# A Novel Approach for Image Encryption based on Parametric Mixing Chaotic System

Gururaj Hanchinamani
Computer Science Department
BVB College of Engineering and Technology, Hubli-580031
India

Linganagouda Kulakarni
Computer Science Department
BVB College of Engineering and Technology, Hubli-580031
India

## ABSTRACT
Advanced image encryption schemes for secure transmission and storage are increasingly needed for a number of applications like medical, military, satellite etc. In this paper, a novel image encryption algorithm based on Logistic and Tinkerbell map is proposed.The proposed method uses two 1-D Logistic maps with different keys and one 2-D Tinkerbell map. The chaotic sequence generated is mixed sequence from the $X$ and $Y$sequences of Tinkerbell map depending on the chaotic sequences of two logistic maps. The main advantage of such a scheme is complex chaotic behavior of the generated chaotic sequences. The security and performance of the proposed method is analyzed thoroughly by using key-sensitivity, key-space, statistical, entropy, differential and performance analysis. The proposed approach achieves the required level of security with only one round of encryption operation. Hence the proposed method is computationally efficient.

**Keywords:** Image encryption, Logistic map, Mixed maps, Tinkerbell map

## 1. INTRODUCTION
The inherent properties of images are massive volume of data, high correlation among adjacent pixels, high redundancy and human perception of decrypted image with small distortions. So images are different from text information. The conventional encryption methods such as AES, DES, IDEA, RSA etc. are computationally intensive hence consume more time and are not suitable for images [4,5,6,7]. There exist several image encryption algorithms in the literature, some of these suffer with brute-force attack, statistical attack, and differential attacks. In this paper, the performance and security of the encryption process is improved by mixing two chaotic sequences controlled by another two chaotic sequences.
The basic idea for image encryption in spatial domain is categorized into three types: (i) pixel position permutation (ii) pixel value modification and (iii) compounding forms [21]. Already there exist several image encryption methods in spatial domain, among which chaotic-based methods are most popular. Chaotic maps have been used as essential component to construct cryptosystems, as they possess several pretty properties, such as simplicity, randomness, sensitivity and ergodicity. Chaotic encryption systems generally have high speed with low cost, which makes them better candidate than conventional methods for multimedia data encryption.

The rest of the paper is organized as follows. In section 2, the literature survey is presented. The Logistic and Tinkerbell maps are discussed in section 3. In section 4, the proposed encryption scheme is described in detail. Simulation results and security analysis is presented in section 5 to show efficacy and validity of the algorithm. Finally, conclusions are drawn in the last section.

## 2. LITERATURE SURVEY
A chaotic-based image encryption schemes typically consists of iteration of two processes (i) permutation and (ii) diffusion. The permutation is achieved by scrambling all the pixels as a whole using 2D chaotic map (such as Baker map, Arnold cat map etc.)[3,4,10,12]. During diffusion, the pixel values are modified sequentially and the change made to a particular pixel depends on the accumulated effect of all the previous pixel values. However, as many rounds of permutation and diffusion or iterations should be taken, the overall encryption speed is slow.

A brief overview of the recently proposed chaotic based encryption schemes are given hereafter. Guodong Ye [1] proposed an image encryption scheme with generalized Arnold map, as the key stream depends on the processed image the method can resist known- and chosen-plain text attacks. To solve the problem of small key space multiple chaotic maps were used in [2,3,6]. Yanbing Liu [5] suggested energy efficient chaotic block cipher suitable for wireless sensor networks. In [7] better diffusion properties were achieved by using an improved hash-based image encryption algorithm. Homomorphism based image encryption scheme were studied in [9] with shorter key and better performance than schemes based on RSA. The authors of [12] combined the permutation and diffusion stages and suggested a fast method for image encryption. In [14], the computational time is reduced by encrypting significant data in spatial domain and insignificant data in wavelet domain. High-dimensional chaotic systems were further studied in [17] to enlarge the key-space for resisting the brute-force attack.

However, some of chaotic based encryption schemes have been successfully cryptanalyzed [8,11]. Liang Zhao et. al. [8] presented chosen-plaintext attack and chosen-ciphertext attack on [12], and proposed an improved image encryption scheme using self-correlations. Rhouma et.al.[11] presented attack on [25] with only one pair of plaintext and ciphertext.

Based on the above discussions, though there exist several encryption schemes, each of them has its own strength and limitations more or less in terms of security level and computational speed. To resist statistical, differential, brute-force attacks and to improve the computational performance, a novel chaotic image encryption scheme is proposed in this paper, in which the generated pseudorandom sequence is mixture of two chaotic sequences controlled by another two chaotic sequences. The main advantage of such a scheme is complex chaotic behavior of the generated chaotic sequences. The proposed method is resistant to brute-force attacks, statistical attacks and differential attacks with high computational speed. The proposed approach achieves the required level of security with only one round of encryption operation. It can be easily implemented and is computationally simple.

## 3. CHAOTIC MAPS

The chaotic maps are used to generate pseudorandom sequences. Chaotic maps are non-periodic, non-convergent, topologically mixing and sensitive to initial conditions and parameters. The following chaotic maps are used in the proposed algorithm.

### 3.1 Logistic map

Logistic map is a classical 1-D map, and is defined as

$$X_{n+1} = \mu X_n(1 - X_n) \qquad (1)$$

Here $X_n$ and $X_{n+1}$ are current and next chaotic values and the values lie in the range [0,1], μ is a control parameter and its range is $0 \leq \mu \leq 4$. When $0 \leq \mu \leq 3$, the sequence is stable. When the value of μ increases gradually, periodic behaviors can be observed from the sequence. When μ > 3.5699, periodicity disappears and chaos shows. The key set for Logistic map is $\{X_0, \mu\}$.

