	0.9651
[18]	0.0003	0.0003	-0.0034
[1]	-0.0064	-0.0003	0.0061
Proposed	-0.018	-0.0109	0.0013
[10]	-0.0022	-0.0016	0.0046
[32]	0.0139	0.0054	0.0153
[11]	- 0.0013	0.0019	0.0009
[16]	0.002553	−0.0138	−0.0028
[28]	0.0084	0.0045	−0.0023



a) Horizontal direction of the test Peppers image. b) Vertical direction of the test Peppers image. c) Diagonal direction of the test Peppers image.

d) Horizontal direction of the cipher Peppers image. e) Vertical direction of the cipher Peppers image. f) Diagonal direction of the cipher Peppers image.

Figure 8. Pixel correlation plot of test and cipher for image Peppers in a horizontal, vertical, diagonal directions.

4.1.3. Global Entropy Analysis

Information entropy, used to find the uncertainty present in cipher image. Entropy value of a cipher is ideally equal to 8. The entropy of the test and cipher image are calculated using the following equation:

$$H(z) = \sum_{i=0}^{L} p(GV_i) log_2 \frac{1}{p(GV_i)} \tag{13}$$

$p(GV_i)$is the probability of the image pixels and L is the maximum pixel value.

Table 7. Global entropy.

Test image	Plain image	Cipher image		
		[18]	**[1]**	**Proposed**
Clock 256x256	6.7057	7.9971	7.9973	7.9973
Peppers 512x512	7.5937	7.9992	7.9991	7.9993
Man 1024x1024	7.5237	7.9998	7.9998	7.9998

It is inferred from Table 7 that the cipher image entropy values for test images of different dimension are very close to 8. The average entropy 7.9989 of the proposed approach is close to ideal value.

Table 8. Comparison of global entropy of Baboon image.

Test image	Plain image	Cipher image					
		[18]	[1]	Proposed	[23]	[24]	[33]
Baboon 512x512	7.3583	7.9927	7.9993	7.9993	7.9974	7.9969	7.9992

The entropy value of the Baboon image of the proposed algorithm is compared with other algorithms in the literature and is listed in Table 8. The proposed approach gives 7.9993, a high entropy value compared to existing methods.

Local entropy is used to test the randomness of encrypted image. The global Shannon entropy measure is inconsistent for images of varying dimension and the local entropy measure provides a relatively fair comparison for image randomness among multiple images.

$$H_{k,TB}(B)=\sum_{i=1}^{k}\frac{H(B_i)}{k} \tag{14}$$

where k is the number of non-overlapping blocks and $H(B_i)$ is the Shannon entropy. The image is divided into k blocks and each block consist of TB pixels. For k=30 and TB=1936 the local entropy is calculated. The average entropy of all blocks gives the local entropy [26].

Table 9. Local entropy analysis.

Test image	[18]	[1]	Proposed
Clock 256x256	7.8998	7.9059	7.9025
Baboon 512x512	7.9041	7.903	7.9042
Peppers 512x512	7.9021	7.9033	7.9035
Man 1024x1024	7.8806	7.9046	7.9021

The test results in Table 9 shows that the proposed method is equal to the ideal value (7.902469317) and passes the test for the images clock, baboon and peppers with a significance level α=0.05.

4.1.4. Chi-Square Test

This test provides the pixel uniformness of the cipher image. The chi-square value is calculated by the equation,

$$x^2=\sum\nolimits_{i=0}^{255}\frac{(ob-ex)^2}{ex} \tag{15}$$

The observed value 'ob' is the count of each gray value has occurred in the cipher image. The expected value 'ex' is given by,

$$expected\ value, ex=\frac{pxq}{256} \tag{16}$$

where p and q is the value of available rows and columns in the image and expected value of a 256x256 image is 256.

The critical chi-square value is $\varkappa^2$=293.2478 for 255 degrees of freedom at 5% significance level or α=0.05 [10]. For a gray scale image the degrees of freedom is related to the pixel intensity values. The test results indicate in Table 10 that all cipher are less than the critical value. This proves that the histogram of the cipher holds uniform distribution and the algorithm works well with all three maps.

Table 10. Chi-square test.

Test image	[18]	[1]	Proposed
Clock 256x256	261.0859	248.6172	242.5156
Baboon 512x512	254.6895	253.5898	271.4922
Man 1024x1024	245.481	286.3892	282.6855

Chi-square value of the peppers image for the proposed algorithm and other algorithms in the literature are shown in Table 11. The Chi-square value of the peppers image using the proposed method is 227.32 which is 19% lesser than 264.77 as reported in [10]. The chi-square value of the proposed method is lesser than [4, 11, 20].

