International Journal of Engineering and Technology Innovation, vol. 8, no. 2, 2018, pp. 77-98

A Review of Image Scrambling Technique Using Chaotic Maps

Shafali Agarwal*

Independent Researcher, Los Angeles, California, USA.

Received 12 September 2017; received in revised form 09 November 2017; accepted 07 December 2017

Abstract

Image scrambling is a process to convert a meaningful image into an unidentifiable or an unordered image by changing the pixel position such that an unauthorized user should not be able to access the original contents. The paper summarizes the various chaotic maps based scrambling methods with their performance measures. The discussion focuses on the scrambling techniques like Arnold map, Affine map, Baker map, Henon map, Logistic map, Fibonacci sequence, Fibonacci-Lucas sequence, and Fibonacci P-code transformation used to shuffle the image pixels in various image encryption algorithms. Each scrambling method is observed by using the adjacent pixels correlation coefficient, NPCR and UACI, scrambling/unscrambling speed, mean value analysis, entropy and peak signal-noise ratio. A comparative table is used to represent the reasonable computational outcome of each analysis parameter. The relative analysis of the performance and security measure highlights the weakness and strength of discussing scrambling techniques and can be used in the case-based image cryptosystem in future research.

Keywords: image scrambling, chaotic map, performance parameters, scrambled image

1. Introduction

With the advancement of communication technologies, text and images are numerously exchanged over the network. Hence an effective and an efficient cryptographic method is required for secure image transmission and storage over the public system. Image encryption is a technique used to convert the original image into another image which is not identifiable by an unauthorized user [1-2]. This is a method of transforming the information embedded in a digital image to a non-recognizable form so that no one can access the data except those having details of the decryption method with a key required to decrypt the data. To achieve a computationally secure cryptosystem, an image encryption algorithm comprises of multiple phases such as permutation, substitution, diffusion, confusion etc.

Since the 1990s, many researchers have noticed about the chaotic system that exhibits the randomness property, unpredictability, and sensitiveness of the key towards initial value appropriate to design a secure cryptosystem. Later in 2000, Fidritch has patented a chaotic cryptosystem to encrypt the data such as image using two and three-dimensional chaotic Baker map [3]. In order to use the chaos in cryptography efficiently and effectively, chaotic maps are implemented to create confusion and diffusion between the image pixels. It helps to reduce the correlation between the adjacent image pixels to enhance the encryption efficiency. These maps can be categorized into two groups: 1D chaotic maps such as Logistic map, Sine map and tent map [4]. Due to the simple structure, their chaotic orbits and initial values may be estimated with the least efforts [5-7]. The other group of chaotic maps contains a high-dimensional chaotic map with rather complex structure and better chaotic performance, such as Arnold map, Henon Map, Lorenz system, etc. Many authors suggested various combinations of 1D-chaotic maps (Logistic map, Tent map, Sine map) to achieve an improved performance of the proposed cryptosystem [8-10].

* Corresponding author. E-mail address: shafali.agarwal@gmail.com

The author utilized chaos features to generate chaotic random phase masks and design a cryptosystem using Gyrator transform and Jigsaw transform. The complexity of the method lies in the decryption process. To decrypt the cipher image, correct rotation angles of the Gyrator transform, initial values of a chaotic map and the random permutation of the Jigsaw transform are required [11]. The Logistic map function is used three times to scramble the row coordinate, column coordinate and to diffuse the plain image respectively [12]. Recently, a noisy Logistic map with an additive system noise and Clifford strange attractor were suggested by the author to encrypt the images in Navy [13]. Besides all the above well-known chaotic maps, a new Beta chaotic map was introduced to generate a key sequence which is based on Beta function [14]. A multiphase symmetric key encryption algorithm was proposed by the author using finite field cosine transformation (FFCT) in which A fractal is used as a source of one-time-pad keystream, provides a secure cryptosystem [15]. A cryptosystem will be relatively more secure if a set of different keys is used to encrypt the plain image on each iteration [16].

An important tool in image encryption is scrambling deals with the change in position of the pixels and helps to minimize the correlation coefficient value [17]. If the correlation coefficient between an original image and an encrypted image is zero or near to zero, a hacker will be unable to guess the encryption method or key. Recently, authors [18] used DNA sequence as a secret key and implemented permutation process using Hao's fractal representation. They also used diffusion and scrambling to make the encryption process more secure and complicated.

A secure cryptosystem was designed by using diffusion and permutation in addition to multiple chaotic based circular mapping, provides many secret keys and key dependent pixel value replacement [19]. Sometimes, researchers analyzed the existing cryptographic algorithms [20] and suggested a better solution based on the outcome. Here in [21], cryptanalysis was carried out by the authors and concluded that the system is completely breakable under chosen plaintext attack. They suggested a more secure cryptosystem by introducing two additional phases, i.e. pixel shuffling phase and pixel encoding phase. Further diffusion process with a combination of a chaotic map was used to encrypt RGB images in [22]. The author used a 128-bit key to encrypt an image which is layered in red, green and blue channel. A repeated execution of diffusion, mixing and substitution process execution for each RGB layer resultant into a secure cryptosystem [23]. To improve the security of a cryptosystem, two chaotic dynamic state variables have been used for encrypting an image pixel each in permutation and diffusion process separately [24].