### 3.2 Tinkerbell map

The Tinkerbell map is a 2-D discrete-time dynamical system, and is defined as

$$X_{n+1} = X_n^2 - Y_n^2 + aX_n + bY_n$$

$$Y_{n+1} = 2X_nY_n + cX_n + dY_n \qquad (2)$$

Where $X_n$, $Y_n$ are current chaotic values and $X_{n+1}$, $Y_{n+1}$ are next chaotic values and $a, b, c, d$ are control parameters. Details of Tinkerbell map can be found in [22, 23]. The key set for Tinkerbell map is $\{X_0, Y_0, a, b, c, d\}$. Commonly used initial values and parameters are $a$= 0.9, $b$ = -0.6013, $c$ = 2.0, $d$=0.50, $X_0$ = -0.72 and $Y_0$ = 0.64.

## 4. PROPOSED ENCRYPTION SCHEME

This section introduces a new novel chaotic image encryption scheme called the Parametric Mixing Chaotic System (*PMCS*). The *PMCS* generates complex chaotic sequences, and these sequences are used in permutation and diffusion steps.

### 4.1 The *PMCS*

The structure of *PMCS* is simple. It is a combination of two1-D Logistic maps with different keys and one 2-D Tinkerbell map. Let $L1_i$ is the chaotic output of first Logistic map with key1. $L2_i$ is the chaotic output of second Logistic map with key2. $X_i$ is the $X$ chaotic output of Tinkerbell map. $Y_i$ is the $Y$ chaotic output of Tinkerbell map. And $M$ and $N$ are the number of rows and columns of the image.

#### *4.1.1 The PMCS for permutation*

Initially generate $M + N$ chaotic values of $L1_i, L2_i, X_i, Y_i$ and then $M + N$ values of $Z1i$ are generated as,

$$Z1_i = \begin{cases} X_i, & if\ L1_i \geq L2_i \\ Y_i, & Otherwise \end{cases} \qquad (3)$$

Here, $Z1_i$ is a mixture output sequence of $X_i$ and $Y_i$, and the two outputs of Logistic maps $L1_i$ and $L2_i$ acts as a control switch to select either $X_i$ or $Y_i$ of Tinkerbell map. The structure of generation of $Z1_i$is shown in Fig.1. The first $M$ chaotic values of $Z1_i$ are used for row scrambling and the next $N$ chaotic values are used for column scrambling.

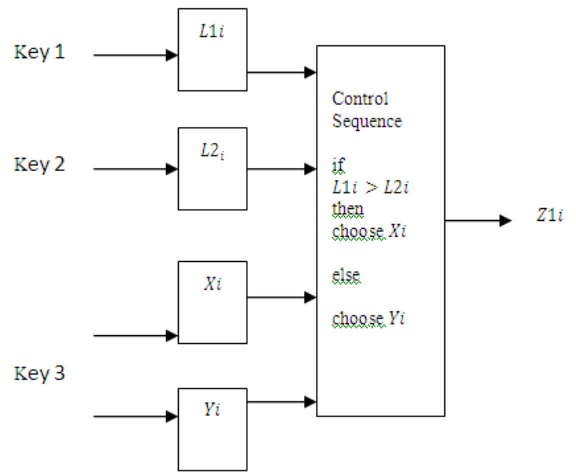


**Fig.1: Generation of $Z1_i$ *PMCS* chaotic sequence for permutation**

#### *4.1.2 The PMCS for diffusion*

Initially generate $M * N$ chaotic values of $L1_i, L2_i, X_i, Y_i$ and then $M * N$ values of $Z2_i$ are generated as,

$$Z2_i = \begin{cases} L1_i, & if\ X_i \geq Y_i \\ L2_i, & otherwise \end{cases} \qquad (4)$$

Here, $Z2_i$ is a mixture output sequence of $L1_i$ and $L2_i$ and the two outputs of Tinkerbell maps $X_i$ and $Y_i$ acts as a control switch to select either $L1_i$ or $L2_i$ of Logistic map. The structure of generation of $Z2_i$ is shown in Fig.2. The $M * N$ chaotic values of $Z2_i$ are used during diffusion process.

#### *4.1.3 Discussion*

As shown in Fig.1 and Fig.2, the proposed *PMCS* has a simple structure that combines two 1-D Logistic maps with different keys and one 2-D Tinkerbell map. The *PMCS* output is specified by six parameters ($\mu_1$, $\mu_2$, $a, b, c, d$) and four initial values ($X_0$ for first logistic map, $X_0$ for second logistic map, and $X_0, Y_0$ for Tinkerbell map). By embedding three maps, the *PMCS* shows more complex chaotic behavior than the existing ones. To quantitatively evaluate the chaotic behavior, the information entropy is used as a measure for the randomness of the output sequences (results are discussed in the next section). *PMCS* has larger information entropy values compared to the individual maps. The *PMCS* output is more randomly distributed. The proposed method is extremely sensitive to initial conditions and parameters. Small change ($10^{-10}$) of the parameter leads to completely different output sequence of the *PMCS*. The *PMCS* contains more parameters and initial values than the individual maps. This ensures more difficulty for unauthorized users to predict the *PMCS* output. So *PMCS* is more suitable for security applications. The propositions of chaotic maps [12] are given in Eq. (5-7). The *PMCS* chaotic output sequence is analyzed by computing mean and self-correlations according the propositions given in Eq. (5-7). It is observed that the mean value is close to 0.5 and the self correlations within the sequence and across two sequences are very close to 0.

