Palestine Journal of Mathematics

Vol 14(1)(2025) , 530–553

COLOR IMAGE ENCRYPTION SCHEME BASED ON RC4A AND CHAOTIC MAPS

Deep Singh, Manoj Kumar Patel and Mukesh*

Communicated by Manoj Kumar Patel

MSC 2020 Classifications: Primary 94A08; Secondary 68P25.

Keywords and phrases: RC4A cryptosystem, RMAC, Henon map, Tent map, Encryption, Confusion and diffusion.

Acknowledgement: *The first author is thankful to the Central University of Punjab, Bathinda, for providing financial support through RSM grant no.: CUPB/Acad./2022/1194, and to DST, India, for support through grant no.: SR/FST/MS-1/2021/104(C) under DST-FIST project. The corresponding author is thankful for the financial assistance from the CSIR, New Delhi, in the form of a Junior Research Fellowship (JRF). The authors would like to thank the reviewers and editor for their constructive comments and valuable suggestions that improved the quality of the paper.*

Abstract Today's technology has significantly advanced, enabling rapid data transfer over open networks with continuous growth. However, open networks constantly face threats of security breaches, allowing eavesdroppers unrestricted access to content. There is an urgent need for a rapid and efficient technique to secure data, minimize the risk of security breaches, and prevent unauthorized access to the content by eavesdroppers. To address these challenges, researchers must develop an efficient and robust cryptographic algorithm. This paper introduces a novel multilayer image encryption algorithm that combines the RC4A and RMAC cryptosystems with the Henon and Tent maps. The algorithm starts by partially encrypting the plain image through XOR with a key generated using the RC4A cryptosystem. To reduce the correlation among the image pixels, the algorithm employs a 2-D Henon map for confusion. The algorithm utilises the RMAC cryptosystem to enhance the security level significantly. Finally, a sequaence is generated using chaotic Tent map and XORed with the partially encrypted image. The scheme's vast keyspace provides additional security to plain images and resists brute-force attacks. Simulations with various colour images demonstrate the efficiency of the proposed encryption algorithm. The performance of the proposed encryption algorithm is evaluated using multiple metrics such as correlation coefficient, histogram analysis, and occlusion attack. The results affirm the robustness and high efficiency of the proposed encryption algorithm in resisting the statistical attacks.

1 Introduction

In today's society, we are overly dependent on the internet and become an essential resource for us. It is a convenient tool for work, education, shopping, banking, personal entertainment, and more. The remarkable evolution of networked multimedia transmission over the past decade has greatly enhanced the sharing of multimedia content like videos, audio, and images. However, the security of this digital data traffic still requires improvement for the research community [39]. In recent decades, cryptographers have focused extensively on the security of confidential data. Network and communication technologies are crucial for transferring images over the internet. Images are promptly utilized in various fields, including engineering services, medical imaging like MRI and CT scans, education, advertising, art, social networking and more. Images represent a significant form of multimedia. Images visually convey information and often provide a richer presentation than text alone. Securing images transmitted via digital communication systems to prevent modification, partial or total removal of their contents, and the addition of incorrect information is a significant research challenge for researchers [7]. During past two

* Corrosponding Author

decades, researchers have developed various types of information security techniques.

The extensive data and high correlation of image pixels make traditional techniques like AES [9, 52], DES [24, 42], RSA [40, 48], ElGamal [23, 31, 38] less efficient for digital image encryption. Recently, a number of image encryption algorithm have been developed to address this issue based on elliptic curve cryptography [51], transform domain [26], chaotic maps [41], DNA encoding [29, 44], S-box [30, 53, 54] to ensure the secure transmission of images over open networks. Chaotic systems have drawn researcher's attention because of their notable properties, such as sensitivity to initial parameters and pseudo-randomness. These properties of chaotic systems are well-suited to the need of image encryption algorithms. A chaotic map typically generates a sequence which is efficiently utlized to introduce confusion and diffusion among image data. In 1989, Mathews introduced the first image encryption algorithm based on chaotic systems . Since then, researchers have proposed numerous enhanced image encryption algorithms by using chaotic systems 2-D and 3-D Arnold map, Henon map, Tent map etc. Alexan et al. [2] proposed a colour image encryption method involving two encryption keys to introduce confusion among image pixels. A combination of a 2-D logistic sine map and a linear congruential generator is employed to generate the first key, whereas The Tent map and the Bernoulli's map is utilized to generate the second key. In [47], Chen's chaotic system generates a random sequence for the permutation of image pixels. Further in [14], image pixels are confused using a logistic map, which reduces the correlation within the image data. Lone et al. [28] proposes a lossless image encryption technique incorporating random matrix affine cipher (RMAC) and two chaotic maps: 2-D Henon and logistic map. In [33], the authors propose a multilayer image encryption algorithm in the discrete Hartley domain by incorporating RMAC, the discrete Hartley transform, and the random matrix shift cipher (RMSC). In [25], wavelet decomposition uses the 'bior3.7' wavelet to get low and high-frequency wavelet coefficient matrix, and a 1-D logistic map reorders the low-frequency wavelet coefficient matrix to encrypt the image. In their color encryption algorithm, Kaur et al. [18] utilized reality-preserving fractional Hartley transformation to simplify the handling of transformed coefficients. DNA-based image encryption techniques are very popular because of their significant parallel processing capabilities and large data storing capacity. Yu et al. [49] break down and encoded the RGB channels of the plain image into DNA matrices by introducing a novel DNA sequencing method called DNA triploid mutation (DNSA-TM) to perform cryptographic conversions. Following this, DNA matrices undergo decoding to achieve good level of confusion and diffusion. The RC4 cryptosystem for image cryptography is actively utlized to develope various methods to obtain improved security and efficiency. In [32], the authors have obtained enhanced randomness and complexity by employing RC4 to generate an S-box for byte substitution. Kumari et al. [21] propose an image encryption scheme utilizing RC4, and a chaotic map to create a random sequence needed for the diffusion process. In [1], the RC4 algorithm and the chaotic Henon map were combined to generate sub-keys. In [8], the authors used a gingerbreadman map to scramble image pixels, while the key stream generated by the RC4A cryptosystem performs the diffusion. Bhowmick et al. [5] use a middle square algorithm to shuffle rows and columns, while RC4A generates a stream of random numbers to diffuse image pixels. In [13], the Peter De Jong map determines the initial keys for the RC4 stream generator, which then produces the pseudo-random numbers that confuse and diffuse image pixels. In the paper [17], authors have utilized a chaotic Henon map to permute the image, followed by encryption using a Hill cipher with a key derived from an orthogonal matrix and concludes with a chaotic Tent map sequence for the final XOR operation. In [16], the authors use a sine map for permutation, a Hill cipher with a prime circulant matrix for substitution, and a chaotic Tent map for diffusion. In the [43], the authors introduce a modified RC-4 with negative numbers in the key generation schedule to mitigate traditional security threats.

2 Motivation and proposed frame work

This paper introduces a novel multilayer encryption algorithm for colour images by combining two cryptosystems. RC4A (Rivest Cipher 4A) and RMAC (Random matrix affine cipher) along

with two chatoic maps: 2-D Henon map and 1-D Tent map. The RC4A is a symmetric stream cipher employed in two stages: key stream generation and pseudo-random number generation. Two state arrays are initialized and used for pseudo-random number generation during the key stream generation stage. In the pseudo-random number generation stage, the algorithm continuously updates the state array to produce a pseudo-random sequence of bytes, which is utilized to diffuse the image pixels. In the RMAC stage, the additive and multiplicative parameters introduce confusion and diffusion among image pixels of each RGB channel. The 2D henon map is utilized to enhance the level of confusion among pixels. Finally, Tent map is employed to obtain perfectly diffused image data.
The prposed algorithm follows this sequence:
Stage(1): In the first stage, the algorithm applies the RC4A cryptosystem to the plain image data. The steps of the RC4A cryptosystem include:

- Initialization
- Keystream generation
- Pseudo-random number generation
- XOR operation stage

Stage (2): In the second stage, the algorithm applies a 2-D Henon map to a partially encrypted image obtained in Stage (1). The 2-D Henon map generates two sequences which contributes to confuse the image pixels.

Stage (3): In the third stage, the algorithm uses RMAC on a partially encrypted image obtained after Stage (2). The algorithm uses different additive and multiplicative parameters to suffle and diffuse even rows, even columns, odd rows, and odd columns.

Stage (4): In the final stage of the encryption process, a 1-D Tent map is applied to generate a sequence to diffuse the partially encrypted image obtained from Stage (3) using XOR operation. To decrypt the encrypted image, the inverse of each operation starting from stage (4) and moving back to stage (1)is performed. We evaluate the performance of the proposed cryptosystem by assessing various quality metrics through simulation. The numerical results illustrate the effectiveness and resilience of the proposed scheme.
The paper is organized as follows: Section 3 outlines the fundamental principles and maps required to formulate the proposed encryption algorithm. Section 4 details the procedure for implementing the proposed encryption algorithm. Section 5 presents the simulation results of the proposed encryption algorithm. Section 6 covers security analysis and statistical analysis. Section 7 concludes the proposed paper.

3 Preliminaries

3.1 Rivest Cipher 4

In 1987, Ron Rivest [35] designed a symmetric stream cipher known as the RC4 technique which is frequently used for data encryption. It is also referred to as "Rivest Cipher 4" and "ARCFOUR" (Alleged RC4). Due to its simplicity, speed, and ease of generating a pseudo-random number generator (PRNG), the RC4 [21] algorithm is often used in design of software and hardware. RC4 is primarily used in internet protocols such as WEP, Skype and WPA for file encryption, confidentiality, and communication security.

3.2 RC4A stream cipher

Souradytuti Paul and Bart Preneel proposed a new variation of RC4 in 2004, which is known as RC4A. Ginting et al. [10] uses a single counter, j, corresponding to state S,whereas RC4A Gaffer et al. [8] uses two counters, j_1 and j_2, corresponding to two states S_1 and S_2. Each increment of i produces two bytes. The RC4A cipher technique requires the same number of operations per output byte, it offers greater parallelism than RC4, resulting in significantly improved

speed and considered as more secure because it is less vulnerable to many attacks compared to RC4 [8].