The author utilized the randomness and unpredictability features of chaos to encrypt the images. He applied Henon map and Lorentz map for pixel shuffling and calculated the correlation coefficient between the original image and cipher image. The result shows that the proposed algorithm is best suited for a wireless communication using any single map [25]. In the paper [26], an external secret key is used to encrypt an image. The author has applied both pixel substitution and pixel permutation process to get a secure cryptosystem. A feedback mechanism is applied to make it secure from the differential attack. An encryption key sequence has generated by utilizing piecewise linear chaotic map and proposed a stream cipher algorithm for color image encryption based on a one-time key and robust chaotic maps [27]. The author proposed a symmetric key image encryption algorithm [28] in which additive and affine encryption technique using six schemes of key sequence derived from a random sequence of cyclic elliptic curve points are discussed. The result concluded that the proposed cryptosystem is secure from statistical, brute-force and cryptanalytic attacks. A combination of one-time key based on the crossover operator, chaos and a secure hash algorithm (SHA-2) is employed to design a cryptosystem for the color image encryption [29]. In the paper [30], the proposed encryption method utilized the magic rectangle in addition to a traditional public key cryptography algorithm such as RSA.

The Authors are regularly proposing and implementing substitution and diffusion phases using various chaotic systems such as 2D Tent Cascade Logistic map, 2D-LSM (Logistic and Sine map), combined features of the 2D Logistic map, 2D Arnold map and Quantum chaotic map, parameter-varied Logistic chaotic map [31-34]. Sudoku puzzle is always known as a mathematical puzzle and the logic was further used as to represent the image matrix elements via sudoku matrix. In

continuation, the author developed an associated 2D parametric bijection used to design an effective Sudoku associated image scrambler without bandwidth expansion [35]. By using the concept of image filtering in a different manner, a new cryptosystem was proposed in 2017 using block-based scrambling and image filtering [36]. Besides of using scrambling of the plain image, it can also be utilized in the generation process of cipher key [37]. The concept of edge map is generally used in image enhancement, compression, edge detection and segmentation. Recently, edge map of a key image is generated using any of the available gradient operators such as Sobel, Roberts, Canny, and Prewit and subsequently used to encrypt a plain image [38]. Nowadays, researchers are working on encryption and authentication in a single pass known as authenticated encryption with associated data (AEAD). The author has proposed a chaos-based AEAD based on a single-key Even-Mansour generated using a chaotic tent map and random s-box for the Type-II generalized Feistel structure [39].

In the last few decades, many scrambling methods have been published as a significant phase of an image encryption algorithm. Here the emphasis is given only to the chaotic map based scrambling methods like Logistic map, Henon map, Fibonacci map, Baker map etc. and their performances. Due to the chaotic properties, these maps are very popular and effective in the design of a cryptosystem. Earlier, many researchers have been analyzed in the performance of the chaotic maps using their original function. To enhance the complexity of these maps or to make it non-vulnerable, authors have proposed modifications to the existing chaotic map function. The novelty of the paper lies in the implementation of the map based scrambling methods as proposed by the authors in its corresponding research paper.

The effect of each scrambler has been investigated on the color image as well as in the bitmap image of the same size. An effective encryption algorithm must resist all kinds of known attacks such as a plain text attack, ciphertext attack, Brute-force attack, differential attacks, etc. Several tests have been conducted on the scrambled and unscrambled images to study the statistical and sensitivity evaluation. Section "Evaluation of chaotic map" of the paper explored the various enlisting chaotic maps based scrambling method. In section "Performance comparison of chaotic maps", a comparison table is presented for statistical and performance analysis of all the discussed scrambling methods followed by a section "Summary" to put an overview of all the mentioned techniques. Finally, a conclusion is drawn in future work direction.

2. Evaluation of Chaotic Map

This paper analyzed the performance of the chaotic maps used for scrambling the image pixels irrespective of the further used encryption method in the corresponding image encryption algorithm. All implementations have been done in MATLAB 2016 with system configuration used in the analysis is Intel® Atom™ x7-z8700 CPU @1. 60GHz with 4 GB RAM.

2.1. 2D Arnold map

Arnold map is used to transform the digital image by changing the pixel position within the image thus preserving the area stretching. This transformation was discovered by V. Arnold [40] in 1968 using an image of a cat. Arnold map transformation is suitable for a squared image so before applying it, resize the image to $N*N$ dimension [41]. The mathematical description of Arnold map is as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix}^k \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod\ N \tag{1}$$

where x_{n+1} and y_{n+1} are scrambled pixel coordinates previously located at x_n and y_n with $N*N$ dimensions. The parameters p, q and k are positive integers working as control parameters [42]. Arnold map provides a complex scrambled image while applying this function iteratively k times. However, after a certain number of iterations, the image converted into its original form. There is also an inverse Arnold transformation function through which shuffled image could be converted into the inputted plain image.

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} pq+1 & -p \\ -q & 1 \end{bmatrix}^k \begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} mod\ N \quad (2)$$

The complexity of Arnold map is reconsidered by Ashkay Chopra *et. al.* [43] by additional parameters calculating steps to the existing one. The paper analyzes the implementation details of modified Arnold map and measured the performance of the given algorithm. The Author has calculated the value of control parameters by using the following functions:

$a = 13 + mod\ (sumR + pnum1, 29)$

$b = 7 + mod\ (sumG + pnum1, 47)$

$k_1 = 3 + mod\ (sumB + pnum1, 13)$

$k_2 = 4 + mod\ (pnum1, 11)$

$k_3 = 5 + mod\ (fnum1, 9)$

Here the values of *sumR*, *sumG*, *sumB*, *pnum1*, and *fnum1* are calculated as follows:

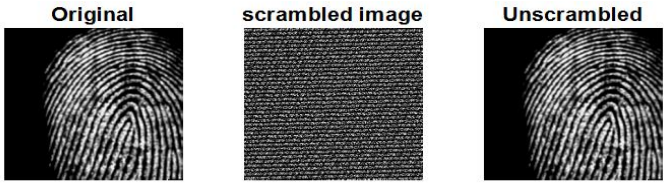
Pnum1 = Number of 1's in the plain image

fnum1 = Number of 1's in the key image

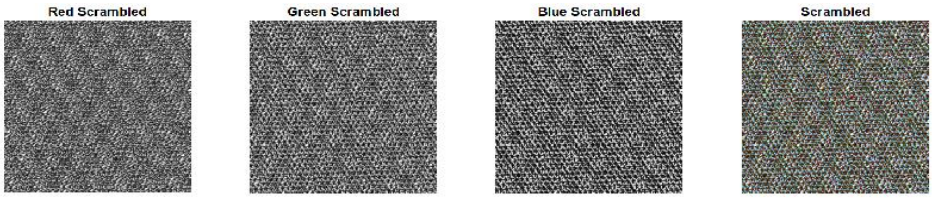
sumR = Sum of all pixels in R plane

sumG = Sum of all pixels in G plane

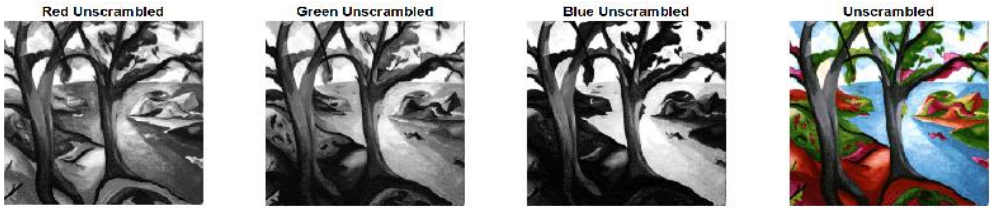
sumB = Sum of all pixels in B plane



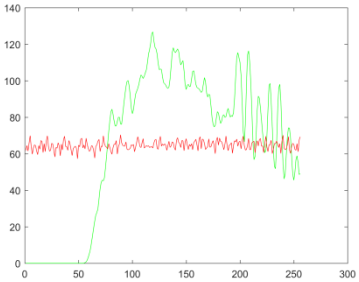
(a) Effect of scrambled and unscrambled process of bitmap image



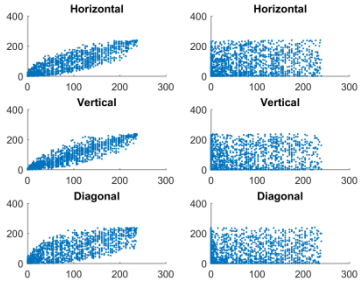
(b) Effect of scrambled process of color image



(c) Effect of unscrambled process of color image

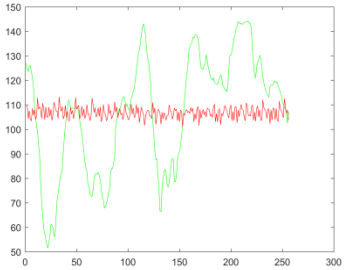


(d) Mean value analysis of bitmap image

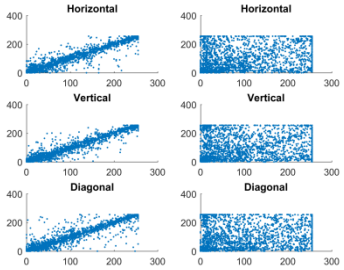


(e) Adjacent pixels correlation of bitmap image

Fig. 1 Arnold scrambling method analysis



(f) Mean value analysis of color image



(g) Adjacent pixels correlation of color image

Fig. 1 Arnold scrambling method analysis (continued)

Although the author proposed an encryption algorithm in his paper using Arnold map scrambling method, this paper just analyzed the performance of the shuffling algorithm used in the given paper. The below Fig. 1 represents the implementation results of color image "tree.png' and a bitmap image 'fingerprint.bmp' after applying Arnold scrambling method.

2.2. Affine 2D transformation

Affine transformation generates a geometrically distorted image by applying a linear combination of translation, scaling, rotation or shearing operations while preserving parallel lines and equi-spaced points among the lines. This paper has implemented the affine transformation method described by Amitava Nag *et al.* [44] to scramble the given image of size $M*N$. The Authors have used a 64bits symmetric key divided into 8 sub-keys K_0-K_7. The values of K_0-K_7 are chosen in such a way that the relation between sub-key and the height and width of the image will be

$$\gcd(K_0, M) = 1 \text{ and } \gcd(K_3, N)=1 \tag{3}$$

The above-given condition makes sure that no more than one location should map to the same destination during the transformation process. Here K_0-K_3 were used in location transformation of the pixel values, whereas rest sub-keys were used in the encryption process. Let suppose a pixel is represented by $\{x, y\}$ where $x \in \{0, 1, 2, \ldots, M\text{-}1\}$ and $y \in \{0, 1, 2, \ldots, N\text{-}1\}$ in an image and is supposed to transform into $\{x', y'\}$. The affine cipher function is as follows:

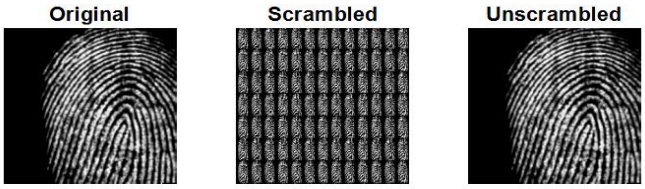
$$x' = (K_0 + K_1 * x) mod\ M \tag{4a}$$

$$y' = (K_2 + K_3 * y) mod\ N \tag{4b}$$

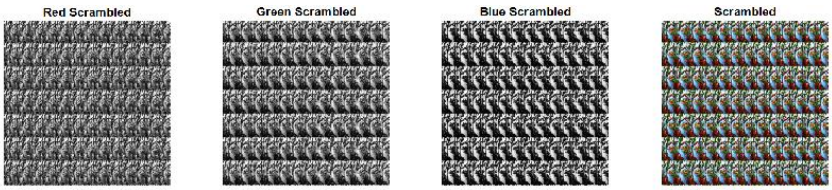
The corresponding inverse affine cipher function is:

$$x = ((x' + (-K_0)) + K_1^{-1} * P) mod\ N \tag{5a}$$

$$y = ((y' + (-K_2)) + K_3^{-1} * P) mod\ N \tag{5b}$$



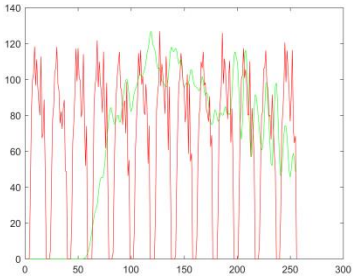
(a) Effect of scrambled and unscrambled process of bitmap image