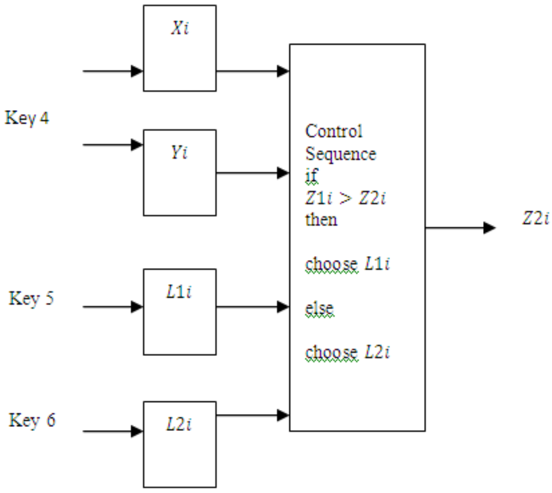


**Fig. 2: Generation of $Z2_i$ *PMCS* chaotic sequence for diffusion**

Proposition 1. The mean value of the chaotic sequence is

$$x_{mean} = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} x_k = 0.5 \qquad (5)$$

Proposition 2. Self-correlation of a chaotic sequence is

$$S1(\beta) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} (x_k - x_{mean})(x_{k+\beta} - x_{mean}) = 0 \qquad (6)$$

Proposition 3. Self-correlation function between two chaotic sequences is

$$S2(\beta) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} (x_k - x_{mean})(y_{k+\beta} - y_{mean}) = 0 \qquad (7)$$

## 4.2 Permutation and diffusions

The algorithm consists of two stages, i.e. permutation and diffusion.

### *4.2.1 Permutation*

Permutation is employed to reduce the high correlation between neighboring pixels in the plain image. Let *I* be a gray plain image of size $M \times N$, it is a digital matrix with $M$ rows and $N$ columns, in which the gray values ranges from 0 to 255. In the process of permutation, initially $M + N$ *PMCS* chaotic values $(Z1_1 - Z1_{M+N})$ are generated by using Eq. (3) after doing iterations in chaos maps. Let $TM = \{Z1_1..Z1_M\}$ and $TN = \{Z1_{M+1}..Z1_{M+N}\}$. Then $TM$ and $TN$ are transformed to integer sequence by using the following transform and stored in $TM'$ and $TN'$.

$$k_i' = (k_i * 10^8) \; mod \; m \qquad (8)$$

Where $k_i$ is real chaotic value, $k_i'$ is transformed integer value, and $m$ is 256 for 8-bit gray image.

Then $TM'$ and $TN'$ are indexed. The next step is to exchange row position of all values from first column to last column according to $TM_1', \dots, TM_M'$. Similarly exchange column position of all values from first row to last row according to $TN_1', \dots, TN_N'$. This process completely shuffles the pixels and decorrelates the adjacent pixels.

### *4.2.2 Diffusion*

The purpose of diffusion function is to modify the gray values of the image pixels to confuse the relationship between plain image and encrypted image. The requirement of diffusion is sensitivity to plain image i.e., a small change in any one pixel of plain image should spread out to almost all pixels in the whole image. First, the 2-D permuted image is transformed to 1-D array $O_{1\times MN}$ by scanning the image left to right and top to bottom. Diffusion of the permuted image is performed by using *PMCS* chaotic sequence and also previously diffused pixels. As the encrypted pixels depends on the previously encrypted pixels and chaotic sequence, the algorithm shows resistance to the differential attacks such as known plain-text attack and known cipher-text attack.
The forward diffusion is performed by using following equation,

$$E_i = ((O_i + E_{i-1}) mod \; 256) \oplus Z2_i, \quad i = 1,2,\dots,MN \qquad (9)$$

Where $+$ indicates addition and $\oplus$ is bitwise $XOR$, $E_i$ and $E_{i-1}$ are current and previous encrypted pixels and $O_i$ is permuted pixels and $Z2_i$ is the *PMCS* chaotic values. $E_0$ can be considered as a constant.

The backward diffusion is performed using Eq. (10), to make the influence of every pixel equal.

$$F_i = \big((E_i + F_{i+1}) \; mod \; 256\big) \oplus Z2_i, \; i = MN, MN-1, \dots, 1 \qquad (10)$$

Where $+$ indicates addition and $\oplus$ is bitwise $XOR$, $F_i$ and $F_{i+1}$ are current and previous encrypted pixels and $E_i$ is forward diffused image pixels and $Z2_i$ is the *PMCS* chaotic values. $E_{MN+1}$ can be considered as a constant. Finally, the encrypted image is obtained after the diffusions using Eq. (9) and Eq. (10) in two directions.

## 4.3 Algorithm

### *4.3.1 Encryption algorithm*

The *PMCS* is now integrated in the encryption algorithm. The encryption algorithm is composed of eleven steps.
**Step 1.** Read the original plain image and store the pixel values in the matrix $I_{M\times N}$.
**Step 2.** Generate $M + N$ *PMCS* chaotic values using Eq. (3)
**Step 3.** Copy first $M$ values of $Z1$ to $TM$ and next $N$ values to $TN$.
**Step 4.** Transform $TM$ and $TN$ to integer sequence using Eq. (8) to obtain $TM'$ and $TN'$. Then index $TM'$ and $TN'$.
**Step 5.** Scramble all the rows by using $TM'$.
**Step 6.** Scramble all the columns by using $TN'$.
**Step 7.**Transform 2-D permuted image to 1-D array i.e. dimension transform from $M \times N \; to \; 1 \times MN$
**Step 8.** Generate $M * N$ *PMCS* chaotic values using Eq. (4).
**Step 9.** Perform forward diffusion using Eq. (9).
**Step 10.** Perform backward diffusion using Eq. (10).
**Step 11.**Transform the 1-D encrypted array to 2-D array i.e. dimension transform from $1 \times MN \; to \; M \times N$.

### 4.3.2 Decryption Algorithm

Decryption involves restoring gray levels of the encrypted image. It is a simple inverse process of the proposed encryption algorithm.