Table 11. Comparison of chi-square value of Pepper image.

Image	[18]	[1]	Proposed	[10]	[20]	[11]	[4]
Peppers	275.07	310.67	227.32	264.77	287.21	271.9	244.49

4.1.5. NIST Test

NIST is computational series of 15 test that validates the randomness of the cipher image by using their binary input to test for randomness and to determine its p-value. It determines whether the sequence is predictable, if the value is less than or equal to 0.01 it fails the test. NIST-Special Publication (NIST-SP) 800-22 revision 1a test suite is used to carry out the test. Table 12 gives the 15 NIST test of the proposed modified perturbed logistic method for the clock cipher image. The test results show that the proposed method passes all the NIST test and has p-value>0.01.

Table 12. NIST test results for Clock cipher image.

Test	Proposed	
	P-value	Result
Frequency	0.324931	Pass
Block frequency	0.199739	Pass
Runs	0.044259	Pass
Longest	0.146922	Pass
Rank	0.962089	Pass
FFT	0.019060	Pass
Non-overlapping template	0.809319	Pass
Overlapping template	0.631976	Pass
Linear complexity	0.281564	Pass
Serial test P-value 1	0.525440	Pass
Serial test P-value 2	0.109853	Pass
Approximate entropy	0.656514	Pass
Cumulative sums-forward	0.604405	Pass
Cumulative sums-reverse	0.529498	Pass
Random excursions test (X=1)	0.443414	Pass
Random excursions test (X=-1)	0.340473	Pass
Random excursions variant test (X=1)	0.425030	Pass
Random excursions variant test (X=-1)	0.732439	Pass

4.1.6. PSNR and MSE Analysis

MSE and PSNR are calculated between the plain and the cipher image to check for any relation between them. MSE finds the average squared pixel difference corresponding to plain and cipher, higher the value the greater is the encryption effect. If the PSNR is low, superior is the Encryption Quality (EQ).

MSE is defined as:

$$MSE = \frac{1}{pxq}\sum_{k=1}^{p}\sum_{l=1}^{q}(C_1(k,l) - C_2(k,l))^2 \qquad (17)$$

$C_1(k, l)$ and $C_2(k, l)$ are the individual pixels of plain and cipher respectively.

PSNR is defined as:

$$PSNR = 10 * log_{10}(\frac{255^2}{MSE}) \qquad (18)$$

where 255 is the maximum pixel value.

Table 13. MSE values.

Test image	[18]	[1]	Proposed
Clock (256x256)	12073	12111.5	12101.67
Baboon (512x512)	7235	7276.58	7240.42
Man (1924x1024)	10312.99	10298.62	10303.05

The MSE is computed for different test images as shown in Table 13. The mean square value of the proposed approach is higher than the HLSE method and lesser than the perturbed logistic method. This indicates that the proposed map produces equivalently large pixel differences between plain and cipher as HLSE map and perturbed logistic map.

Table 14. Comparison of MSE of Peppers image.

Image	[18]	[1]	Proposed	[4]	[16]	[11]
Peppers	8408.1	8400.34	8389.80	8260	5413.9	8465.8

The MSE value of the peppers image for the proposed modified perturbed logistic method is compared with the other algorithm and is shown in Table 14. It is inferred that the MSE values of the proposed method is slightly less than HLSE and perturbed logistic method. The MSE value of the proposed algorithm for peppers image is 1.56% and 43.12% higher than [4, 16] respectively. This shows EQ is high.

Table 15. PSNR values.

Test image	[18]	[1]	Proposed
Clock 256x256	7.3125	7.30	7.30
Baboon 512x512	8.8838	9.51	9.53
Man 1024x1024	9.5364	8.00	8.00

Table 15 indicates that the PSNR of the proposed approach is approximately same as that of perturbed logistic method and 2.614% less than the HLSE method. The result also illustrates that for different image dimension (256x256, 512x512 and 1024x1024) PSNR is less than 10dB for all three maps, indicating high EQ.

Table 16. Comparison of PSNR of Peppers image.

Test image	[18]	[1]	Proposed	[4]	[16]	[11]
Peppers	8.88	8.89	8.89	9.0234	10.7957	8.8541

Test results in Table 16 shows the PSNR value for peppers image. The PSNR of the modified perturbed logistic method is 1.49%, 19.36% less than [4, 16]. MSE and PSNR are contrarily equivalent to each other. An increase in MSE value decreases the PSNR. As the PSNR is less than 10dB, it is not possible to retrieve the plain image from the cipher.