3.3 RC4A algorithm

The RC4A Gaffar et al. [8] cipher consists of two main steps: the key scheduling algorithm (KSA) and the pseudo-random generation algorithm (PRGA).

Key Scheduling Algorithm

Figure 1(a) explains the key scheduling algorithm. The key scheduling algorithm consists of three stages. The first step of the key scheduling algorithm involves initializing two states, S_1 and S_2, each of length 256, such that $S_1[0] = S_2[0] = 0$, $S_1[1] = S_2[1] = 1, ...,$ and $S_1[255] = S_2[255] = 255$. In states S_1 and S_2, each element represents a byte. Here, the secret key, K, has a length between $1 \leq K \leq 256$. The key length is the number of bytes in the key (K). P is a temporary array of length 256. If the key (K) length is 256, then $P = K$; otherwise, elements of the key (K) are repeated to fill P until it reaches length 256. Figure 1(a) depicts the key scheduling algorithm of RC4A.

The second step of the key scheduling algorithm involves the initial permutation of state S_1. Calculate j for each index i using the formula $(j + S_1[i] + P[i])mod\,256$, followed by swapping $S[i]$ and $S[j]$. This process is detailed below.

If K is '87654321', the array P is filled as follows: [8765432187654321...78]. For $i = 0$ and initial $j = 0$, update j using the formula $(j + S_1[i] + P[i])mod\,256$, which simplifies to $(0 + S_1[0] + P[0])mod\,256$, resulting in (0 + 0 + 8) = 8. Next, swap the elements at positions i^{th} $(i = 0)$ and j^{th} $(j = 8)$, resulting in the interchange of $S_1[0]$ and $S_1[8]$. Now, the array S_1 becomes [8, 1, 2, 3, 4, 5, 6, 7, 0, 9, 10, 11, 12, 13, ..., 255]. Calculate j again for $i = 1$ with initial $j = 8$, and update the state S_1 accordingly. Update state S_1 from $i = 0$ to $i = 255$ based on the corresponding value of j calculated using the earlier process. Similarly, in the third step of the key scheduling algorithm, the state S_2 was updated, as shown in Figure 1(a).

Pseudo-random generation algorithm

The pesudo-random genrator algorithm evaluates keystreams U_1 and U_2 using two state array S_1 and S_2. The pseudo-code shown in Figure 1(b) utilizes an image's number of rows and columns. Calculate keystream U_1 using a method similar to RC4, but evaluate it based on the state array S_2 rather than S_1. Generate keystream U_2 by indexing j_2 in the state array S_2 with i fixed. Referencing Figure 1(b) for the algorithm, calculate the result of $(S_2[i]+S_2[j_2])$ using S_1 instead of S_2.

Encryption algorithm

The encryption process is given by as:

$$A_1 = X \oplus U_1$$
$$A_2 = A_1 \oplus U_2$$

where, X represents the plaintext image, U_1 and U_2 denote the keystreams, and $\oplus$ denotes the XOR operation.

3.4 2-D Henon map

The two-dimensional Henon map, introduced by the French astronomer Michel Henon in 1976, is a simplified representation of the Lorenz system [27]. The 2-D Henon map is non-linear and chaotic, originating from a discrete-time dynamical system grounded in differential equations. It

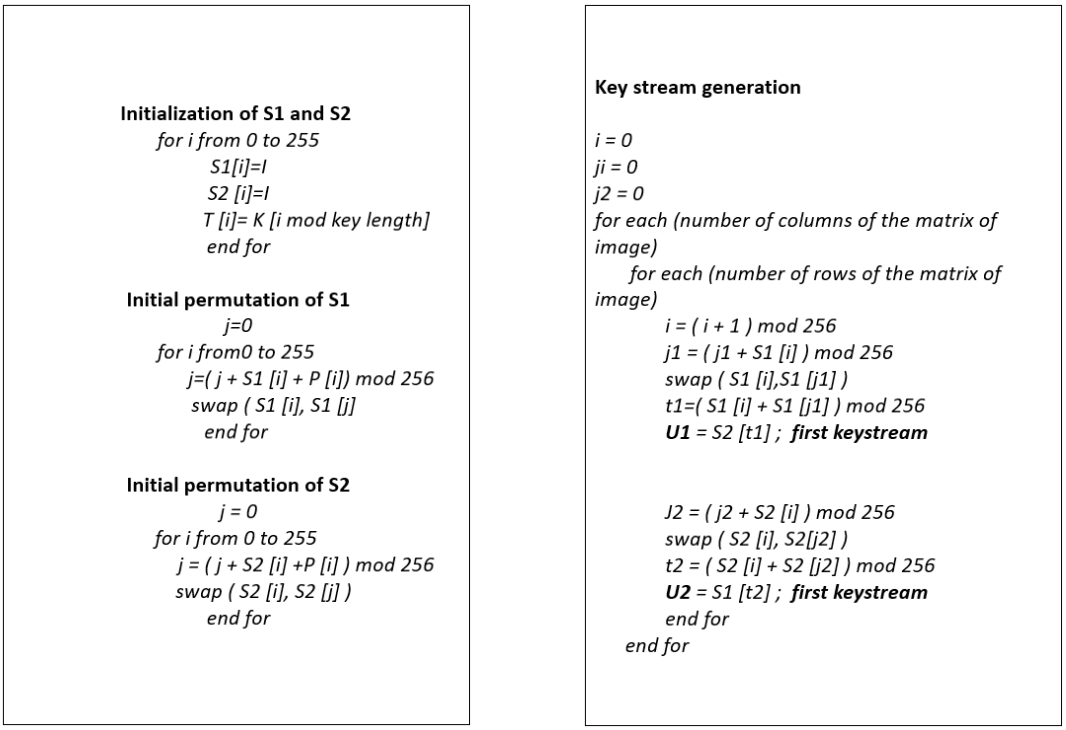

Figure 1: (a) key stream generation algorithm and (b) pesudo random generation algorithm

is a one-to-one mapping with properties of folding and stretching. The two-dimensional Henon map [28] is described as follows:

$$\begin{cases} x_{n+1} = 1 - a{x_n}^2 + y_n \\ y_{n+1} = bx_n. \end{cases} \tag{3.1}$$

The chaotic behavior of the Henon map depends on the parameters a and b as well as the variables (x_n, y_n) and the number of iterations n. Here a and b are the bifurcation parameters, also known as chaotic attractors, which control the degree of stretching and the thickness of folding. (x_n, y_n) are known as state variables and n is the number of iterations. The higher chaotic behavior of the dynamical system can be demonstrated when $a = 1.4$ and $b = 0.3$.

From the classical Henon map defined in (3.1), it is observed that time is discrete while the state variables are continuous. In an image cryptosystem, converting or sorting floating-point numbers is time-consuming. To mitigate this issue, the Henon map is discretized and implemented in the integer domain [37] as follows:

$$\begin{cases} x_{m+1} = (1 - a_1{x_m}^2 + y_m) \, mod \, n \\ y_{m+1} = (a_2 x_m + a_3) \, mod \, n \end{cases} \tag{3.2}$$

where a_1, a_2, a_3 are the control parameters, (x_m, y_m) and (x_{m+1}, y_{m+1}) represent the original and iterated positions of the pixel values and n is the size of digital image. If $a_2 = 1$; the Henon map is one-to-one corrospodence and a_1, a_3 acts as key parameters where a_1, $a_3 \in \mathbb{Z}_n$. The inverse transformation of Henon map is defined as:

$$\begin{cases} x_m = (y_{m+1} - a_3) \, mod \, n \\ y_m = (a_1{x_m}^2 + x_{m+1} - 1) \, mod \, n. \end{cases} \tag{3.3}$$

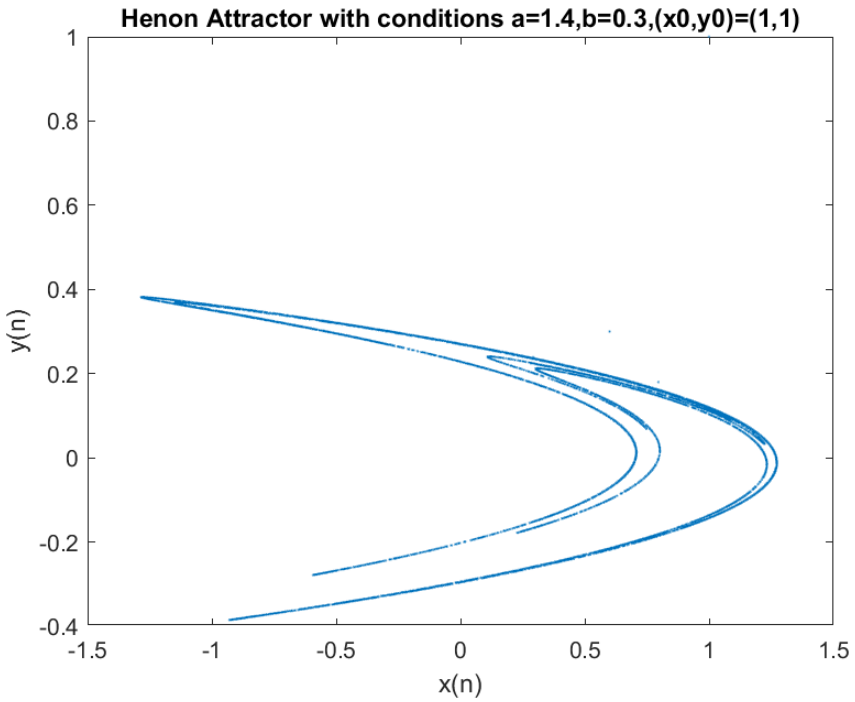

Figure 2: Attractor of 2-D Henon map

3.5 Random matrix affine cipher (RMAC)

The affine cipher is a widely known mono-alphabetic substitution cipher. People commonly use the affine cipher to encrypt textual data. The RMAC technique surpasses the affine cipher. The RMAC technique offers security in both the coordinate and geometric domains [39]. The RMAC algorithm implements both confusion and diffusion [28]. The pixel values in even-numbered rows are shifted by α_1 and multiplied by λ_1, while the pixel values in odd-numbered rows are shifted by α_2 and multiplied by λ_2 respectively. Similarly, the pixel values in even-numbered columns are shifted by β_1 and multiplied by η_1, while the pixel values in odd-numbered columns are shifted by β_2 and multiplied by η_2 respectively under the condition that $\alpha_1 \neq \alpha_2 \in \mathbb{Z}_m - \{0\}$ and $\beta_1 \neq \beta_2 \in \mathbb{Z}_n - \{0\}$ and multiplicative parameters λ_1 and λ_2 should be invertible under $mod\, m$ and the multiplicative parameters η_1 and η_2 should be invertible under $mod\, n$ i.e λ_1, λ_2 $\in \mathbb{U}$(m) and η_1, $\eta_2 \in \mathbb{U}$(n) respectively, where $\mathbb{U}$(n) is group of units multiplication $modulo\, n$. Mathematically $\mathbb{U}$(n)[6, 15] is defined below in Eqaution 3.4.