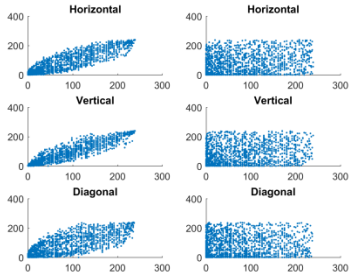
(b) Effect of scrambled process of color image

Fig. 2 Affine 2D transformation method analysis

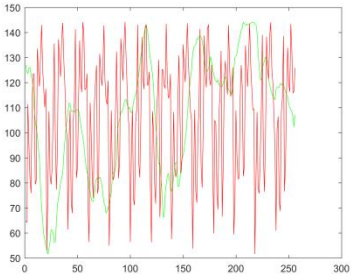
(c) Effect of unscrambled process of color image



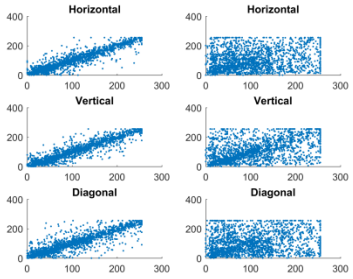
(d) Mean value analysis of bitmap image



(e) Adjacent pixels correlation of bitmap image



(f) Mean value analysis of color image



(g) Adjacent pixels correlation of color image

Fig. 2 Affine 2D transformation method analysis (continued)

The outcome images of the above-given affine transformation method can be seen in Fig. 2.

2.3. *Baker map*

Baker map 2D transformation is an extension of the 1D tent map with the invertible property. It is a chaotic bijection of the unit square matrix *N*N* onto itself based on the shuffling of pixel positions of an image without impacting the pixel values of an image. The transformation can be achieved in two ways: one is folded in which one slice is folded over or rotate before joining the other one and another one is unfolding in which upper section is unfolded [45].

The paper implemented the Baker transformation (see Fig. 3) described by the author in the following steps [46]:

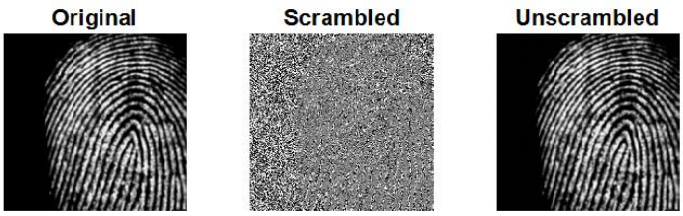
1. Initialize the values of x and y with the help of x_0, y_0 and r.
2. Repeat the given steps till a predetermined number of times

$$Z(X,Y)=\begin{cases}[x_{i+1},y_{i+1}]=[2x_i,\dfrac{y_i}{2}],\ if\ \ 0\le x_i<0.5\\ [x_{i+1},y_{i+1}]=[2(1-x_i),1-\dfrac{y_i}{2}],\ if\ \ 0.5\le x_i<1.0\end{cases} \tag{6}$$

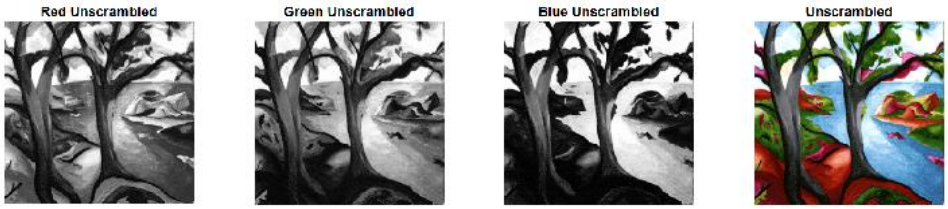
3. Extract the result into a variable and binarize it, using the given function:

$$P(x)=\begin{cases}0,\ if\ \ 0<Z(i,j)\le 0.5\\ 1,\ if\ \ Z(i,j)>0.5\end{cases} \tag{7}$$

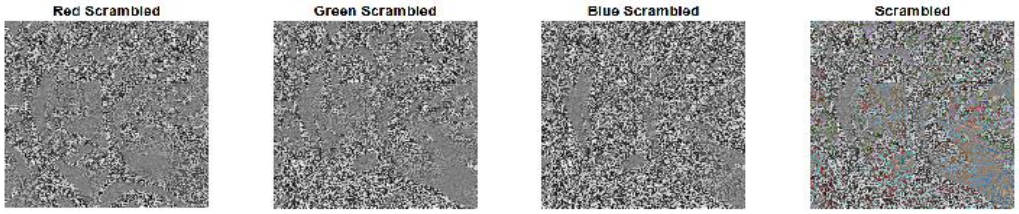
4. To scramble the given image, each instance of the P matrix is bitXORed with each respective bit of the image.
5. The corresponding unscrambled image can be obtained by applying all steps in reverse to an appropriate value of x0, y0 and r.