## 5. EXPERIMENTS and SECURITY ANALYSIS

The proposed algorithm is implemented by using C on the Linux platform using a personal computer with an intel (R) Core(TM) i3-2120 CPU at 3.30 GHz with 2.91 GB of RAM. The initial parameters are randomly set to {first Logistic map $X_0$= 0.64, μ=3.591; second Logistic map $X_0$= 0.65, μ= 3.692; Tinkerbell map $X_0$= -0.72, $Y_0$= 0.64, $a$ = 0.9, $b$ = -0.6013, $c$ = 2.0, $d$ = 0.5}. Test images are 256X256 gray-scale images chosen from *USC-SIPI* image database. A good image encryption scheme should resist the attacks such as brute-force attacks, statistical attacks, differential attacks and so on. In this section, the properties of the proposed *PMCS* scheme are analyzed to show its effectiveness in resisting these attacks.

The proposed *PMCS* algorithm has been applied to several test images. Fig. 3 shows the encryption results of different images after applied only one round of encryption algorithm. The first row shows the original images, second row shows the encrypted images and the last row shows the decrypted images. The encrypted images are totally unrecognizable, random and noise-like images without any leakage of the original information. This demonstrates that the proposed algorithm can be used to protect various images for diverse protection. The decrypted images are exactly same as the original images.

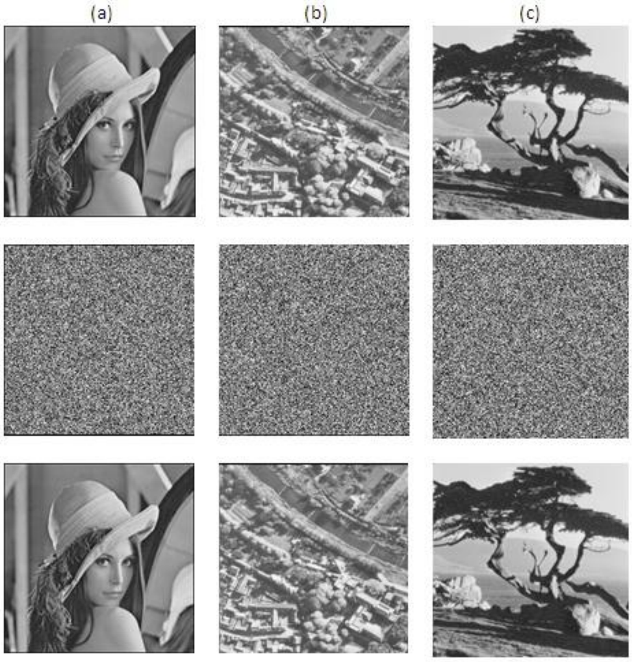


**Fig. 3: Original images, encrypted images and decrypted images with proposed *PMCS* algorithm (a) Lena image (b) aerial image (c) Tree image**

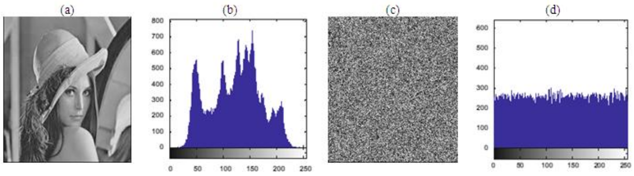


**Fig. 4: Histograms of original image and encrypted image. (a) original image (b) histogram of original image (c) encrypted image (d) histogram of encrypted image**

### 5.1 Histogram analysis

To prevent the leakage of information to an attacker, the encrypted image is expected to have no statistical similarity with the original image. An image histogram plots the number of pixels for each gray value. By looking at the histogram of an image a viewer will be able to judge the entire gray distribution. The histograms present the statistical characteristics of images. The histogram of original image consists of large spikes and will have some shape. These spikes correspond to gray values that appear more often in the image. The histogram of encrypted image is expected to be uniformly distributed to resist statistical attack. The histograms of several plain images and encrypted images are calculated and analyzed. The results for Lena image is shown in Fig. 4. The histogram of the encrypted image is uniformly distributed and is significantly different from that of the original image, and bear no statistical resemblance to the original image. Hence the proposed algorithm resists statistical attacks.

### 5.2 Key-space analysis

A brute-force attack is a method of breaking an encryption system by exhaustively searching all possible keys. The feasibility of a brute-force attack depends on the number of possible keys and on the amount of computational power available to the attacker. The key-space should be large enough to resist brute-force attacks. The proposed *PMCS* scheme makes use of four initial values and six parameters, hence the key comprises of totally ten real values { first Logistic map ($X_0$, μ) ; second Logistic map ($X_0$,μ) ; Tinkerbell map ($X_0$, $Y_0$, $a, b, c, d$) }. Each real value requires 64 bits and there are ten real values, so the key-length is 640 bits and the key-space is $2^{640}$. Hence the proposed algorithm has larger key-space and resists brute-force attacks.. Table 1 shows the key-space size of the proposed algorithm and other algorithms.

**Table 1. Key-space of the proposed method and some of the other methods in the literature**

| **Encryption scheme** | **Proposed *PMCS*** | **Ref.[2]** | **Ref.[12]** | **Ref.[18]** |
|---|---|---|---|---|
| Key-space size | $2^{640}$ | $2^{349}$ | $2^{128}$ | $2^{256}$ |

### 5.3 Information Entropy analysis

Entropy is a measure of unpredictability in information content, which quantifies the expected value of the information contained in a message. It is used to test whether an encrypted image is random-like image with pixel values randomly distributed. Suppose the gray level image has $2^8$ gray values with equal probabilities, $K = (K_0, K_1, K_2, \ldots, K_{255})$, according to Eq. (11), we obtain its entropy value $H(K) = 8$. In the original image, the entropy value is generally smaller than ideal value 8, because the pixel values are seldom random. Entropy is defined as,

$$H(K) = \sum_{i=0}^{q-1} P(K_i) log_2 \frac{1}{P(K_i)} \quad (11)$$

Where $q$ is the number of symbols, and is 256 for grayscale image. $K_i$ represents the pixel values, and $P(K_i)$ is the probability of the symbol $K_i$. Entropy reaches the maximum values when all pixel values are randomly distributed. Table 2 lists the entropy values for the original images and the encrypted images. From the results it is clear that the entropy of encrypted images are very close to the ideal value of 8. The information leakage in the proposed *PMCS* encryption scheme is negligible and the encryption scheme is secure against the entropy based attacks. Table 3 shows comparison of entropy values of the proposed method with other methods.