4.2. Differential Attack Analysis

The encryption algorithm must yield a totally different encrypted image even there is only one pixel deviation in the plain image. The sensitivity of the algorithm can be proved by finding Number of Pixel Change Rate (NPCR) and Unified Average Change Intensity (UACI) values using the Equations (19) and (21) respectively.

$$NPCR = \frac{1}{pxq}\sum_{j,k} D(j,k) * 100\% \qquad (19)$$

where $D(j,k) = \begin{cases} 0, \ I_1(j,k) = I_2(j,k) \\ 1, I_1(j,k) \neq I_2(j,k) \end{cases} \qquad (20)$

The unified average change intensity is calculated using the equation,

$$UACI = \frac{1}{pxq}\frac{\sum_{j,k}|I_1(j,k) - I_2(j,k)|}{max} * 100\% \qquad (21)$$

where I_1 and I_2 are the two ciphers of the plain images having one pixel difference and ‘max’ stands for maximum intensity of the pixel. The one-pixel difference in plain image is achieved by arbitrarily selecting a pixel value and adding one to it.

Table 17. NPCR values.

Test image	[18]	[1]	Proposed
Clock 256x256	99.5697	99.5926	99.6338
Baboon 512x512	99.6273	99.6201	99.6178
Man 1024x1024	99.6126	99.6070	99.6069

Test results in Table 17 renders the NPCR values for images of different dimension and the proposed results are above critical value. The average NPCR of the proposed algorithm is 99.6148, which is higher than [1, 18]. The sensitivity of the proposed encryption algorithm to one pixel variation is proved. This emphasizes the protection against differential attacks.

Table 18. Comparison of NPCR values of Peppers image.

Test image	[18]	[1]	Proposed	[32]	[16]	[20]
Peppers	99.6075	99.6307	99.6201	99.6086	99.6067	99.6108

Test results in Table 18 shows NPCR of proposed algorithm for peppers image is less than perturbed logistic method and higher than all other methods. For image dimension 256x256, 512x512 and 1024x1024, the critical NPCR is 99.5693%, 99.5893% and 99.5994% respectively. The proposed approach can withstand the differential attack as the NPCR is above the critical value for all test images.

Table 19. UACI values.

Test image	[18]	[1]	Proposed
Clock 256x256	33.5536	33.4741	33.5384
Baboon 512x512	33.5232	33.4265	33.5468
Man 1024x1024	33.4592	33.4624	33.4806

Test results in Table 19 reveals that the proposed algorithm gives highest UACI value for different plain text. The average value of the proposed method 33.5125 is also better than other two methods. This ensures that the modified perturbed logistic method produces a totally altered cipher even for a one pixel variation.

Table 20. Comparison of UACI values of Peppers image.

Test image	[18]	[1]	Proposed	[32]	[16]	[20]
Peppers	33.4526	33.4508	33.5203	33.4398	33.4332	33.5173

The UACI of peppers image is compared with [16, 20, 32] in Table 20 and the proposed method shows highest value. The critical UACI value can lie in the range (33.2824%, 33.6447%), (33.3730%, 33.5541%) and (33.4183%, 33.5088%) for image dimension 256x256, 512x512 and 1024x1024 respectively.

4.3. Encryption Quality Analysis

Maximum Deviation (MD) is a metric that is use find the deviation of density of pixel values between cipher and plain image. If the MD value is high, it implies that there is high deviation between the pixel density of cipher and plain image. EQ is the metric which evaluates the pixel intensity deviation between the original and cipher.

4.3.1. Maximum Deviation

MD can be defined as:

$$Max\ Deviation = \frac{g_0 + g_{255}}{2} + \sum_{i=1}^{254} g_i \qquad (22)$$

where g_i is given as $|I_i\text{-}C_i|$ where I_i and C_i are each pixel value frequency in plain and cipher image, g_0 and g_{255} stands for number of pixels having 0 and 255 as grayscale value respectively.

Table 21. Comparison of MD values.

Test image	[18]	[1]	Proposed
Clock 256x256	61678	61701	61087
Peppers 512x512	143232.5	142866	144127.5
Man 1024x1024	576981.5	577334.5	578517

MD value of proposed algorithm is compared in Table 21 with HLSE method and perturbed logistic method.

Table 22. Comparison of MD for Baboon image.

Test image	[18]	[1]	Proposed	[20]
Baboon	196214	196590	197185	46540

Test results in Table 22 indicates that the proposed modified perturbed logistic map gives high MD value for Baboon image compared to other methods. The MD of Baboon image calculated using the proposed method is 123% higher than that of [20].

4.3.2. Encryption Quality (EQ)

EQ is represented by,

$$Encryption\ Quality\ EQ = \sum_{i=0}^{255} \frac{g_i}{256} \qquad (23)$$

Table 23. EQ values for different test images.