$$\mathbb{U}(n) = \{a \in n \,:\, a < n;\ gcd(a, n) = 1\} \tag{3.4}$$

Mathematically, RMAC algorithm on image A of size $m \times n$ is defined as:

$$A'_{even\, row, r1} \equiv \lambda_1 A_{even\, row, j+\alpha_1 (mod\, n)}, \tag{3.5}$$

$$A'_{odd\, row, r_2} \equiv \lambda_2 A_{odd\, row, j+\alpha_2 (mod\, n)}, \tag{3.6}$$

$$A'_{s_1,\, even\, column} \equiv \eta_1 A_{i+\beta_1 (mod\, m),\, even\, column}, \tag{3.7}$$

$$A'_{s_2,\, odd\, column} \equiv \eta_2 A_{i+\beta_2 (mod\, m),\, odd\, column}, \tag{3.8}$$

where A is the original image and A' is the image which is obtained by after applying the RMAC. The inverse of RMAC is defined as:

$$A_{even\, row, i} \equiv \lambda_1{}' A'_{even\, row, r_1+n-\alpha_1 (mod\, n)}, \tag{3.9}$$

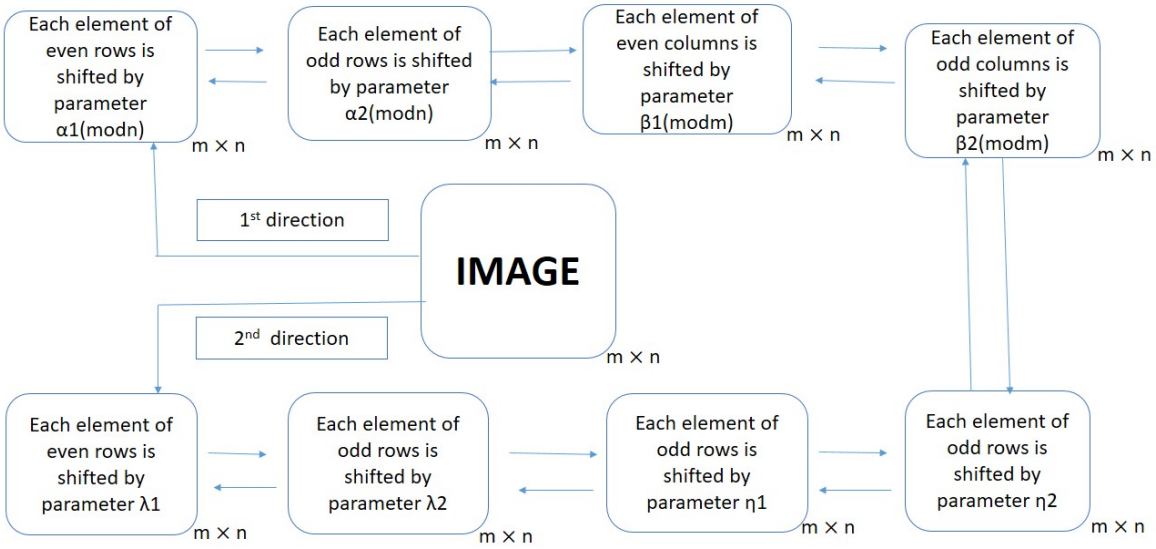

Figure 3: Flowchart of RMAC of colour image of size $m \times n$

$$A_{odd\,row,i} \equiv \lambda_2{}' A'_{odd\,row,r_2+n-\alpha_2(mod\,n)}, \tag{3.10}$$

$$A_{j,even\,row} \equiv \eta_1{}' A'_{s_1+m-\beta_1(mod\,m),even\,column} \tag{3.11}$$

$$A_{j,odd\,row} \equiv \eta_2{}' A'_{s_2+m-\beta_2(mod\,m),odd\,column} \tag{3.12}$$

where $\lambda_1{}'$, $\lambda_2{}'$ are the multiplicative inverse of λ_1, λ_2 w.r.t. $modulo\,m$ whereas $\eta_1{}'$, $\eta_2{}'$ are the multiplicative inverse of η_1, η_2 w.r.t. $modulo\,n$.

3.6 Tent map

The Tent map [22] is the simplest one-dimensional chaotic map defined as follows:

$$x_{m+1} = \begin{cases} \alpha x_m & if\ x_m \leq 0.5 \\ \alpha(1-x_m) & if\ x_m \geq 0.5 \end{cases} \tag{3.13}$$

where this map is a function on [0, 1] $i.e\ x_m \in [0,1]$ for $m \geq 0$. The initial condition, denoted as x_0, serves as a parameter where x_0 lies within the interval [0, 1]. The set $x_0, x_1, x_2, ...x_n, ...$ is referred to as the orbit of the system. The parameter α, where $\alpha \in [0, 2]$, acts as the control parameter of the system. The Tent map is a non-invertible, piecewise linear discrete map. Its chaotic behavior depends on the control parameter α. For the value of x_0=0.1 and α=1.9998, the function (x_m, α) gives its maximum at x_m=0.5, with the maximum value being $\alpha/2$, as shown in the Figure 4.

4 Proposed work

The proposed multilayer encryption scheme combines RC4A, Henon map, RMAC, and the Tent map. The detailed encryption algorithm is as follows:

Diffusion using RC4A:

Step 1: Generate two sequences, S_1 and S_2, each of length 256, using the key scheduling algorithm (KSA) of RC4A, as shown in Figure 1(a).

Step 2: Now, two keystreams, U_1 and U_2, are generated using the pseudo-random generator

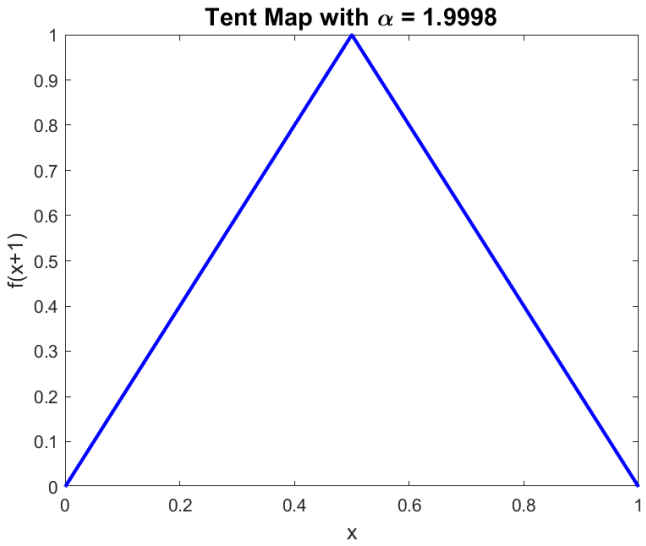

Figure 4: Graph of Tent map with initial condition x=0.1 and α=1.9998

(PRGA) of the RC4A algorithm, as shown in Figure 1(b).
Step 3: Encryption process using the RC4A algorithm is given as

$$for\ k = 1\ to\ 3$$
$$for\ l = 1\ to\ mn$$
$$A_{1k}(l) = X_k(l) \oplus U_1(l)$$
$$A_{2k}(l) = A_{1k}(l) \oplus U_2(l)$$
$$end\ for$$
$$end\ for$$

Now, A_{2k} is the diffused image obtained after applying the RC4A algorithm on the plaintext image X.

Confusion using Henon map: After applying the RC4A algorithm, the 2-D Henon map with parameters a,$c \in \mathbb{Z}_n$, is applied on partially encrypted image A_{2k} to obtain a new partially encrypted image P.

Confusion and diffusion using RMAC: Each component of partially encrypted colour image P is futher suffled and diffused by using the shift parameters and the multiplier parameters of RMAC cryptosystem, discussed in section 2.3.
The shift and multiplier parameters used in this algorithm are as follows:
Step 1: For the red channel(R): ϕ_R, ψ_R, ρ_R, σ_R are the shift parameters, and τ_R, Π_R, μ_R, ϵ_R are the multiple parameters.

Step 2: For the green channel(G): ϕ_G, ψ_G, ρ_G, σ_G are the shift parameters, and τ_G, Π_G, μ_G, ϵ_G are the multiple parameters.

Step 3: For the blue channel(B): ϕ_B, ψ_B, ρ_B, σ_B are the shift parameters, and τ_B, Π_B, μ_B, ϵ_B are the multiple parameters.

Diffusion using Tent map: Generate the sequence Y using a Tent map with the initial condition x_0 for diffusion. Apply XOR between control parameter and the partially encrypted image P_1 to obtain P_2.
Decryption process: The decryption process is simply the reverse of the encryption process. Key points to keep in mind during decryption are as follows:
a) The shift parameters used in the decryption process of RMAC are the additive inverses of those used in the encryption process.