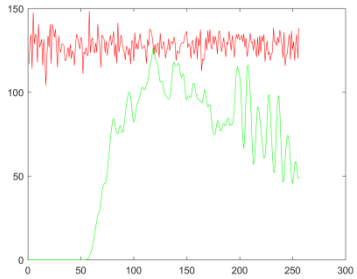
(a) Effect of scrambled and unscrambled process of bitmap image



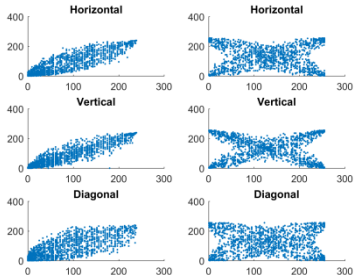
(b) Effect of scrambled process of color image



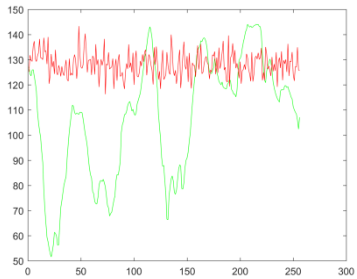
(c) Effect of unscrambled process of color image



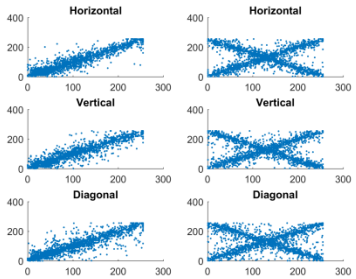
(d) Mean value analysis of bitmap image



(e) Adjacent pixels correlation of bitmap image



(f) Mean value analysis of color image



(g) Adjacent pixels correlation of color image

Fig. 3 Baker map scrambling method analysis

2.4. *Fibonacci sequence transformation*

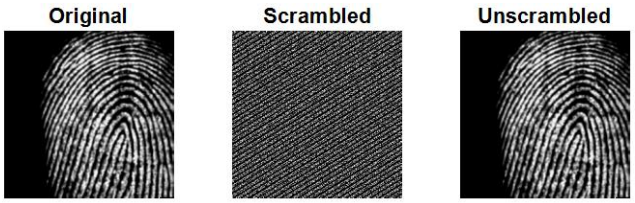
The Fibonacci sequence F_n is a recurrence sequence of integers as 1, 1, 2, 3, 5, 8, 13, 21, 34,…. given by an Italian Mathematician Leonardo of Pisa. The function used to derive the Fibonacci sequence is:

$$F_n = \begin{cases} 0, \ if \ \ n = 1 \\ 1, \ if \ \ n = 2 \\ F_{n-1} + F_{n-2}, \ Otherwise \end{cases} \tag{8}$$

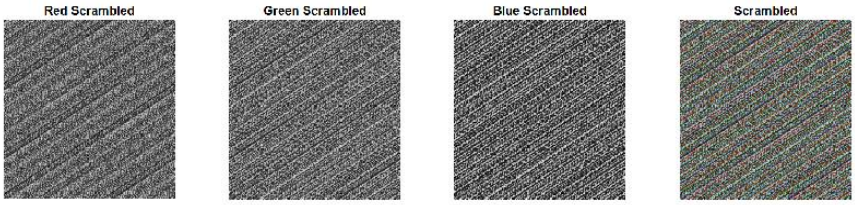
In 2010, the author used four consecutive terms of a Fibonacci unimodular matrix to replace 2D Arnold transformation matrix and used it as an image scrambler [47] because of its periodic nature. In general, a Fibonacci transforms system for a square matrix can be represented as:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} F_i & F_{i+1} \\ F_{i+2} & F_{i+3} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} mod\ N \tag{9}$$

Here $x, y \in \{0,1,2,3,\ldots, N\text{-}1\}$, x_1 and y_1 will be the transformed values, F_i denotes the i^{th} term of the Fibonacci sequence and N is the size of the matrix.

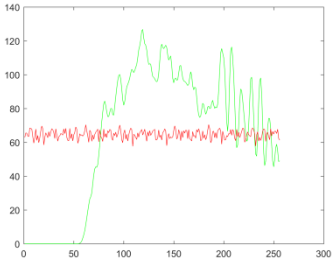


(a) Effect of scrambled and unscrambled process of bitmap image

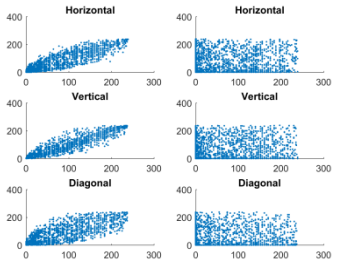


(b) Effect of scrambled process of color image

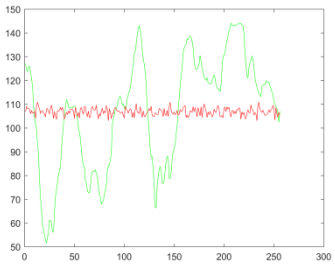
(c) Effect of unscrambled process of color image



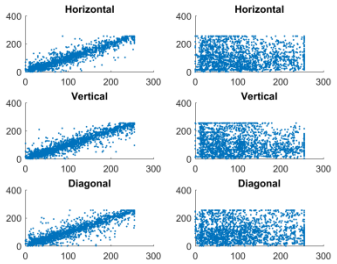
(d) Mean value analysis of bitmap image



(e) Adjacent pixels correlation of bitmap image



(f) Mean value analysis of color image



(g) Adjacent pixels correlation of color image

Fig. 4 Fibonacci sequence transformation analysis (continued)

The paper analyzed the performance of Fibonacci sequence scrambler for the four consecutive numbers i, i+1, i+2, i+3, where i=8 and depicted the results in Fig. 4.