Test image	[32]	[11]	Proposed
Clock 256x256	242.9063	242.9922	240.6484
Baboon 512x512	774.90	775.84	778.1641
Man 1024x1024	2328.5	2329.7	2333.8

Test results in Table 23 unveils that the EQ values of proposed computation method is greater than HLSE method and perturbed logistic method for Baboon and Man images. High EQ values confirm the EQ of the cipher.

Table 24. Comparison of EQ for Peppers image.

Test image	[18]	[1]	Proposed
Peppers	568	565.96	570.9531

It is inferred from Table 24 that the proposed modified perturbed logistic method gives higher value of EQ compared with other algorithms.

4.3.3. Irregular Deviation

The Irregular Deviation (ID) is a metric used to measure the irregularity or non-uniformity of pixel distribution in a cipher image histogram. The ID value is used to evaluate the evenness of pixel distribution in the cipher image. A lower ID value indicates a more uniform or even distribution of pixels in the image, whereas a higher ID value indicates a more irregular or non-uniform distribution.

$$Irregular_{Deviation} = \sum_{i=0}^{255} |g_i - \gamma_h| \qquad (24)$$

where γ_h is the histogram average.

Table 25. Comparison of ID.

Test image	Proposed	[18]	[1]
Clock 256x256	32881	32802	32807

The results infer that the proposed method produced equivalently comparable result as that of existing methods as shown in Table 25.

4.3.4. Deviation from Uniform Histogram

The deviation from uniform histogram measures the discrepancy between the cipher histogram and an ideal uniform histogram.

$$Image\ histogram\ I_h = \begin{cases} \frac{1}{256}\ pxq, 0 \le i \le 255 \\ 0, \quad elsewhere \end{cases} \qquad (25)$$

$$Deviation_{histogram} = \frac{\sum_{h_i=0}^{255} |I_{h_i} - I_h|}{pxq} \qquad (26)$$

where I_{h_i} is the cipher image gray scale value at position i. The $Deviation_{histogram}$ should be less to indicate that the distribution is uniform.

The results infer a minimum deviation and thereby proves that the pixels are uniformly distributed in the proposed method as indicated in Table 26.

Table 26. Comparison of DUH.

Test image	Proposed	[18]	[1]
Clock 256x256	0.04965	0.05035	0.04925
Peppers 512x512	0.0240	0.0301	0.0276
Baboon 512x512	0.0252	0.0299	0.0247
Man 1024x1024	0.0130	0.0131	0.0131

4.3.5. Contrast Analysis

In contrast analysis the variation in gray level is used to find patterns. The analysis uses co-occurrence matrix, *CoOccurrence* (*i*, *j*) to identify specific patterns in the image.

$$Contrast = \sum_{i=0}^{256}\sum_{j=0}^{256} |i-j|^2\, CoOccurrence(i,j) \qquad (27)$$

where *i* and *j* represent 8 bit gray levels.

Table 27. Contrast analysis for different test images.

Test image	[18]	[1]	Proposed
Clock 256x256	10910.469	10855.3661	10876.5668
Peppers 512x512	10863.654	10842.007	10861.8307
Baboon 512x512	10882.011	10890.3316	10846.965
Man 1024x1024	10919.678	10887.0349	10908.0449

The contrast analysis of proposed method shown in Table 27 infers comparable performance with that of existing techniques.

4.3.6. Key Space Analysis

The key space of the proposed method for five 128 bit keys is $2^{128}x2^{128}x2^{128}x2^{128}x2^{128}$ which is $2^{640} > 2^{100}$ and sufficient to resist brute force attack. The larger key space increases ability of the algorithm to prevent intensive attacks.

Table 28. Key space analysis.

[18]	[1]	Proposed
10^{224}	2^{128}	2^{640}

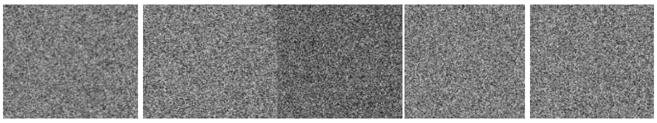
Test results in Table 28 compares the key space of the proposed modified perturbed logistic method and other methods in the literature.

4.3.7. Key Sensitivity Analysis

Key sensitivity is a phenomenon where even slight modifications in encryption or decryption keys results in significant changes in the resulting ciphertext or plaintext. Key sensitivity analysis involves generating two distinct keys:

Key$_1$=504b444ea02e166342b1e53a5aad76f87b0cfda
Key$_2$=504b444ea02e166342b1e53a5aad76f87b0ceda.