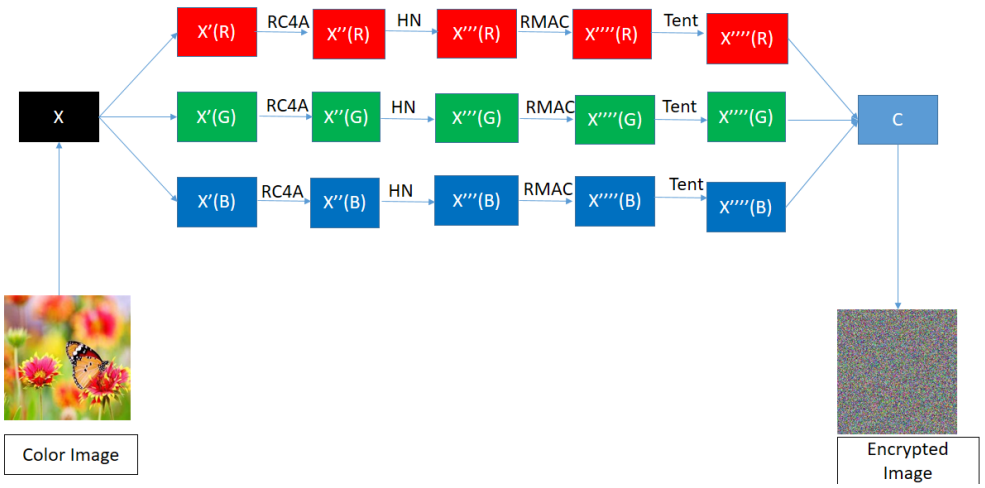

Figure 5: Diagram of encryption proposed algorithm

b) The decryption process of the Henon map uses the multiplicative inverses of the chaotic attractors a and b.

5 Result simulations

This section presents the simulation results of the proposed image encryption algorithm to assess its authenticity, robustness, and feasibility. All numerical and visual results are obtained using Matlab 2021b on a system equipped with an Intel (R) Core i5-9300 @ 2.40 GHz processor and 8GB of RAM. Four RGB images (Butterfly, Female, House, and Jelly), each with dimensions of 256 x 256 x 3, are taken from the USC-SIPI database to demonstrate the proposed algorithm. Figure 6(a-d) displays the four experimental images, and Figure 6(e-h) presents their encrypted versions. Figure 6(i-l) shows the decrypted images, which match with the original experimental images. Therefore, the proposed algorithm efficiently decrypts the encrypted images.

6 Security analysis

6.1 Key space

The key space comprises all potential keys for encrypting and decrypting an image within the algorithm. An encryption algorithm becomes more robust with a larger key space, making it difficult for an eavesdropper to conduct an exhaustive key search attack. Therefore, an effective encryption algorithm should possess a sufficiently sizeable key space to discourage unauthorized access. The total key count for the proposed algorithm is summarized as follows.

- The initial step includes diffusing the original image using the RC4A cryptosystem, employing two key matrices, U_1 and U_2, which are of same size, $m \times n$, as the original image. Each entry in the key matrices consists of eight bits. Therefore, the total key size of the RC4A cryptosystem is
$$(m \times n \times 8)^2.$$
- The key parameters of a 2-D Henon map are a, b and c, with $b = 1$ and $a, c \in \mathbb{Z}_n$. Thus, the possible key size for the 2-D Henon map is n^2.
- In the RMAC stage, the image pixels are confused using additive parameters α_1, α_2, β_1 and β_2 and diffused using multiplicative parameters λ_1, λ_2, η_1 and η_2. Here, $\alpha_1 \neq \alpha_2 \in$

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

(e) **(f)** **(g)** **(h)**

(i) **(j)** **(k)** (a) **(l)**

Figure 6: Figure 6(a-d) show the experimental images for proposed algorithm. Figure (e-h) show the encrypted images corrosponding to the experimental images. Figure (i-l) show the decrypted images corrosponding to the encrypted images.

$\mathbb{Z}_m - \{0\}$, $\beta_1 \neq \beta_2 \in \mathbb{Z}_n - \{0\}$, $\lambda_1, \lambda_2 \in \mathbb{U}(m)$ - {1} and $\eta_1, \eta_2 \in \mathbb{U}(n)$ - {1}. Therefore, the choice of parameters in the RMAC stage will be:

$$(m-1)(m-2)(n-1)(n-2)(\phi(m)-1)^2(\phi(n)-1)^2,$$

where $\phi(m)$ represents Euler's phi function given by

$$\phi(m) = |\{a \in \mathbb{N} : 1 \leq a < m \; and \; gcd(a,m) = 1\}|$$

- With a precision of 10^{-14}, the initial values and chaotic parameters in the 1-D Tent map contribute 10^{28} additional bits to the key space.

By combining the steps mentioned above, the total key space of the proposed algorithm is

$$10^{28}[(m \times n \times 8)^2 n^2 (m-1)(m-2)(n-1)(n-2)(\phi(m)-1)^2(\phi(n)-1)^2]^3.$$

For experimental purposes, we have used a color image (Butterfly) of size 256×256. The total size of the key space is $\approx 2^{401}$, which is significantly greater than the minimum required size of 2^{100}. Therefore, the proposed algorithm can withstand exhaustive key search attacks [3, 50] and ensure secure data communication.

6.2 Key sensitivity

Key sensitivity means that even a slight change in any encryption key parameter will significantly impact the resulting encrypted image. A robust encryption algorithm should exhibit high sensitivity towards its key parameters. The key sensitivity property is essential for maintaining the security of the encryption process, as it avoids attackers from deriving the correct key by using minor variations or approximations of the original key. The key sensitivity of the proposed algorithm is demonstrated in Figure 7 by making slight changes to the key parameters and initial conditions.

It can be observed that even a slight change in key parameters causes the decrypted image to differ entirely from the original image. Therefore, the proposed algorithm is highly sensitive towards the keys and their arrangements. This highlights the robustness of the proposed scheme.

6.3 Correlation analysis

Correlation analysis plays a crucial role in evaluating the robustness of an image encryption scheme. Correlation analysis is used to assess the quality of the encryption technique. Correlation refers to the statistical relationship between 4 adjacent pixels in an image, encompassing horizontal correlation (HC), vertical correlation (VC), and diagonal correlation (DC). Correlation test is applied to both, original image and compared image, which is defined as:

$$\begin{aligned} H(k) &= \frac{1}{M}\sum_{i=1}^{M} k_i \\ I(k) &= \frac{1}{M}\sum_{i=1}^{M}(k_i - H(k))^2 \\ \text{cov}(k,l) &= \frac{1}{M}\sum_{i=1}^{M}(k_i - H(k))(l_i - H(l)) \\ C_a &= \frac{\text{cov}(k,l)}{\sqrt{I(k)I(l)}} \end{aligned} \tag{6.1}$$

The values of the correlation coefficient lie between [-1, 1]. If the value of C_a is close to 1, it indicates a positive correlation, which suggests that the original image and the compared image are similar. This makes it easy for an attacker to steal information from the image. If the value of C_a is near 0, it indicates no relationship between the original and compared image. This makes it more difficult for an attacker to steal information from the image. If C_a has a value of -1, it means the compared image is the negative of the original image. Hence, the encryption system becomes more robust if the C_a value is close to zero.

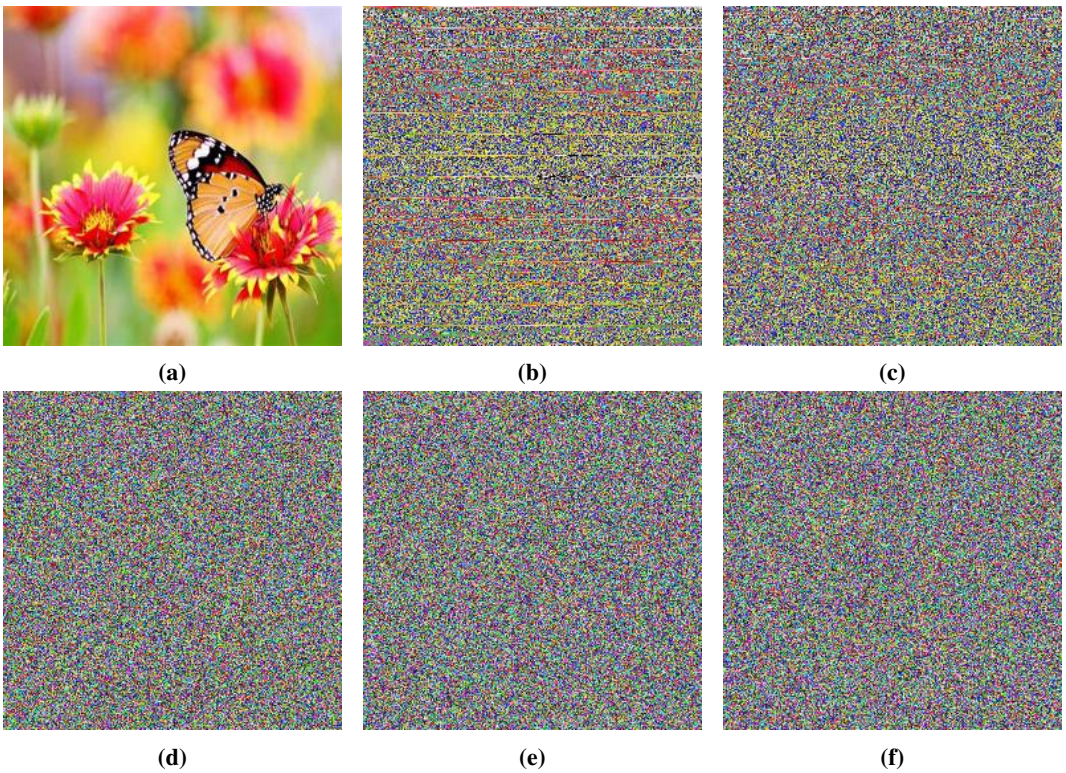

Figure 7: (a) original image. (b-c) show decrypted image by changing key parameters a and c respectively of Henon map. (d) shows decrypted image by changing key parameters of RMAC. (e) shows decrypted image by changing key parameters of Tent map.