2.5. *Fibonacci-Lucas sequence transformation*

The Fibonacci sequence can be combined with Lucas series to design a more innovative image scrambler [48]. The Lucas sequence is a special form of Fibonacci function and is given by a French mathematician François Édouard Anatole Lucas. To generate a Lucas series, the recurrence relation will be:

$$L_n = \begin{cases} 2, \ if \ n = 1 \\ 1, \ if \ n = 2 \\ L_{n-1} + L_{n-2}, \ Otherwise \end{cases} \quad (10)$$

Accordingly, the Lucas sequence consists of integers: 2, 1, 3, 4, 7, 11, 18, 29, 74,…

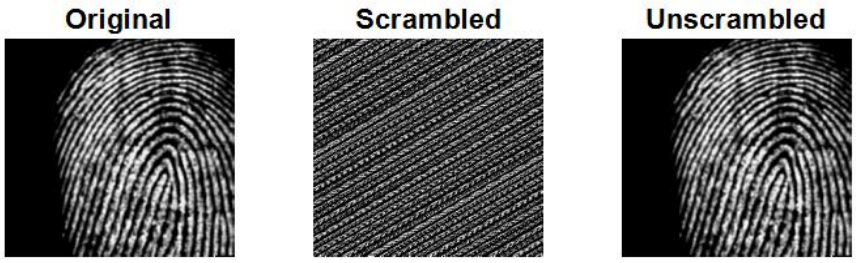
When we combine the Lucas series with the Fibonacci sequence, the resultant hybrid transformation will be as follows:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} F_i & F_{i+1} \\ L_i & L_{i+1} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} mod \ N \quad (11)$$

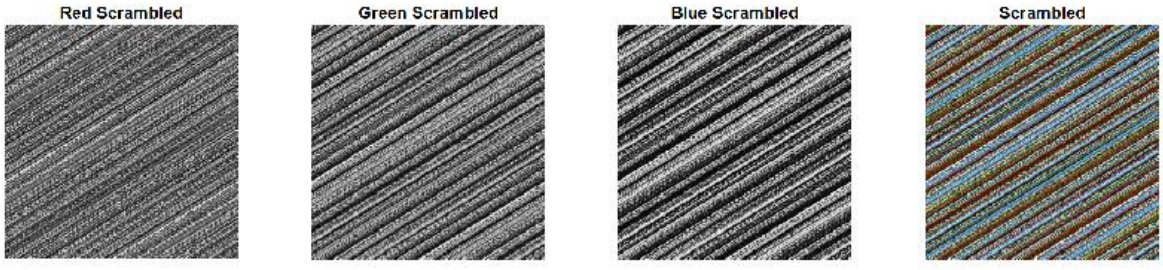
Here $x, y \in \{0,1,2,3,\ldots,N-1\}$, x_1 and y_1 will be the transformed values, F_i denote the i^{th} term of the Fibonacci sequence, L_i denote the i^{th} term of the Lucas series and N is the size of the matrix. We can start the transformation from any term in the given sequence (Fibonacci/Lucas) so it gives infinite transformation matrices to scramble the given image.

An image scrambler can't be designed solely with the help of the Lucas series because it does not form a unimodular periodic map to reverse the scrambling process. An important point to note that the transformation period for various sequences as well for the changed value of i^{th} term will be different for each one [48].

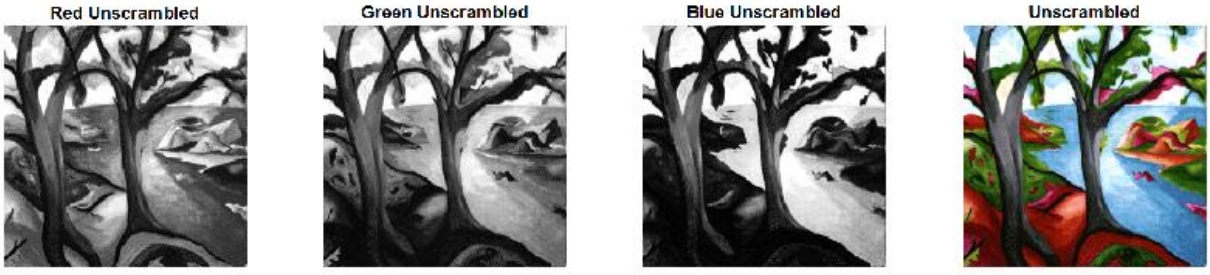
The below-given Fig. 5 represents the scrambling/unscrambling result of the combined transformation system of the Fibonacci sequence and Lucas series. Here the used Fibonacci sequence starts from 3 and 2 whereas Lucas series follows the standard format with the i=6th term of both the sequence. The Fibonacci sequence for the above-given value will be 3, 2, 5, 7, 12, 19, 31, ….



(a) Effect of scrambled and unscrambled process of bitmap image

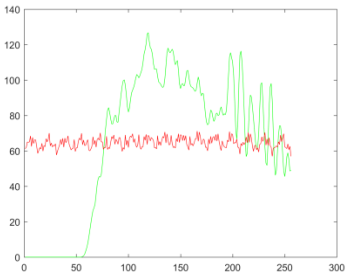


(b) Effect of scrambled process of color image

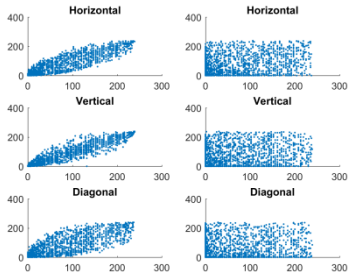


(c) Effect of unscrambled process of color image

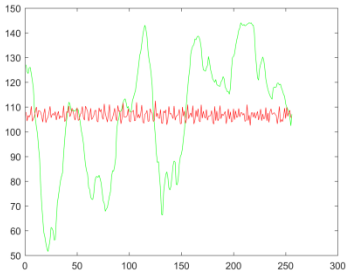
Fig. 5 Fibonacci-Lucas sequence transformation analysis