**Table 2. Entropy values for original and encrypted images for different images**

| Image | Entropy | |
|---|---|---|
| | **original image** | **encrypted image** |
| Lena | 7.426985 | 7.996877 |
| Aerial | 7.313656 | 7.997127 |
| Couple | 7.145428 | 7.997529 |
| Earth | 7.044457 | 7.997207 |
| Boat | 7.161232 | 7.996996 |
| House | 6.503890 | 7.997349 |
| Tree | 7.313894 | 7.997303 |
| Airport | 6.690835 | 7.996789 |
| Pepper | 7.577819 | 7.996747 |
| Toy Vehicle | 6.141587 | 7.997180 |

**Table 3. The entropy analysis of the proposed scheme with other methods for Lena image**

| Method | Entropy values |
|---|---|
| Proposed *PMCS* | 7.996877 |
| Ref. [24] | 7.9884 |
| RC5 | 7.9812 |
| RC6 | 7.9829 |
| Ref. [25] | 7.9923 |

## 5.4 Correlation analysis

Generally, for a plain-image, each pixel is highly correlated with its adjacent pixels in horizontal, vertical and diagonal directions. An ideal encryption scheme should produce encrypted images with no such correlations in the adjacent pixels. The correlation coefficient of adjacent pixels is calculated according to Eq. (15).

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (12)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2 \quad (13)$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \quad (14)$$

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad (15)$$

Where $x$ and $y$ are adjacent pixels of original or encrypted images, $E(x)$ is the mean value, $D(x)$ is the deviation with respect to mean, $cov(x, y)$ is the covariance between adjacent pixels, and $r_{xy}$ is the correlation coefficient. 2048 pairs of adjacent pixels are randomly selected in horizontal, vertical and diagonal directions for original and encrypted images and their correlation coefficients are computed using Eq. (15). The Table 4 specifies the computed correlation coefficients of original and encrypted images for different images. From Table 4 it is observed that two adjacent pixels in the original image are highly correlated to each other, whereas the correlation coefficients obtained for encrypted images are close to zero. Hence the proposed *PMCS* scheme is resistant to statistical attacks. The correlation results are compared with other methods and the analysis is given in Table 5.

**Table 4. Correlation coefficients of adjacent pixels in different directions for original and encrypted images**

| Image | Correlation coefficients for original images | | |
|---|---|---|---|
| | **Horizontal** | **Vertical** | **Diagonal** |
| Lena | 0.977352 | 0.851794 | 0.761644 |
| Aerial | 0.876954 | 0.922625 | 0.820881 |
| Couple | 0.933642 | 0.887662 | 0.810929 |
| Earth | 0.937537 | 0.901347 | 0.843996 |
| Boat | 0.886796 | 0.879324 | 0.890380 |
| House | 0.946241 | 0.965549 | 0.866505 |
| Tree | 0.919143 | 0.957095 | 0.879163 |
| Airport | 0.905118 | 0.881506 | 0.742995 |
| Pepper | 0.967597 | 0.970009 | 0.972013 |
| Toy Vehicle | 0.887444 | 0.987668 | 0.949336 |
| **Image** | **Correlation coefficients for encrypted images** | | |
| | **Horizontal** | **Vertical** | **Diagonal** |
| Lena | -0.005527 | 0.008554 | 0.004630 |
| Aerial | -0.008521 | 0.004132 | 0.002172 |
| Couple | 0.014752 | -0.001751 | -0.020083 |
| Earth | 0.002863 | 0.028621 | 0.004007 |
| Boat | -0.005115 | -0.007043 | -0.007880 |
| House | 0.005228 | 0.008609 | 0.029511 |
| Tree | -0.033626 | 0.016171 | 0.025959 |
| Airport | -0.020627 | 0.008818 | -0.000523 |
| Pepper | -0.003333 | 0.034722 | 0.019418 |
| Toy Vehicle | 0.002611 | -0.018214 | -0.018248 |

**Table 5. The correlation analysis of the proposed scheme with other methods for Lena image**

| Method | Direction | | |
|---|---|---|---|
| | **Horizontal** | **Vertical** | **Diagonal** |
| Plain-image | 0.977352 | 0.851794 | 0.761644 |
| Proposed *PMCS* | -0.005527 | 0.008554 | 0.004630 |

| | | | |
|---|---|---|---|
| Ref.[25] | 0.0117 | 0.0102 | 0.0153 |
| Ref.[7] | 0.0089 | -0.0215 | -0.0074 |
| AES | -0.0160 | 0.8018 | -0.0140 |
| Chen’s | 0.0442 | 0.9728 | 0.0469 |
| Arnold’s | 0.0787 | -0.0793 | -0.0633 |

## 5.5 Gray Value Degree ($GVD$) analysis

In an image, the gray difference of a pixel with its four neighbors can be computed as follows.

$$G = \frac{\sum\ [I(m,n) - I(m',n')]^2}{4}\ , here\ (m',n') = \begin{cases} (m-1,n) \\ (m+1,n) \\ (m,n-1) \\ (m,n+1) \end{cases} \tag{16}$$

Where $I(m,n)$ denotes the pixel value in position$(m,n)$, and $I(m',n')$ denotes the pixel values of four neighbor pixels. The average neighborhood gray difference for the entire image can be computed by Eq. (17).

$$W(\ G(m,n)\ ) = \frac{\sum_{m=2}^{M-1}\sum_{n=2}^{N-1} G(m,n)}{(M-2)\times(N-2)} \tag{17}$$

Where $M$ and $N$ are the number of rows and columns of the image. And the gray value degree is defined by

$$GVD = \frac{W'\big(G(m,n)\big) - W\big(G(m,n)\big)}{W'\big(G(m,n)\big) + W\big(G(m,n)\big)} \tag{18}$$

Where $W'$ and $W$ denote the average neighborhood gray difference of original and encrypted images. Table 6 shows the gray value degree values for different images by the proposed *PMCS* scheme. It is observed that the value of gray degree in the proposed method is nearer to 1. Table 7 shows the comparison of $GVD$ with other methods.