Table 1: Horizontal correlation analysis for the original images and corrosponding encrypted images

Image	HC of the original image				HC of encrypted image			
	R	G	B	Average	R	G	B	Average
Butterfly	0.9433	0.9492	0.9479	0.9468	-0.0033	-0.0020	0.0028	- 0.0008
Female	0.9873	0.9492	0.9479	0.9614	-0.0081	1.9715e-04	-0.0021	-0.0033
House	0.9085	0.9603	0.9735	0.9474	-0.0014	-0.0061	-0.0058	0.0044
Jelly	0.9741	0.9841	0.9872	0.9818	0.0045	0.0053	0.0017	0.0038

Table 2: Vertical correlation(VC) analysis for the original images and corrosponding encrypted images

Image	VC of the original image				VC of encrypted image			
	R	G	B	Average	R	G	B	Average
Butterfly	0.9502	0.9571	0.9571	0.9548	-0.0033	-0.0019	0.0029	-0.0007
Female	0.9906	0.9871	0.9781	0.9852	-0.0082	5.0476e-04	-0.0019	-0.0032
House	0.9125	0.9634	0.9770	0.9509	-0.0016	-0.0063	-0.0061	-0.0046
Jelly	0.9746	0.9843	0.9875	0.9821	0.0043	0.0053	0.0016	0.0037

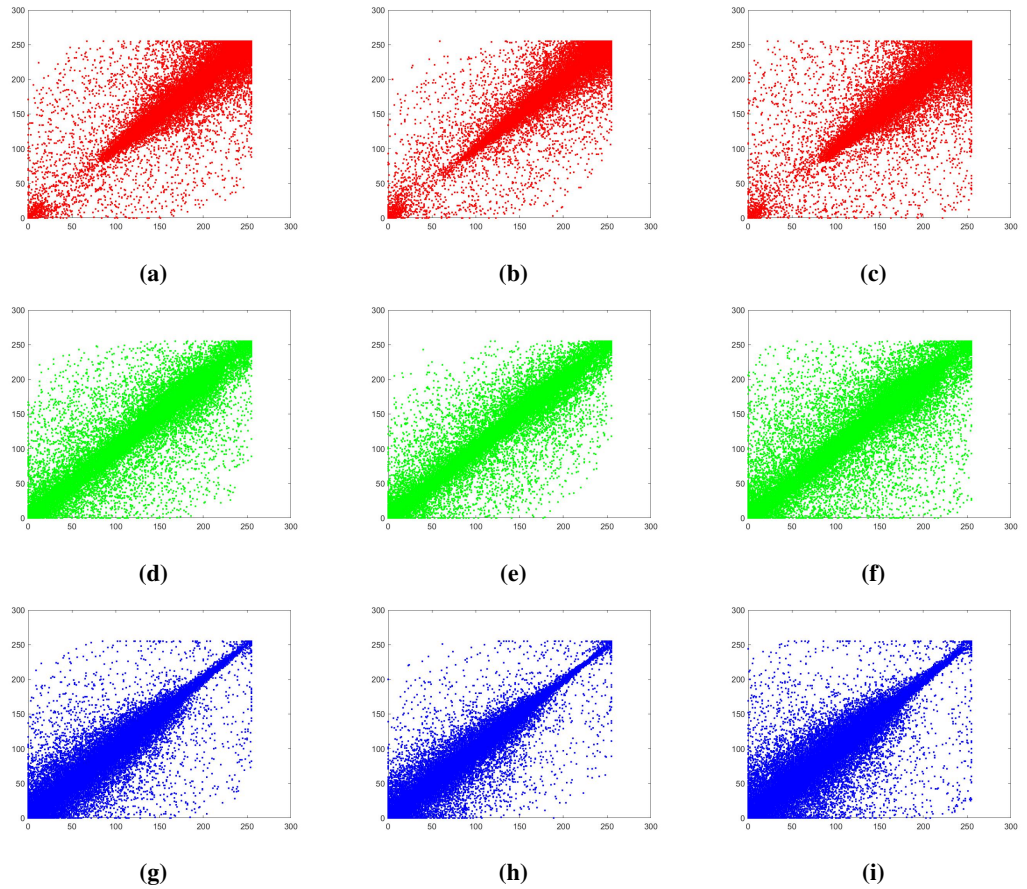

Figure 8: (a-c) show the horizontal, vertical and diagonal correlation of red component of Butterfly image. (d-f) show the horizontal, vertical and diagonal correlation of green component of Butterfly image. (g-i) show the horizontal, vertical and diagonal correlation of blue component of Butterfly image.

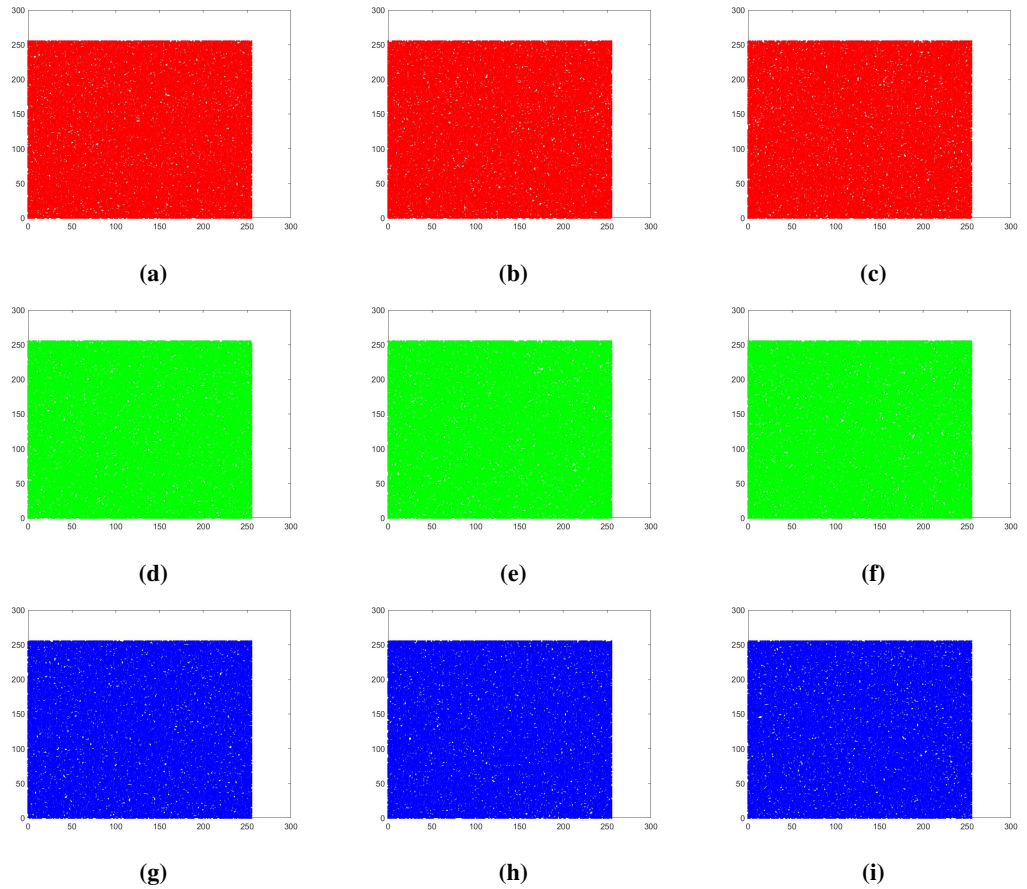

Figure 9: (a-c) show the horizontal, vertical and diagonal correlation of red component of encrypted image of Butterfly image. (d-f) show the horizontal, vertical and diagonal correlation of green component of encrypted image of Butterfly image. (g-i) show the horizontal, vertical and diagonal correlation of blue component of encrypted image of Butterfly image.

Table 3: Diagonal Correlation(DC) analysis for the original images and corrosponding encrypted images

Image	DC of the original image				DC of encrypted image			
	R	G	B	Average	R	G	B	Average
Butterfly	0.9076	0.9053	0.9149	0.9092	0.0077	0.0035	0.0019	0.0043
Female	0.9727	0.9600	0.9398	0.9575	2.5242e-04	-0.0019	-0.0067	-0.0028
House	0.8841	0.9498	0.9671	0.9336	-0.0013	-0.0024	-1.8673e-04	-0.0013
Jelly	0.9506	0.9677	0.9769	0.9650	-0.0023	-0.0017	0.0029	-0.0003

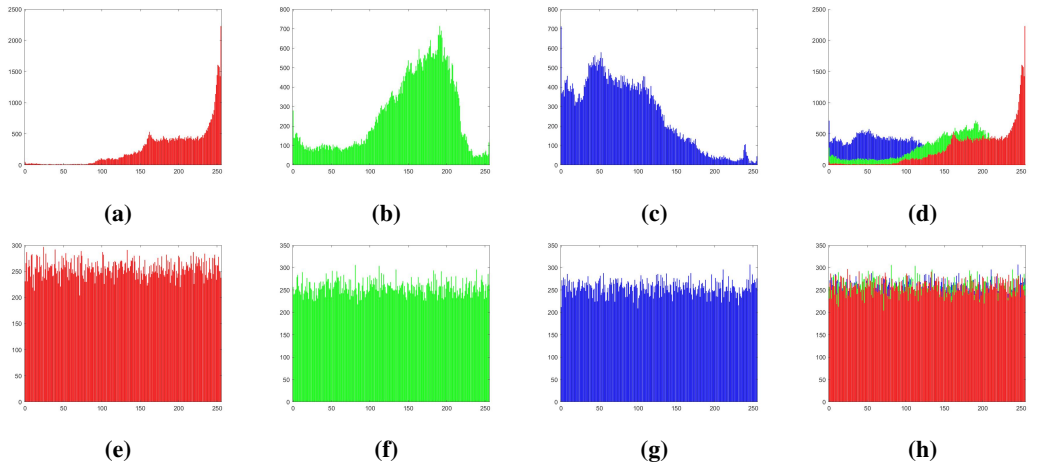

Figure 10: (a-c) show image histogram of red, green and blue components of Butterfly image. (d) shows the image histogram of butterfly image. (e-g) show image histogram of red, green and blue components of corrosponding encrypted image of Butterfly image. (h) shows the image histogram of encrrypted image of butterfly image.