Key$_1$ and *Key*$_2$ are used to encrypt the original image, yielding two distinct cipher images, *CI*$_1$ and *CI*$_2$ as shown in Figure 9. To quantify the visual dissimilarity caused by key divergence, the absolute difference between *CI*$_1$ and *CI*$_2$ is computed. Moreover, employing *Key*$_2$ to decrypt cipher image *CI*$_1$ and *Key*$_1$ for cipher image *CI*$_2$ provides insights into the impact of key interchange on decryption outcomes. This shows that the algorithm is highly sensitive to keys.



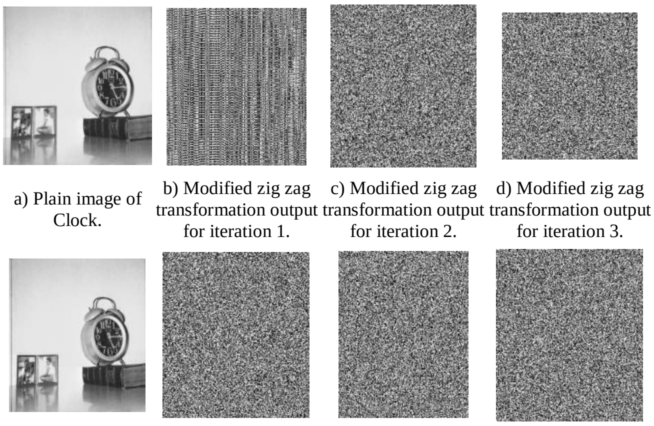
a) Cipher *CI*$_1$ using *Key*$_1$. b) Cipher *CI*$_2$ using *Key*$_2$. c) Absolute difference between *CI*$_1$ and *CI*$_2$. d) Decrypt *CI*$_1$ using *Key*$_2$. e) Decrypt *CI*$_2$ using *Key*$_1$.

Figure 9. Key sensitivity analysis using two different Keys.

4.4. Performance Analysis

4.4.1. Permutation Effect Analysis

The permutation effect is shown for the modified zig zag transformation of the image matrix and for a total of three iterations. Figure 10 depicts that the number of iterations increases the permutation effect. The transformation is more uniform and outperforms the pervious iterations.



a) Plain image of Clock. b) Modified zig zag transformation output for iteration 1. c) Modified zig zag transformation output for iteration 2. d) Modified zig zag transformation output for iteration 3.

e) Plain image of Clock. f) Completely encrypted image for iteration 1. g) Completely encrypted image for iteration 2. h) Completely encrypted image for iteration 3.

Figure 10. Permutation effect of modified zig zag transformation and completely encrypted output for different iterations.

4.4.2. Correlation Effect of Permutation

The correlation coefficients of plain image after zig zag transformation is tabulated in Table 29. These values are given for iterations one, two and three.

Table 29. Correlation values of permuted Peppers image.

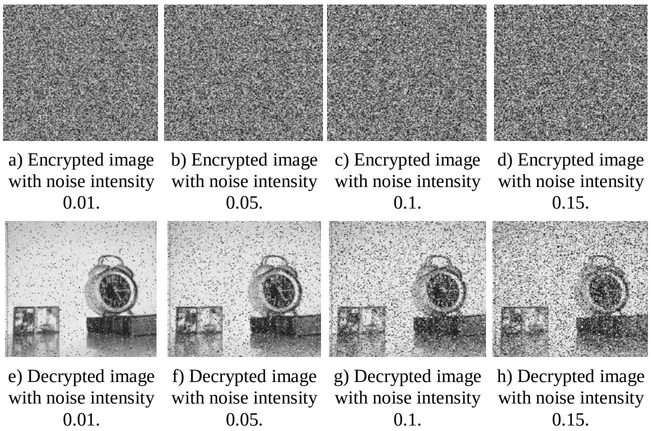
Iteration	Direction	Plain image	Permuted image
Iteration=1	Horizontal	0.9721	-0.00148
	Vertical	0.9767	-0.00139
	Diagonal	0.9651	0.00061
Iteration=2	Horizontal		0.00059
	Vertical		0.00075
	Diagonal		0.00097
Iteration=3	Horizontal		0.00032
	Vertical		0.00038
	Diagonal		-0.00347

The correlation values decrease as the iteration number increase.

4.5. Robustness Analysis

4.5.1. Noise Attack Analysis

The cipher images underwent a deliberate injection of salt and pepper noise at varying intensities: 0.01, 0.05, 0.1, and 0.15. Following this, the noisy cipher images were subjected to the decryption process and are presented in Figure 11. Decrypted images retain noticeable traces of the introduced noise. However, it is worth highlighting that despite the noise, the images remain quite recognizable. This significant observation underscores the algorithm's capacity to effectively withstand noise interference, showcasing its robust anti-noise property.