**Table 6. Gray value degree values for different test images.**

| Image | $GVD$ value | Image | $GVD$ value |
|---|---|---|---|
| Lena | 0.962115 | House | 0.974234 |
| Aerial | 0.917029 | Tree | 0.931290 |
| Couple | 0.949203 | Airport | 0.936798 |
| Earth | 0.958555 | Pepper | 0.964125 |
| Boat | 0.946033 | Toy Vehicle | 0.994195 |

**Table 7. The $GVD$ analysis of the proposed scheme with other methods**

| Image | $GVD$ value | | |
|---|---|---|---|
| | Proposed *PMCS* | Arnold’s | Ref.[12] |
| Lena | 0.962115 | 0.89 | 0.954 |

## 5.6 Peak Signal to Noise Ratio ($PSNR$) analysis

By considering the original image as a signal and encrypted image as a noise, the objective evaluation of the encryption scheme can be done by computing $PSNR$. The $PSNR$ can be calculated by using the following formula,

$$PSNR = 20 \times log_{10}\left(\frac{255}{\sqrt{MSE}}\right) dB \tag{19}$$

Where $MSE$ is mean square error and is computed according to Eq. (20)

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(\ |I(i,j) - I'(i,j)|\ )^2 \tag{20}$$

Where $I(i,j)$ and $I'(i,j)$ are pixel values of original and encrypted images at position$(i,j)$. The $PSNR$ is calculated for different images and is shown in Table 8. The lower value of $PSNR$ indicates the difficulty in getting original image from encrypted image for attackers.

**Table 8. $PSNR$ values for different images**

| Image | $PSNR$ (dB) | Image | $PSNR$ (dB) |
|---|---|---|---|
| Lena | 9.234015 | House | 9.256644 |
| Aerial | 9.299457 | Tree | 8.111625 |
| Couple | 9.729434 | Airport | 8.851973 |
| Earth | 9.377149 | Pepper | 8.862039 |
| Boat | 9.383699 | Toy Vehicle | 7.824519 |

## 5.7 Key sensitivity analysis

Key sensitivity is extremely important for image encryption schemes. Key sensitivity means the change of a single bit in the secret key should produce completely different encrypted image. The key sensitivity test is performed with following steps.

**Step 1.** The original image is encrypted by using the key $K_1$

**Step 2.** The original image is encrypted again with a small change in the key $K_1$, i.e. $K_2$

**Step 3**. The two cipher images with slightly different keys are compared pixel by pixel to see the number of differing pixels.

The key sensitivity can be assessed by using $NPCR$ and $UACI$ parameters as given below.

$NPCR$ (Number of Pixels Change Rate) is used to measure the number of different pixels in two cipher images and is calculated with the following formula.

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\% \tag{21}$$

$$D(i,j) = \begin{cases} 1, & if\ C_1(i,j) \neq C_2(i,j) \\ 0 & ,Otherwise \end{cases} \tag{22}$$

Where $C_1$and $C_2$ are two ciphered images with slightly different keys $K_1$ and $K_2$. $C_1(i,j)$ and $C_2(i,j)$ are the gray-scale values of $C_1$ and $C_2$ at position $(i,j)$. $D$ is a bipolar array with same size as $C_1$ and $C_2$ and its values are either 0 or 1 based on Eq. (22).

$UACI$ (Unified Average Changing Intensity) is used to measure the average intensity difference between two cipher images and is given by,

$$UACI = \frac{1}{M \times N}\left[\sum_{ij}\frac{C_1(i,j) - C_2(i,j)}{255}\right] \times 100\% \tag{23}$$

To assess the influence of single bit change in the key for the encrypted images, the $NPCR$ and $UACI$ values are computed and the results are presented in Table 9. From Table 9 it is observed that $NPCR$ and $UACI$ values are close to their ideal values 99.6 and 33.4. This demonstrates the high key sensitivity of the proposed *PMCS* scheme.

**Table 9. Key sensitivity results**

| Encrypted Images | $NPCR$ (%) | $UACI$ (%) |
|---|---|---|
| Lena | 99.595642 | 33.459881 |
| Aerial | 99.592590 | 33.401413 |
| Couple | 99.613953 | 33.471210 |
| Earth | 99.644470 | 33.431862 |
| Boat | 99.645996 | 33.421654 |
| House | 99.620056 | 33.529099 |
| Tree | 99.629211 | 33.374493 |
| Airport | 99.613953 | 33.396263 |
| Pepper | 99.601746 | 33.575848 |
| Toy Vehicle | 99.649048 | 33.386688 |

Key sensitivity is further analyzed diagrammatically with the following scheme. The original key is modified with a small change and a different key is generated. The keys can be described as, original key $\boldsymbol{K_1}$= (0.64, 3.591, 0.65, 3.692, -0.72, 0.64, 0.9, -0.6013, 2.0, 0.5), and the slightly modified key $\boldsymbol{K_2}$= (**0.640000000001**, 3.591, 0.65, 3.692, -0.72, 0.64, 0.9, -0.6013, 2.0, 0.5). Let $C_1$and $C_2$ are cipher images encrypted with keys $K_1$ and $K_2$. The encrypted image obtained for the original encryption key $K_1$ and the slightly changed key $K_2$ are shown in Fig.5b-c, respectively. Even though both look alike, they are significantly different from each other. This can be verified by analyzing the difference image between $C_1$ and $C_2$. Fig. 5d shows the difference image $C_1 - C_2$. From the difference image it is observed that the most of the pixels in Fig.5d are non-zero, i.e. the difference is big enough.