6.4 Image histogram

A histogram of an image provides a clear graphical representation of the distribution of pixel intensities [11]. It is a statistical distribution graph that depicts the relationship between pixel intensity and the number of pixels in an image, mathematically defined as:

$$H(x_\alpha) = m_\beta, \tag{6.2}$$

where m_β is the geometric multiplicity of the α^{th} value x_α. For an algorithm to be secure, the histogram of an encrypted image should exhibit a uniform pixel intensity distribution and lack any distinct patterns to resist statistical attacks. In most cases, the histogram of an original image is characterized by peaks and uneven distribution. Figure 10(a-d) show the histogram of RGB components of the Butterfly image, while Figure 10(e-h) display the histogram of the RGB components of the encrypted Butterfly image. These figures clearly demonstrate that the histograms of the original image components and the encrypted image components are different. Additionally, the histogram of the encrypted image is uniform and lacks any peaks, making it resistant to statistical attacks.

6.5 Chi-square test

Histogram analysis provides a visual representation of an image's pixel values, whereas the Chi-square test offers a numerical interpretation of the uniformity in the pixel distribution of an

encrypted image. The Chi-square test [34] is defined by the following

$$\chi^2_{(exp)} = \sum_{j=1}^{256} \frac{(O_j - C_j)^2}{C_j}, \tag{6.3}$$

where $E_j = m*n/256$; E_j represents the expected value in encrypted images, while O_j denotes the observed value in the encrypted image. Table 4 presents the chi-square values at various levels of significance. The critical values of the chi-square test are 310.4574 and 316.9194 for degrees of freedom at significance levels of 0.005 and 0.01, respectively. From Table 4, it is evident that the chi-square values are less than the critical values, indicating that the pixel distribution in the encrypted image is uniform at both levels of significance. If the chi-square values are less than the critical values, it suggests that the proposed algorithm demonstrates resistance against statistical attacks.

Table 4: Chi-square values of different encrypted images

Image	chi-square values of the encrypted images			
	R	G	B	Average
Buterfly(256×256)	239.7812	229.7656	224.7266	231.4244
Female(256×256)	291.1719	298.3125	305.2266	298.2370
House(256×256)	289.8438	235.3906	259.1641	261.4661
Jelly(256×256)	252.7422	271.3359	238.3906	254.1562

6.6 Quality measure

Mean square error(MSE)

The mean square error (MSE)[36] between the original image and the cipher image is obtained by

$$MSE = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} [I_1(i,j) - I_2(i,j)], \tag{6.4}$$

where I_1 and I_2 represent the original and cipher images respectively, and $m \times n$ denotes the image size. A higher MSE value between original image and cipher image indicates better encryption quality. Table 6 presents the MSE values of the encrypted and decrypted images. The zero MSE values between the original and decrypted images confirm the accurate decryption of the encrypted image which is shown in Table 5 .

Table 5: MSE and PSNR values between original and decrypted images for different test imges

Image	MSE			PSNR		
	R	G	B	R	G	B
Butterfly	0	0	0	∞	∞	∞
Female	0	0	0	∞	∞	∞
House	0	0	0	∞	∞	∞
Jelly	0	0	0	∞	∞	∞

Peak signal-to-noise ratio(PSNR)

The peak-signal-to-noise ratio(PSNR)[20] is a metric calculated between the input and output images using

$$PSNR = 10\,log_{10}\left(\frac{2^n - 1}{MSE}\right), \tag{6.5}$$

where n is the number of bits per pixel. A good encryption algorithm produces lower PSER values between the input and encrypted images, typically less than 10 db. The PSNR values between the input and encrypted images using the specified algorithm are presented in Table 7. As seen in the Table, the PSNR values between the plain and encrypted images are below 10 db, indicating the effectiveness of the proposed algorithm.

Table 6: MSE values of different encrypted images

Image	MSE values between original and encrypted images			
	R	G	B	Average
Butterfly	1.30338e+04	9.0650e+03	1.0472e+04	1.0857e+04
Female	9.3533e+03	9.0476e+03	7.0491e+03	8.4833e+03
House	6.8863e+03	8.5312e+03	9.5875e+03	8.3350e+03
Jelly	9.2963e+03	1.0181e+04	7.4373e+03	8.9715e+03

Table 7: PSNR values of different encrypted images

Image	PSNR values of the encrypted images			
	R	G	B	Average
Buterfly(256×256)	6.987	8.5571	7.9305	7.8248
Female(256×256)	8.4212	8.5655	9.6494	8.8787
House(256×256)	9.7509	8.8207	8.3138	8.9618
Jelly(256×256)	8.4477	8.0531	9.4167	8.6391

Structural similarly index measure (SSIM)

The structural similarity index measure (SSIM)[45] is a quality measure to assess the likeness between two images. The SSIM helps us to compare the similarities between the original and decrypted images.

$$SSIM(r, s) = \frac{(2\mu_r\mu_s + s)(2\sigma_{rs} + s')}{(\mu_r^2 + \mu_s^2 + s)(\sigma_r^2 + \sigma_s^2 + s')}, \tag{6.6}$$

where σ_{rs},(μ_r, μ_s), and (σ_r, σ_s)represents the covariance, mean, and standard deviation of the original and encrypted images,respectively.
The value of SSIM lies in the interval [-1,1], and for a secure image encryption algorithm, SSIM values approach to 0. Table 8 shows the SSIM values between encrypted and decrypted images with plain images. Table 8 indicates that the SSIM values between plain and encrypted images are near to zero; hence, the proposed algorithm is efficient enough.

6.7 Entropy analysis

Entropy is defined as a measure of the randomness within a system [46]. Mathematically, it is computed as

$$Entropy = -\sum_{a=0}^{N-1} p(M_\alpha) log_2\,(p(M_\alpha)), \tag{6.7}$$

where M represents a data source and N denotes the different symbols present in M. $p(M_\alpha), M_\alpha \in M$ represents the probability of the pixel value M_α. In a robust encryption algorithm, the entropy

Table 8: SSIM analysis of decrypted images

Image	SSIM values of the decrypted images			
	R	G	B	Average
Buterfly(256 × 256)	0.0099	0.0094	0.0082	0.0091
Female(256 × 256)	0.0084	0.0106	0.0107	0.0099
House(256 × 256)	0.0109	0.0104	0.0096	0.0103
Jelly(256 × 256)	0.0103	0.0092	0.0115	0.0103

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

Figure 11: (a-e) represents the corrosponding encrypted images of Butterfly image after adding noise of 5%, 10%, 20%, 25%, 50%.

of the encrypted image should be close to 8. This indicates that all pixels in the encrypted image have nearly equal probabilities. Table 9 displays the entropy of the RGB image components of several RGB plain test images and encrypted images. Table 9 clearly indicates that the entropy of the encrypted image is very close to 8. This suggests that the proposed algorithm is resistent towards information leakage, ensuring security against entropy-based attacks.

Table 9: Entropy analysis for the plain images and corrosponding encrypted images

Image	Entropy of the original image				Entropy of encrypted image			
	R	G	B	Average	R	G	B	Average
Butterfly	7.1124	7.5928	7.5805	7.4285	7.9974	7.9975	7.9975	7.9974
Female	7.4961	7.4495	7.0539	7.3331	7.9970	7.9970	7.9970	7.9970
House	6.4510	6.6208	6.5657	6.5458	7.9972	7.9972	7.9973	7.9972
Jelly	5.4308	5.7598	6.6130	5.9345	7.9974	7.9974	7.9976	7.9974

6.8 Noise resistance

During data transmission over an open network, noise draws our attention to the issue of noise attacks [4]. In this attack, noise is first mixed into the encrypted images at varying intensity levels, and then these noisy encrypted images are decrypted. After adding salt and pepper noise, the decrypted images are shown in Figure 11(a-e). If these noise-affected images are decrypted to visualize clearly, the proposed algorithm performs well, providing maximum image resolution and demonstrating a high degree of similarity with the original image.

6.9 Occlusion attack

Some data gets tampered during transmission over open network channels. An occlusion attack [12] is a technique to retrieve the original data from the corrupted data. In this method, the encrypted image is cropped into blocks sized 10%, 25%, 50% and 75% of the original encrypted image; then, the proposed decryption algorithm is applied to these cropped images. Figure 12(a-d) displays the various sizes of cropped images, while Figure 12(e-h) presents the decrypted

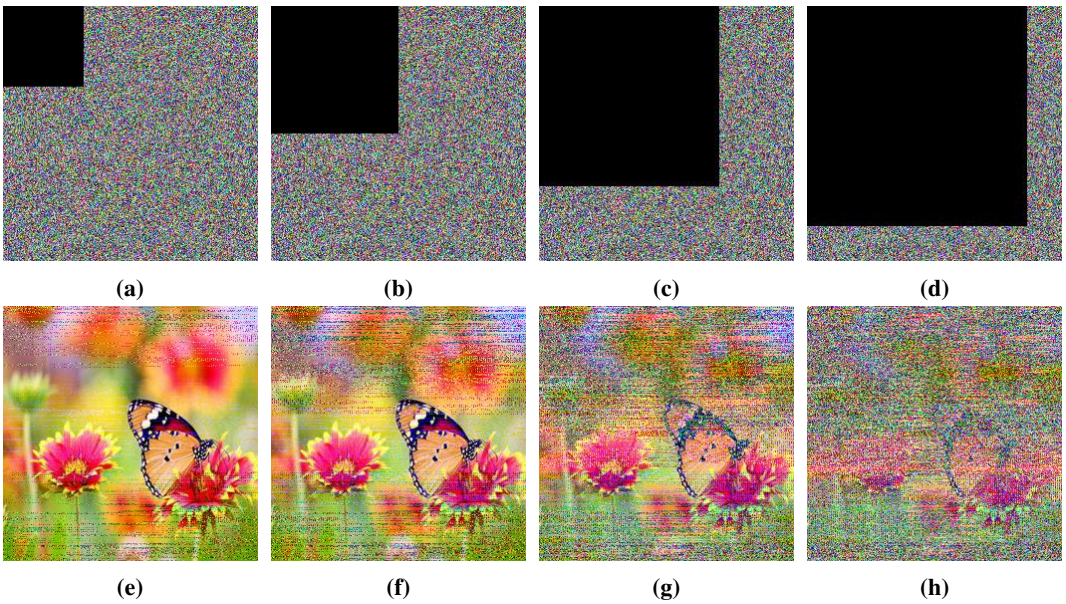

Figure 12: (a-d) shows the encrypted image of Butterfly with 10%, 25%, 50%, 75% cropped. (e-h) shows the corrosponding decrypted image of Butterfly with 10%, 25%, 50%, 75% cropped.

images corresponding to these cropped images. Figure 12(e-h) clearly shows that the decrypted images are comprehensible and visible to the human eye. Therefore, the proposed algorithm is resistant to occlusion attacks.