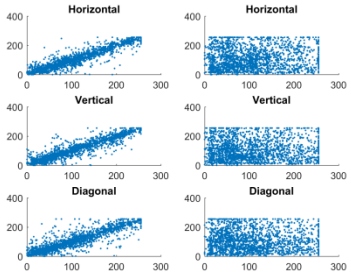
(d) Mean value analysis of bitmap image



(e) Adjacent pixels correlation of bitmap image



(f) Mean value analysis of color image



(g) Adjacent pixels correlation of color image

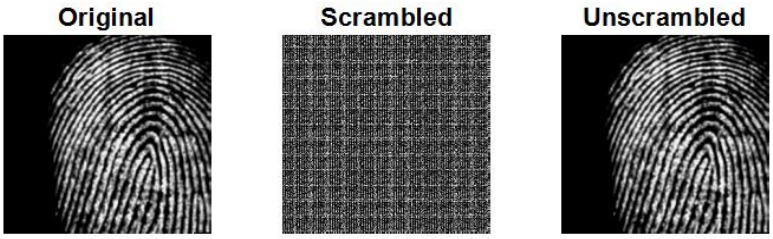
Fig. 5 Fibonacci-Lucas sequence transformation analysis (continued)

2.6. Fibonacci P-Code transformation

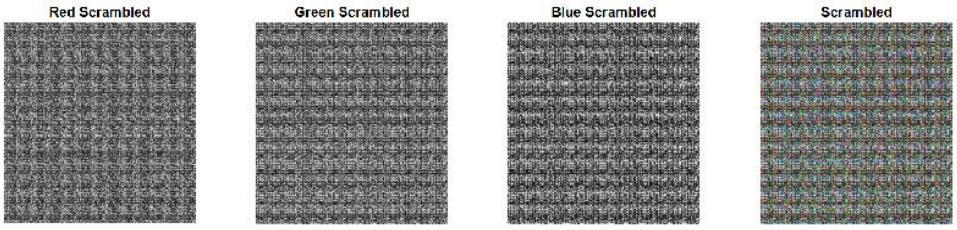
The next part of this section shows the resultant images of parametric Fibonacci sequence, i.e. Fibonacci p-code in spatial domain [49] in Fig. 6. The Fibonacci p-code sequence is defined as follows:

$$F_p(n) = \begin{cases} 0, \ if \ n < 1 \\ 1, \ if \ n = 1 \\ F(n-1) + F(n-p-1), \ n > 1 \end{cases} \tag{12}$$

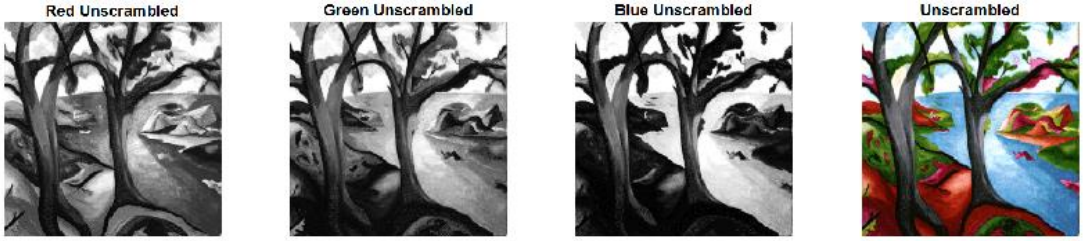
For p=1, The above recurrence relation would give the same output as the standard Fibonacci series.



(a) Effect of scrambled and unscrambled process of bitmap image

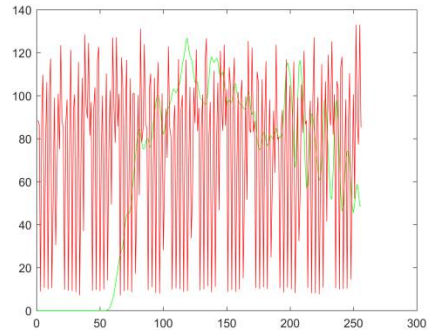


(b) Effect of scrambled process of color image

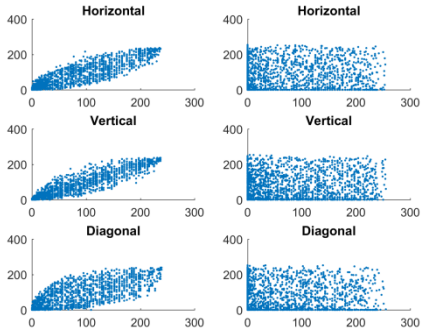


(c) Effect of unscrambled process of color image

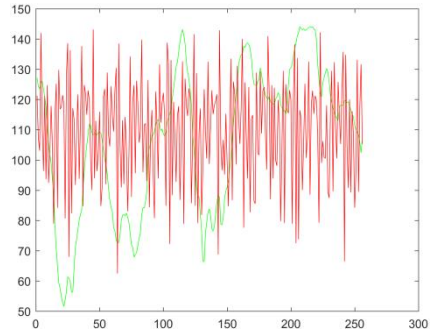
Fig. 6 Fibonacci p-code transformation analysis



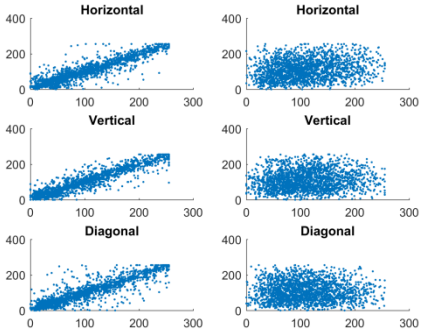
(d) Mean value analysis of bitmap image



(e) Adjacent pixels correlation of bitmap image



(f) Mean value analysis of color image



(g) Adjacent pixels correlation of color image

Fig. 6 Fibonacci p-code transformation analysis (continued)