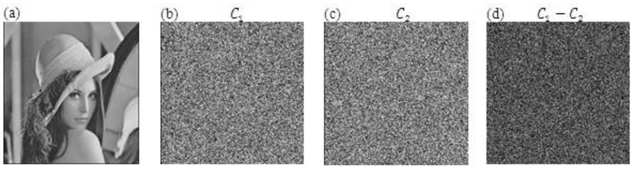


**Fig. 5: Key sensitivity analysis for encryption process for Lena image (a) original image (b) encrypted image with correct key (c) encrypted images with slightly different key (d) difference image between $\boldsymbol{C_1}$with $\boldsymbol{C_2}$**

The key sensitivity test is also conducted for decryption process. Fig.6a shows the decrypted image with correct key $\boldsymbol{K_1}$ = (0.64, 3.591, 0.65, 3.692, -0.72, 0.64, 0.9, -0.6013, 2.0, 0.5) and Fig.6b-c are decrypted images with slightly modified keys $\boldsymbol{K_2}$ = (**0.640000000001**, 3.591, 0.65, 3.692, -0.72, 0.64, 0.9,-0.6013, 2.0, 0.5), $\boldsymbol{K_3}$ = (0.64, 3.591, 0.65, 3.692, -0.72, 0.64, **0.90000000001**, -0.6013, 2.0, 0.5). Hence, the correct decryption cannot be achieved even when there is a small alteration in any one parameter of the of the decryption key.

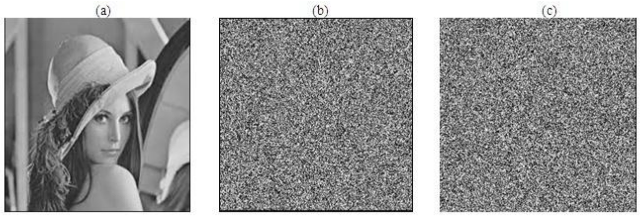


**Fig. 6: Key sensitivity analysis for decryption process for Lena image. (a) decryption with correct key (b-c) decryption with slightly changed keys**

## 5.8 Plain-image sensitivity analysis

Plain-image sensitivity means a small change in the plain-image will cause a great change in the cipher-image. Plain-image sensitivity is tested by performing $NPCR$ and $UACI$ analysis as given in Eq. (21-23). Table 10 shows the $NPCR$ and $UACI$ values calculated at randomly selected positions for the Lena image. The average values of $NPCR$ and $UACI$ achieved are 99.817505 and 33.438884. It is observed that the $NPCR$ and $UACI$ values are close to their ideal values irrespective of the pixel position selected. Thus the proposed *PMCS* scheme has high sensitivity to plain-images and resists differential attacks.

**Table 10. Plain-image sensitivity test at different positions for Lena image**

| Position | $NPCR$ (%) | $UACI$ (%) |
|---|---|---|
| (0,0) | 99.909973 | 33.399666 |
| (20,199) | 99.807739 | 33.456295 |
| (55,120) | 99.877930 | 33.496948 |
| (90,15) | 99.739075 | 33.424767 |
| (128,128) | 99.668884 | 33.423607 |
| (150,255) | 99.816895 | 33.394123 |
| (177,100) | 99.884033 | 33.404869 |
| (199,125) | 99.882507 | 33.620941 |
| (235,150) | 99.865723 | 33.257824 |
| (255,255) | 99.722290 | 33.509804 |
| Average values | 99.817505 | 33.438884 |

## 5.9 Computational speed analysis

The proposed *PMCS* scheme does not contain time-consuming calculations. Therefore the proposed scheme can offer a fast and efficient way for image encryption. The time complexity is $O(M \times N)$, where $M$ and $N$ are the number of rows and columns of the image. The time taken to encrypt $256 \times 256$ image is 3.43 micro-seconds and for decryption it is the same. So our scheme can encrypt 19106 Mbytes of data per second. Table 11 shows the comparison of our algorithm with the other methods having similar structure of encryption. Hence the proposed *PMCS* scheme has good time efficiency.

**Table 11. Execution time analysis using a 256x256 image**

| Methods | Encryption time |
|---|---|
| Proposed *PMCS* scheme | 3.43 micro seconds |
| Guodong Ye | 0.150 seconds |
| Gao. T.G | 0.633 seconds |
| Ye.R.S | >10 seconds |
| Huang X.L. | 0.547 seconds |

## 6. CONCLUSIONS

This paper proposes a new image encryption algorithm based on parametrically mixing chaotic system by using two 1-D Logistic maps with different keys and one 2-D Tinkerbell map.

The proposed method has high speed, high security level and is easily implementable. The *PMCS* is implemented using C on Linux platform and the speed achieved is 3.43 μs, so the proposed scheme is computationally efficient compared to other schemes in the literature. The key space of the proposed scheme is $2^{640}$, which is large enough to resist brute-force attacks. The average entropy achieved is 7.997110 which is close to its ideal value of 8. The correlations are close to zero, the $GVD$ is near to 1, the $PSNR$ is lower. The key sensitivity parameters $NPCR$ and $UACI$ are close to their ideal values of 99.6% and 33.4%. The *PMCS* has high plain-image sensitivity. So the proposed scheme is resistant to statistical and differential attacks. The security results presented in section 5 are the results after one round of encryption/decryption operation. So it is observed that even in the first round all the security parameters are already high. This scheme can be extended for color images and *PMCS* can also be implemented with other chaotic maps.