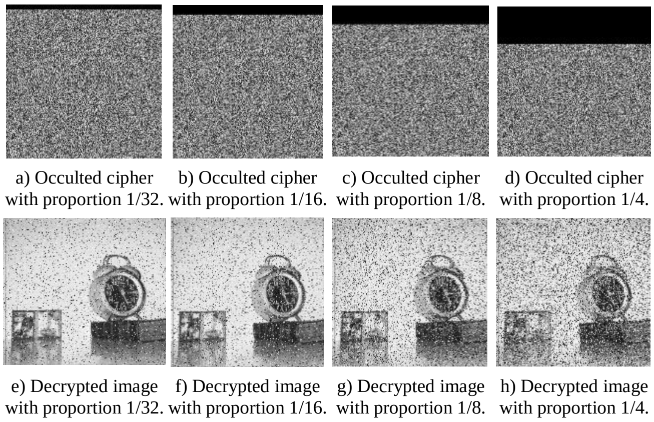

a) Encrypted image with noise intensity 0.01. b) Encrypted image with noise intensity 0.05. c) Encrypted image with noise intensity 0.1. d) Encrypted image with noise intensity 0.15.

e) Decrypted image with noise intensity 0.01. f) Decrypted image with noise intensity 0.05. g) Decrypted image with noise intensity 0.1. h) Decrypted image with noise intensity 0.15.

Figure 11. Noise attack analysis of encrypted and decrypted Clock image using different noise intensity levels.

4.5.2. Cropping Attack Analysis

The original clock image is introduced with varying degrees of occlusion (1/32, 1/16, 1/8, and 1/4) as shown in Figure 12-a), (b), (c), and (d). The subsequent step involved decrypting the resulting cropped cipher images, which are depicted in Figure 12-e), (f), (g), and (h). Despite the intentional occlusion, the decrypted images displayed remarkable clarity, showcasing the algorithm's resilience against cropping attacks. This shows the algorithm's ability to maintain image fidelity, even in scenarios involving significant content concealment.


a) Occulted cipher with proportion 1/32. b) Occulted cipher with proportion 1/16. c) Occulted cipher with proportion 1/8. d) Occulted cipher with proportion 1/4.

e) Decrypted image with proportion 1/32. f) Decrypted image with proportion 1/16. g) Decrypted image with proportion 1/8. h) Decrypted image with proportion 1/4.

Figure 12. Cropping attack analysis of Clock image using different degrees of occlusion.

4.5.3. Speed Analysis

A faster algorithm is needed to handle large data like images quickly and efficiently. Adding more rounds to a cipher increases security but slows performance. On the other hand, reducing rounds improves performance but lowers security.

Table 30. Speed analysis.

Test image	[18]	[1]	Proposed
256x256	4.7973	4.5112	4.8562
512x512	9.1884	12.0848	11.6921
1024x1024	32.2786	42.1770	43.1967

Table 30 infers that the time taken in seconds by the proposed method to execute the program is comparable with the other existing methods.

4. Conclusions

In this paper, an image encryption algorithm using modified perturbed logistic map is proposed. The performance of the proposed algorithm is evaluated using different performance metrics like EQ, MD, chi square test, NPCR, UACI, correlation analysis, local entropy, global entropy, noise attack analysis, cropping attack analysis, key space and key sensitivity. The metric values obtained using proposed method is compared with other encryption schemes available in the literature. The key space of the proposed scheme is large. Hence the proposed work can resist brute force attack.

References

[1] Alawida M., "A Novel Chaos-Based Permutation for Image Encryption," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 6, pp. 101595, 2023. https://doi.org/10.1016/j.jksuci.2023.101595

[2] Al-Hazaimeh O., Al-Jamal M., Bawaneh M., Alhindawi N., and Hamdoni B., "A New Image Encryption Scheme Using Dual Chaotic Map Synchronization," *The International Arab Journal of Information Technology*, vol. 18 no. 1, pp. 95-102, 2021. https://www.iajit.org/portal/PDF/Vol%2018,%20No.%201/19572.pdf

[3] Demirtas M., "A Novel Multiple Grayscale Image Encryption Method Based on 3D Bit-Scrambling and Diffusion," *Optik*, vol. 266, pp. 169624, 2022. https://doi.org/10.1016/j.ijleo.2022.169624

[4] Gayathri J. and Subashini S., "A Spatiotemporal Chaotic Image Encryption Scheme Based on Self Adaptive Model and Dynamic Keystream Fetching Technique," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 24751-24787, 2018. https://doi.org/10.1007/s11042-018-5675-4

[5] Hua Z., Zhang K., Li Y., and Zhou Y., "Visually Secure Image Encryption Using Adaptive-