6.10 Differential attack analysis

To prevent known plaintext attacks, a small alteration in the original image should result in a substantial change in the encrypted image. Two metrics are used to quantify this requirement are Number of Pixels Changing Rate (NPCR), and Unified Average Changed Intensity (UACI). Both metrics are used to measure the effectiveness of differential attacks. NPCR is a percentage that indicates how many changes occur in the encrypted image when there is a 1-bit change in the pixel values of the original image. In contrast, UACI measures the difference between the pixel values of these encrypted images. To determine NPCR and UACI, a comparison is made between the encrypted version of the original image and the encrypted version of the image where one bit in the pixel values has been altered.

The NPCR and UACI are derived as

$$NPCR = \frac{1}{m \times n} \times \sum_{i,j} D_1(i,j) \times 100\%, \tag{6.8}$$

$$and \qquad UACI = \frac{1}{m \times n} \left[\sum_{i,j} \frac{|C(i,j) - C'(i,j)|}{255} \right] \times 100\%. \tag{6.9}$$

The expression below gives $D_1(i,j)$

$$D_1(i,j) = \begin{cases} 1, & if \;\; C(i,j) \neq C'(i,j) \\ 0, & elsewere, \end{cases} \tag{6.10}$$

where mn represents the total number of pixels in the image. $C(i,j)$ and $C'(i,j)$ denote the encrypted images obtained by altering one bit in the pixel value of the original image. For an

image, the standard theoretical values of NPCR and UACI are 99.6094% and 33.4635%, respectively. The values observed in Table 10 closely align with the standard NPCR and UACI values, affirming the robustness of the anti-differential scheme.

Table 10: NPCR and UACI analysis

Image	NPCR				UACI			
	R	G	B	Average	R	G	B	Average
Butterfly	99.8642	99.8642	99.8642	99.8642	33.1093	33.3295	33.4857	33.3081
Female	99.6292	99.6445	99.5911	99.6216	33.4050	33.4641	33.4935	33.4542
House	99.6033	99.6002	99.5895	99.5976	33.3969	33.4402	33.2967	33.3779
Jelly	99.5041	99.6307	99.6460	99.5936	33.3727	33.5128	33.4886	33.4580

6.11 Run's test for randomness

The Run's test assesses the randomness of an event's probability. The Run's test [34] is commonly applied to determine the randomness of encrypted image data. If the data in the encrypted image is random, the test yields a zero result. Conversely, if the data is not random, it returns a value of one. We utilized Matlab software to apply the Run's test to our data. The results for all encrypted images using our proposed algorithm are showing a value of zero, as presented in Table 11. The table results indicate that the proposed algorithm achieves a high level of randomness in the encrypted images.

Table 11: Run's test on encrypted test images

Image (256×256)		Butterfly	Female	House	jelly
Rsults for Run test	R	0	0	0	0
	G	0	0	0	0
	B	0	0	0	0
Randomness		Pass	Pass	Pass	Pass

6.12 Plaintext and ciphertext attacks

Attackers conduct cryptographic attacks to retrieve plaintext from encrypted data. Several possible attacks [19] can be employed to break a cryptosystem, a few are listed below:
Ciphertext-only attack: In these types of attacks, the attacker(s) is aware of the proposed encryption scheme and possesses a set of ciphertexts. They aim to discover the original secret key from this set to decode the encrypted image.
Known plaintext attack: In this attack, the attacker(s) endeavour to deduce the original secret key by analyzing certain plaintexts and their corresponding ciphertexts.
Chosen plaintext attack: In this attack, the attacker(s) attempt to uncover the original secret key by selecting plaintexts of their choice. This represents the most resilient and formidable attack.
Chosen ciphertext-only attacks: In these attacks, the attacker(s) select a list of ciphertexts and attempt to obtain their corresponding plaintexts. This form of attack is feasible in asymmetric key algorithms.
Based on the key sensitivity analysis, the proposed algorithm demonstrates resilience against minor alterations in the secret key. Therefore, these types of attacks are ineffective against the proposed algorithm.

6.13 Encryption time analysis

Encryption time is the duration required to encrypt data using a cryptographic algorithm. The time taken for encryption depends on various factors, including the algorithm used, the length of the key, and the available computational resources. The proposed algorithm was implemented using Matlab 2021b on a system equipped with an Intel (R) Core i5-9300 @ 2.40 GHz processor and 8GB of RAM and tested on a $256 \times 256 \times 3$ image. The encryption time for the proposed algorithm, applied to various plain test images, is presented in Table 12. Table 13 indicates that the proposed algorithm consumes a significantly shorter encryption time than other algorithms. The encryption time can also be reduced by utilizing appropriate hardware and software on high-end computers.

Table 12: Encryption time analysis

Images	Size	Encryption time (s)
Buterfly	$256 \times 256 \times 3$	0.133568
Female	$256 \times 256 \times 3$	0.126869
House	$256 \times 256 \times 3$	0.133973
Jelly	$256 \times 256 \times 3$	0.134247

Table 13: Comparative analysis for encryption time with some different techniques

Images	Size	Encryption time (s)
ours	$256 \times 256 \times 3$	0.133568
[55]	256×256	1.170844
[56]	$256 \times 256 \times 3$	5.55679
[35]	$256 \times 256 \times 3$	47.367

7 Conclusion remarks

This paper proposes a faster, highly secure, and efficient algorithm that enhances encryption quality by utilizing RC4A and RMAC cryptosystems along with two chaotic maps: 2-D Henon map and 1-D Tent map. The algorithm is a multistage encryption process consisting of four stages. In the first encryption stage, the RC4A cryptosystem diffuses the plain image, resulting in a partially encrypted image I_1. In the next stage, the algorithm employs a 2-D Henon map to confuse the image pixels of the partially encrypted image I_1, yielding image I_2. In the next stage, the algorithm simultaneously confuses and diffuses the partially encrypted image I_2 by using the RMAC cryptosystem. This process involves confusing and diffusing even rows, odd rows, even columns, and odd columns with different RMAC parameters and produces image I_3. In the final encryption stage, the algorithm uses a 1-D Tent map to diffuse the image I_3, resulting in a finally encrypted image.

The proposed algorithm undergoes various tests and analyses. The numerical and visual results underscored the algorithm's efficiency, demonstrating its resilience against statistical, differential, and other attacks. Statistical analyses, including key space analysis, key sensitivity, correlation analysis, histogram analysis, and chi-square analysis, demonstrate the proposed algorithm's robustness against different statistical attacks. Also, the differential attacks, such as NPCR, UACI analysis, and run's randomness test, are conducted to check the robustness of the scheme. Additional quality measures, including entropy analysis, PSNR, SSIM, MSE, occlusion, and noise attacks are performed. These analyses demonstrate that the proposed algorithm is efficient, highly secure, and suitable for image security applications.

8 Future plan

Nowadays a lot of digital data being transmitted over the internet. A number of image encryption scheme have been proposed in the literature. An important issue is related to the secure and fast communication of digital data. In future, our aim is to minimize the total computational cost by using some more suitable tools and mathematical approaches. Also, we will work to explore other chaotic maps to design efficient and secure image encryption algorithms. We will focus on the Internet of Things (IoT) and seek to identify secure encryption techniques suitable for these devices. We will explore more efficient and reliable techniques for handling medical images.

References

[1] Dena Alani and Salah Al Iesawi, *Image encryption algorithm based on rc4 and henon map* (2018).

[2] Wassim Alexan, Marwa Elkandoz, Maggie Mashaly, Eman Azab, and Amr Aboshousha, *Color image encryption through chaos and kaa map*, Ieee Access **11** (2023), 11541–11554.

[3] Gonzalo Alvarez and Shujun Li, *Some basic cryptographic requirements for chaos-based cryptosystems*, International journal of bifurcation and chaos **16** (2006), no. 08, 2129–2151.

[4] Long Bao and Yicong Zhou, *Image encryption: Generating visually meaningful encrypted images*, Information Sciences **324** (2015), 197–207.

[5] Anirban Bhowmick, Nishith Sinha, R Vijaya Arjunan, and B Kishore, *Permutation-substitution architecture based image encryption algorithm using middle square and rc4 prng*, 2017 international conference on inventive systems and control (icisc), 2017, pp. 1–6.

[6] Matthew Curtis and Gurudeo Anand Tularam, *The importance of numbers and the need to study primes: The prime questions*, Journal of Mathematics and Statistics **7** (2011), no. 4, 262–269.

[7] Osama S Faragallah, Ashraf Afifi, Walid El-Shafai, Hala S El-Sayed, Mohammed A Alzain, Jehad F Al-Amri, and Fathi E Abd El-Samie, *Efficiently encrypting color images with few details based on rc6 and different operation modes for cybersecurity applications*, IEEE Access **8** (2020), 103200–103218.

[8] Abdul Gaffar, AB Joshi, and Dhanesh Kumar, *Image encryption using gingerbreadman map and rc4a stream cipher*, Applications and Applied Mathematics: An International Journal (AAM) **15** (2020), no. 2, 30.

[9] Sneha Ghoradkar and Aparna Shinde, *Review on image encryption and decryption using aes algorithm*, International Journal of Computer Applications **975** (2015), 8887.

[10] Riah Ukur Ginting and Rocky Yefrenes Dillak, *Digital color image encryption using rc4 stream cipher and chaotic logistic map*, 2013 international conference on information technology and electrical engineering (icitee), 2013, pp. 101–105.

[11] Rafael C Gonzalez, *Digital image processing*, Pearson education india, 2009.

[12] Vandana Guleria, Shazia Sabir, and Deep Chandra Mishra, *Security of multiple rgb images by rsa cryptosystem combined with frdct and arnold transform*, Journal of Information Security and Applications **54** (2020), 102524.