# 7. REFERENCES

[1] Guodong Ye, et.al. "An efficient chaotic image encryption algorithm based on a generalized Arnold map", Nonlinear Dynamics: Springer, 2012, pp. 2079-2087

[2] Ahmed A. Abd El-Latif et.al. "Digital image encryption scheme based on multiple chaotic systems", Sensing and Imaging: An international journal on continuing subsurface sensing technologies and applications: Springer, June 2012, vol. 56, Issue 2, pp. 67-88

[3] Shatheesh Sam et.al. "A novel image cipher based on a mixed transformed logistic maps", Multimedia tools and applications: An international Journal: Springer, Jan 2012, vol. 56, Issue 2, pp. 315-330

[4] C.K.Huang et.al. "Implementation of gray image encryption with pixel shuffling and gray-level encryption by single chaotic system", Telecommunication systems: Springer, February 2013, vol. 52, Issue 2, pp. 563-571

[5] Yanbing Liu et.al. "Design and statistical analysis of a new chaotic block cipher for wireless sensor networks", Communications in nonlinear science and numerical simulation: Elsevier, August 2012,vol. 17, Issue 8, pp. 3267-3278

[6] M. Francois et.al. "A new image encryption scheme based on a chaotic function", Signal Processing; Image Communications: Elsevier, March 2012, vol.27, Issue 3, pp. 249-259

[7] Shaojiang Deng et.al. "Analysis and improvement of a hash-based image encryption algorithm", Communications in nonlinear science and numerical simulations: Elsevier, August 2011, vol. 16, Issue 8, pp. 3269-3278

[8] Liang Zhao et.al. "On the security analysis of an image scrambling encryption of pixel bit and its improved scheme based on self-correlation encryption", Communications in nonlinear science and numerical simulations: Elsevier, August 2012, vol. 17, Issue 8, pp. 3303-3327

[9] Li Li et.al. "Elliptic curve EIGamal based homomorphic image encryption scheme for sharing secret images", Signal Processing:Elsevier, April 2012, vol. 92, Issue 4, pp. 1069-1078

[10] Yicong Zhou et.al. "Image encryption using a new parametric switching chaotic system", Signal Processing:Elsevier, November 2013, vol. 93, issue 11, pp. 3039-3052

[11] Rhouma Rhouma et.al. "Cryptanalysis of a new substitution-diffusion based image cipher", Communications in nonlinear science and numerical simulations: Elsevier, July 2010, vol. 15, Issue 7, pp. 1887-1892

[12] Guodong Ye, "Image scrambling encryption algorithm of pixel bit based on chao map", Pattern recognition letters: Elsevier, April 2010, vol.31, Issue 5, pp. 347-354

[13] G.A.Satishkumar et.al. "A novel algorithm for image encryption by integrated pixel scrambling plus diffusion [IISPD] utilizing duo chaos mapping applicability in wireless system", Procedia computer science, World conference on information technology: Elsevier, 2011, vol. 3, pp. 378-387

[14] Nidhi Taneja et.al. "Combinational domain encryption for still visual data", Multimedia tools and applications An international journal: Springer, August 2012, vol. 159, Issue 3. pp. 775-793

[15] Linhua Zhang, "An image encryption approach based on chaotic maps", Chaos, Solitons and fractals: Elsevier, May 2005, vol. 24, Issue 3, pp. 759-765

[16] Shiguo Lian et.al. "A block cipher based on a suitable use of the chaotic standard map", Chaos, Solitons and fractals: Elsevier, October 2005, vol. 26, Issue 1, pp. 117-129

[17] Guanrong Chen et.al. "A symmetric encryption scheme based on 3D chaotic maps", Chaos, Solitons and fractals: Elsevier, July 2004, vol. 21, Issue 3, pp. 749-761

[18] D. Chattopadhyay et.al. "symmetric key chaotic image encryption using circle map", Indian journal of science and technology, May 2011, vol.4, pp. 593-599

[19] Mohammad Ali Bani Younes et.al. "An image encryption approach using a combination of permutation technique followed by encryption", International journal of computer science and network security, April 2008, vol. 8, pp. 191-197

[20] Ji won Yoon, "An image encryption scheme with a pseudorandom permutation based on chaotic maps", Communications in nonlinear science and numerical simulations: Elsevier, December 2010, vol. 15, Issue 12, pp. 3998-4006

[21] Xiaofeng Liao et.al. "A novel image encryption algorithm based on self-adaptive wave transmission", Signal processing: Elsevier, September 2010, vol. 90, Issue 9, pp. 2714-2722

[22] Alexandre Goldsztejn et.al. "Tinkerbell is chaotic*", SIAM Journal applied dynamical system, 2011, vol. 10, No. 4, pp. 1480-1501

[23] Iftichar M.T. Al-Shara'a et.al."The dynamics of the 2-D piecewise Tinkerbell map", Mathematical theory and modeling, IISTE, 2013, vol. 3, No.8. pp. 121-132

[24] Pareek NK et.al."Image encryption using chaotic logistic map", Image Vision and computing, 2006, 24, pp. 926-934

[25] Patidar V et.al."A new substitution diffusion based image cipher using chaotic standard and logistic maps", Communications in nonlinear science and numerical simulations: Elsevier, 2009, 14, pp. 3056-3075

## 8. AUTHORS

**Linganagouda Kulakarni**

He received PhD degree in pattern recognition from Mysore university, India. His research interests are image processing, computer networks and information security. He is currently working as professor at computer science department, BVB college of engineering and technology, Hubli, India.

**Gururaj Hanchinamani**

He received ME degree in computer science and engineering from Walchand college of engineering Sangli, India. He is pursuing PhD degree at Visvesvaraiah technological university Belgaum. His research interests are information security and computer architectures. He is currently working as associate professor at computer science department, BVB college of engineering and technology, Hubli, India.