Thresholding Sparsification and Parallel Compressive Sensing," *Signal Processing*, vol. 183, pp. 107998, 2021. https://doi.org/10.1016/j.sigpro.2021.107998

[6] Jiang X., Jiang G., Wang Q., and Shu D., "Image Encryption Algorithm Based on 2D-CLICM Chaotic System," *IET Image Processing*, vol. 17, no. 7, pp. 2127-2141, 2023. https://doi.org/10.1049/ipr2.12778

[7] Khanzadi H., Eshghi M., and Borujeni S., "Image Encryption Using Random Bit Sequence Based on Chaotic Maps," *Arabian Journal for Science and Engineering*, vol. 39, no. 2, pp. 1039-1047, 2014. https://doi.org/10.1007/s13369-013-0713-z

[8] Liu Y., Qin Z., Liao X., and Wu J., "A Chaotic Image Encryption Scheme Based on Hénon-Chebyshev Modulation Map and Genetic Operations," *International Journal of Bifurcation and Chaos*, vol. 30, no. 6, pp. 2050090, 2020. https://doi.org/10.1142/S021812742050090X

[9] Man Z., Li J., Di X., Sheng Y., and Liu Z., "Double Image Encryption Algorithm Based on Neural Network and Chaos," *Chaos, Solitons and Fractals*, vol. 152, pp. 111318, 2021. https://doi.org/10.1016/j.chaos.2021.111318

[10] Patel S., Bharath K., and Kumar R., "Symmetric Keys Image Encryption and Decryption Using 3D Chaotic maps with DNA Encoding Technique," *Multimedia Tools and Applications*, vol. 79, no. 43-44, pp. 31739-31757, 2020. https://doi.org/10.1007/s11042-020-09551-9

[11] Patro K., Soni A., Netam P., and Acharya B., "Multiple Grayscale Image Encryption Using Cross-Coupled Chaotic Maps," *Journal of Information Security and Applications*, vol. 52, pp. 102470, 2020. https://doi.org/10.1016/j.jisa.2020.102470

[12] Ponuma R. and Amutha R., "Image Encryption Using Sparse Coding and Compressive Sensing," *Multidimensional Systems and Signal Processing*, vol. 30, no. 4, pp. 1895-1909, 2019. https://doi.org/10.1007/s11045-019-00634-x

[13] Ponuma R. and Amutha R., "Compressive Sensing Based Image Compression-Encryption Using Novel 1D-Chaotic Map," *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 19209-19234, 2018. https://doi.org/10.1007/s11042-017-5378-2

[14] Ponuma R. and Amutha R., "Encryption of Image Data Using Compressive Sensing and Chaotic System," *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 11857-11881, 2019. https://doi.org/10.1007/s11042-018-6745-3

[15] Ponuma R., Amutha R., Aparna S., and Gopal G., "Visually Meaningful Image Encryption Using Data Hiding and Chaotic Compressive Sensing," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 25707-25729, 2019. https://doi.org/10.1007/s11042-019-07808-6

[16] Shakiba A., "A Novel Randomized Bit-Level Two-Dimensional Hyperchaotic Image Encryption Algorithm," *Multimedia Tools and Applications*, vol. 79, no. 43-44, pp. 32575-32605, 2020. https://doi.org/10.1007/s11042-020-09434-z

[17] Sheela S., Sanjay A., Suresh K., Tandur D., and Shubha G., "Image Encryption Based on 5D Hyperchaotic System Using Hybrid Random Matrix Transform," *Multidimensional Systems and Signal Processing*, vol. 33, no. 2, pp. 579-595, 2022. https://doi.org/10.1007/s11045-021-00814-8

[18] Wang Q., Zhang X., and Zhao X., "Image Encryption Algorithm Based on Improved Zig Zag Transformation and Quaternary DNA Coding," *Journal of Information Security and Applications*, vol. 70, pp. 103340, 2022. https://doi.org/10.1016/j.jisa.2022.103340

[19] Wang X. and Si R., "A New Chaotic Image Encryption Scheme Based on Dynamic L-Shaped Scrambling and Combined Map Diffusion," *Optik*, vol. 245, pp. 167658, 2021. https://doi.org/10.1016/j.ijleo.2021.167658

[20] Wang X. and Zhang M., "An Image Encryption Algorithm Based on New Chaos and Diffusion Values of a Truth Table," *Information Sciences*, vol. 579, pp. 128-149, 2021. https://doi.org/10.1016/j.ins.2021.07.096

[21] Wang X. and Zhao M., "An Image Encryption Algorithm Based on Hyperchaotic System and DNA Coding," *Optics and Laser Technology*, vol. 143, pp. 107316, 2021. https://doi.org/10.1016/j.optlastec.2021.107316