[13] Gururaj Hanchinamani and Linganagouda Kulkarni, *An efficient image encryption scheme based on a peter de jong chaotic map and a rc4 stream cipher*, 3D Research **6** (2015Jul), no. 3, 30.

[14] Ehsan Hasanzadeh and Mahdi Yaghoobi, *A novel color image encryption algorithm based on substitution box and hyper-chaotic system with fractal keys*, Multimedia Tools and Applications **79** (2020), 7279–7297.

[15] Israel Nathan Herstein, *Topics in algebra*, John Wiley & Sons, 1991.

[16] Shamsa Kanwal, Saba Inam, Fahima Hajjej, Omar Cheikhrouhou, Zainab Nawaz, Ayesha Waqar, and Majid Khan, *A new image encryption technique based on sine map, chaotic tent map, and circulant matrices*, Security and Communication Networks **2022** (2022), no. 1, 4152683.

[17] Shamsa Kanwal, Saba Inam, Mohamed Tahar Ben Othman, Ayesha Waqar, Muhammad Ibrahim, Fariha Nawaz, Zainab Nawaz, and Habib Hamam, *An effective color image encryption based on henon map, tent chaotic map, and orthogonal matrices*, Sensors **22** (2022), no. 12, 4359.

[18] Gurpreet Kaur, Rekha Agarwal, and Vinod Patidar, *Color image encryption scheme based on fractional hartley transform and chaotic substitution–permutation*, The visual computer **38** (2022), no. 3, 1027–1050.

[19] Jan Sher Khan and Jawad Ahmad, *Chaos based efficient selective image encryption*, Multidimensional Systems and Signal Processing **30** (2019), 943–961.

[20] Manoj Kumar and Manoj Diwakar, *A new exponentially directional weighted function based ct image denoising using total variation*, Journal of King Saud University-Computer and Information Sciences **31** (2019), no. 1, 113–124.

[21] Manju Kumari and Shailender Gupta, *A novel image encryption scheme based on intertwining chaotic maps and rc4 stream cipher*, 3D Research **9** (2018), 1–20.

[22] Chunhu Li, Guangchun Luo, Ke Qin, and Chunbao Li, *An image encryption scheme based on chaotic tent map*, Nonlinear Dynamics **87** (2017), 127–133.

[23] Li Li, Ahmed A Abd El-Latif, and Xiamu Niu, *Elliptic curve elgamal based homomorphic image encryption scheme for sharing secret images*, Signal Processing **92** (2012), no. 4, 1069–1078.

[24] Shih-Yu Li, Yanwen Gai, Kai-Chung Shih, and Chin-Sheng Chen, *An efficient image encryption algorithm based on innovative des structure and hyperchaotic keys*, IEEE Transactions on Circuits and Systems I: Regular Papers (2023).

[25] Xiaofeng Li and Yinhui Zhang, *Digital image encryption and decryption algorithm based on wavelet transform and chaos system*, 2016 IEEE advanced information management, communicates, electronic and automation control conference (imcec), 2016, pp. 253–257.

[26] Kuang Tsan Lin, *Image encryption using arnold transform technique and hartley transform domain*, 2013 ninth international conference on intelligent information hiding and multimedia signal processing, 2013, pp. 84–87.

[27] Parveiz Nazir Lone and Deep Singh, *Application of algebra and chaos theory in security of color images*, Optik **218** (2020), 165155.

[28] Parveiz Nazir Lone, Deep Singh, and Umar Hussain Mir, *A novel image encryption using random matrix affine cipher and the chaotic maps*, Journal of Modern Optics **68** (2021), no. 10, 507–521.

[29] Parveiz Nazir Lone, Deep Singh, and Umar Hussain Mir, *Image encryption using DNA coding and three-dimensional chaotic systems*, Multimedia tools and Applications **81** (2022), no. 4, 5669–5693.

[30] Qing Lu, Congxu Zhu, and Xiaoheng Deng, *An efficient image encryption scheme based on the lss chaotic map and single s-box*, IEEE Access **8** (2020), 25664–25678.

[31] Yuling Luo, Xue Ouyang, Junxiu Liu, and Lvchen Cao, *An image encryption method based on elliptic curve elgamal encryption and chaotic systems*, IEEE Access **7** (2019), 38507–38522.

[32] Anjali Malik, Sangeeta Dhall, and Shailender Gupta, *An improved bit plane image encryption technique using rc4 and quantum chaotic demeanour*, Multimedia Tools and Applications **80** (2021), no. 5, 7911–7937.

[33] Umar H Mir, Deep Singh, DC Mishra, and Parveiz N Lone, *Multilayer security of rgb image in discrete hartley domain*, Applications and Applied Mathematics: An International Journal (AAM) **15** (2020), no. 2, 29.

[34] Umar Hussain Mir, Deep Singh, and Parveiz Nazir Lone, *Color image encryption using rsa cryptosystem with a chaotic map in hartley domain*, Information Security Journal: A Global Perspective **31** (2022), no. 1, 49–63.

[35] Rose Mohammed and L Mohammed Jawad, *Secure image encryption scheme using chaotic maps and rc4 algorithm*, Solid State Technology **63** (2020), no. 3, 3465–3449.

[36] HT Panduranga and Naveen Kumar SK, *Advanced partial image encryption using two-stage hill cipher technique*, International journal of computer applications **60** (2012), no. 16.

[37] Ping Ping, Feng Xu, Yingchi Mao, and Zhijian Wang, *Designing permutation–substitution image encryption networks with henon map*, Neurocomputing **283** (2018), 53–63.

[38] Dian Rachmawati, Mohammad Andri Budiman, and Muhammad Ishan Wardhono, *Hybrid cryptosystem for image security by using hill cipher 4x4 and elgamal elliptic curve algorithm*, 2018 IEEE international conference on communication, networks and satellite (comnetsat), 2018, pp. 49–54.

[39] Shazia Sabir and Vandana Guleria, *Multi-layer color image encryption using random matrix affine cipher, rp2dfrht and 2d arnold map*, Multimedia Tools and Applications **80** (2021), no. 18, 27829–27853.

[40] Aradhana Sahoo, Pratyasha Mohanty, and Purna Chandra Sethi, *Image encryption using rsa algorithm*, Intelligent systems: Proceedings of icmib 2021, 2022, pp. 641–652.

[41] Priya R Sankpal and PA Vijaya, *Image encryption using chaotic maps: a survey*, 2014 fifth international conference on signal and image processing, 2014, pp. 102–107.

[42] VM Silva-García, R Flores-Carapia, I López-Yáñez, and C Rentería-Márquez, *Image encryption based on the modified triple-des cryptosystem*, International mathematical forum, 2012, pp. 2929–2942.

[43] Purushottam Singh, Sandip Dutta, and Prashant Pranav, *A modified rc-4 cryptosystems to enhance security by using negative key schedule*, Security and Privacy, e438.

[44] Xingyuan Wang and Chuanming Liu, *A novel and effective image encryption algorithm based on chaos and DNA encoding*, Multimedia Tools and Applications **76** (2017), 6229–6245.

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, *Image quality assessment: from error visibility to structural similarity*, IEEE transactions on image processing **13** (2004), no. 4, 600–612.

[46] Yue Wu, Yicong Zhou, George Saveriades, Sos Agaian, Joseph P Noonan, and Premkumar Natarajan, *Local shannon entropy measure with statistical tests for image randomness*, Information Sciences **222** (2013), 323–342.

[47] Abolfazl Yaghouti Niyat and Mohammad Hossein Moattar, *Color image encryption based on hybrid chaotic system and DNA sequences*, Multimedia Tools and Applications **79** (2020), no. 1, 1497–1518.

[48] Guodong Ye, Kaixin Jiao, Huishan Wu, Chen Pan, and Xiaoling Huang, *An asymmetric image encryption algorithm based on a fractional-order chaotic system and the rsa public-key cryptosystem*, International Journal of Bifurcation and Chaos **30** (2020), no. 15, 2050233.

[49] Jinwei Yu, Wei Xie, Zhenyu Zhong, and Huan Wang, *Image encryption algorithm based on hyperchaotic system and a new DNA sequence operation*, Chaos, Solitons & Fractals **162** (2022), 112456.

[50] Qiang Zhang, Ling Guo, and Xiaopeng Wei, *Image encryption using DNA addition combining with chaotic maps*, Mathematical and Computer Modelling **52** (2010), no. 11-12, 2028–2035.

[51] Xiaoqiang Zhang and Xuesong Wang, *Digital image encryption algorithm based on elliptic curve public cryptosystem*, IEEE Access **6** (2018), 70025–70034.

[52] Yong Zhang, Xueqian Li, and Wengang Hou, *A fast image encryption scheme based on aes*, 2017 2nd international conference on image, vision and computing (icivc), 2017, pp. 624–628.

[53] Jiming Zheng and Qingxia Zeng, *An image encryption algorithm using a dynamic s-box and chaotic maps*, Applied Intelligence **52** (2022), no. 13, 15703–15717.

[54] Congxu Zhu, Guojun Wang, and Kehui Sun, *Cryptanalysis and improvement on an image encryption algorithm design using a novel chaos based s-box*, Symmetry **10** (2018), no. 9, 399.

[55] Laiphrakpam Dolendro Singh and Khumanthem Manglem Singh, *Image encryption using elliptic curve cryptography*, Procedia Computer Science **54** (2015), 472–481.

[56] Junxin Chen, Yu Zhang, Lin Qi, Chong Fu, and Lisheng Xu, *Exploiting chaos-based compressed sensing and cryptographic algorithm for image encryption and compression*, Optics & Laser Technology **99** (2018), 238–248.

Author information

Deep Singh, Department of Mathematics and Statistics, Central University of Punjab, Bathinda, India.
E-mail: deepsinghspn@gmail.com

Manoj Kumar Patel, Department of Mathematics, National Institute of Technology, Nagaland, India.
E-mail: mkpitb@gmail.com

Mukesh*, Department of Mathematics and Statistics, Central University of Punjab, Bathinda, India.
E-mail: mukesh01259@gmail.com

Received: 2024-03-01
Accepted: 2024-08-25