[22] Wang X., Guan N., and Yang J., "Image Encryption Algorithm with Random Scrambling Based on One-Dimensional Logistic Self-Embedding Chaotic Map," *Chaos, Solitons and Fractals*, vol. 150, pp. 111117, 2021. https://doi.org/10.1016/j.chaos.2021.111117

[23] Wang X., Liu L., and Zhang Y., "A Novel Chaotic Block Image Encryption Algorithm Based on Dynamic Random Growth Technique," *Optics and Lasers in Engineering*, vol. 66, pp. 10-18, 2015. https://doi.org/10.1016/j.optlaseng.2014.08.005

[24] Wang X., Zhang Y., and Bao X., "A Novel Chaotic Image Encryption Scheme Using DNA Sequence Operations," *Optics and Lasers in Engineering*, vol. 73, pp. 53-61, 2015. https://doi.org/10.1016/j.optlaseng.2015.03.022

[25] Wei D., Jiang M., and Deng Y., "A Secure Image Encryption Algorithm Based on Hyper-Chaotic and Bit-Level Permutation," *Expert Systems with Applications*, vol. 213, pp. 119074, 2023. https://doi.org/10.1016/j.eswa.2022.119074

[26] Wu Y., Zhou Y., Saveriades G., Agaian S., Noonan J., and Natarajan P., "Local Shannon Entropy Measure with Statistical Tests for Image Randomness," *Information Sciences*, vol. 222, pp.

323-342, 2013. https://doi.org/10.1016/j.ins.2012.07.049

[27] Xian Y. and Wang X., "Fractal Sorting Matrix and its Application on Chaotic Image Encryption," *Information Sciences*, vol. 547, pp. 1154-1169, 2021. https://doi.org/10.1016/j.ins.2020.09.055

[28] Xu C., Sun J., and Wang C., "An Image Encryption Algorithm Based on Random Walk and Hyperchaotic Systems," *International Journal of Bifurcation and Chaos*, vol. 30, no. 4, pp. 2050060, 2020. https://doi.org/10.1142/S0218127420500601

[29] Xu D., Li G., Xu W., and Wei C., "Design of Artificial Intelligence Image Encryption Algorithm Based on Hyperchaos," *Ain Shams Engineering Journal*, vol. 14, no. 3, pp. 101891, 2023. https://doi.org/10.1016/j.asej.2022.101891

[30] Yang F., Mou J., Cao Y., and Chu R., "An Image Encryption Algorithm Based on BP Neural Network and Hyperchaotic System," *China Communications*, vol. 17, no. 5, pp. 21-28, 2020. https://doi.org/10.23919/JCC.2020.05.003

[31] Ye G., Pan C., Dong Y., Shi Y., and Huang X., "Image Encryption and Hiding Algorithm Based on Compressive Sensing and Random Numbers Insertion," *Signal Processing*, vol. 172, pp. 107563, 2020. https://doi.org/10.1016/j.sigpro.2020.107563

[32] Ye G., Wu H., Jiao K., and Mei D., "Asymmetric Image Encryption Scheme Based on the Quantum Logistic Map and Cyclic Modulo Diffusion," *Mathematical Biosciences and Engineering*, vol. 18, no. 5, pp. 5427-5448, 2021. DOI:10.3934/mbe.2021275

[33] Zhang Q., Han J., and Ye Y., "Multi-Image Encryption Algorithm Based on Image Hash, Bit-Plane Decomposition and Dynamic DNA Coding," *IET Image Processing*, vol. 15, no. 4, pp. 885-896, 2021. https://doi.org/10.1049/ipr2.12069

[34] Zhang Q., Liu L., and Wei X., "Improved Algorithm for Image Encryption Based on DNA Encoding and Multi-Chaotic Maps," *AEU-International Journal of Electronics and Communications*, vol. 68, no. 3, pp. 186-192, 2014. https://doi.org/10.1016/j.aeue.2013.08.007

Subhashini Kumaran obtained her UG degree in ECE from Madras University in the year 2001 and PG degree in Medical Electronics from College of Engineering, Guindy, Chennai in the year 2007. She is currently working in Sri Sai Ram Engineering College, Chennai, India. She is a senior member of IEEE, fellow member of IETE and life member of ISTE. Her current research interest includes Image Processing.

Amutha Ramachandran is a Professor in the Department of ECE, Sri Sivasubramaniya Nadar College of Engineering, Chennai, India. Received her UG degree in ECE from Thiagarajar college of Engineering, India, in 1987. Received her PG degree from PSG college of Technology, Coimbatore, India and she was awarded PhD degree in the year 2006 from Anna University. Her total teaching experience is 37 years which includes 20 years of research experience. Her research interest includes Wireless Communication Network and Image Processing.