The scrambled matrix of giving plain image I of size $M*N$ is calculated using the p-code Fibonacci transformation function:

$$S = T_r * I * T_c \tag{13}$$

where T_r and T_c are row coefficient matrix and column coefficient matrix respectively. The process to calculate the values of T_r and T_c can be stated as:

Let F_p (n) and F_p (n+1) are two consecutive Fibonacci p-code terms of the Fibonacci sequence. The permutation {$T1$, $T2$, ….,$T_{Fp(n+1)-1}$} of an input sequence {1, 2, 3, …, $F_p(n+1)$-1} is called 1D P-Fibonacci transform if {$T1$, $T2$, ….,$T_{Fp(n+1)-1}$} is defined by:

$$T_k = k[F_p(n) + i] mod\ F_p(n+1) \tag{14}$$

where k= 0, 1, ..., $F_p(n+1)$-1; i=-3, -2, -1, 0, 1, 2, 3; $F_p(n)+i<F_p(n+1)$

The row coefficient matrix $T_r(M*M)$ is calculated as:

$$T_r(i, j) = \begin{cases} 1, & (i, T_{pi}) \\ 0, & Otherwise \end{cases} \tag{15a}$$

Similarly, the column coefficient $T_c(N*N)$ is calculated as:

$$T_c(i, j) = \begin{cases} 1, & (T_{pi}, i) \\ 0, & Otherwise \end{cases} \tag{15b}$$

At the receiver end, receiver applied an inverse Fibonacci p-code transformation function to unscramble the cipher image. The equation is:

$$R = T_r^{-1} S T_c^{-1} \tag{16}$$

2.7. Henon map based transformation

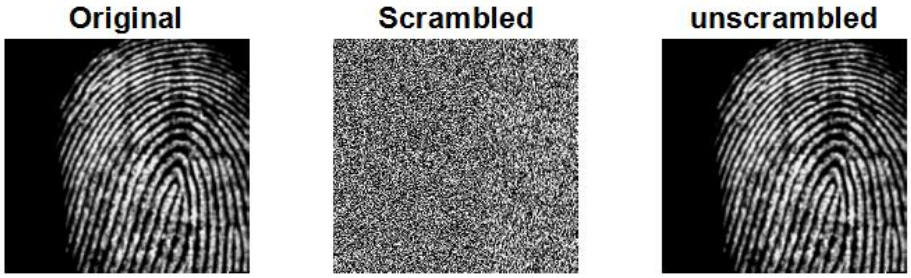
A Henon map is a two-dimensional chaotic system with quadratic non-linearity in which a point is mapped to a different place in the same plan. See:

$$x_{n+1} = 1 - a * x_n^2 + y_n \tag{17a}$$

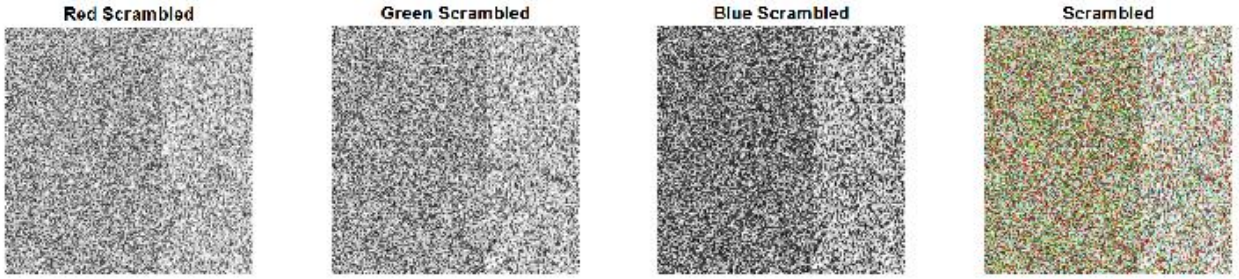
$$y_{n+1} = b * x_n \tag{17b}$$

The Henon map shows chaotic behavior with a boomerang-shaped chaotic attractor in a range of the variable *a i.e.* [1.07,1.4] for which it diverges to infinity. Otherwise, the map converges to a constant value or behaves periodic. The considered paper [25] has used a=1.76 and b=0.1 as initial values.

To scramble a plain image (see Fig. 7)

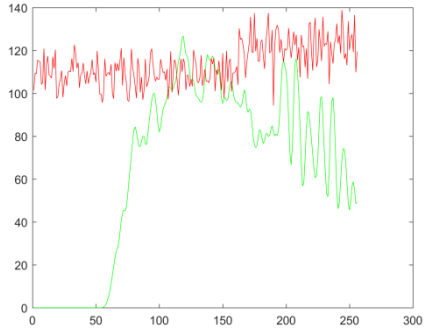


(a) Effect of scrambled and unscrambled process of bitmap image

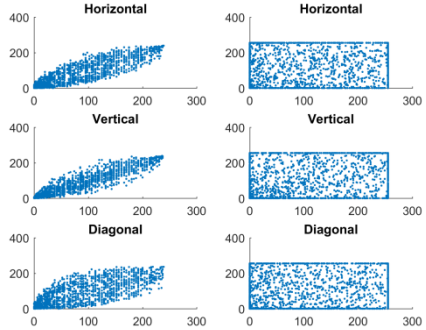


(b) Effect of scrambled process of color image

(c) Effect of unscrambled process of color image

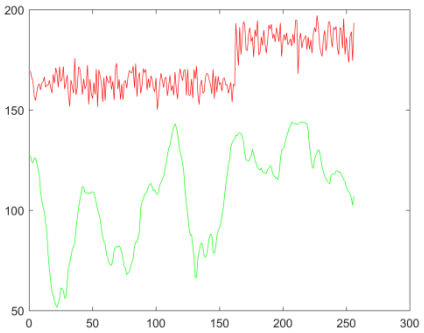


(d) Mean value analysis of bitmap image

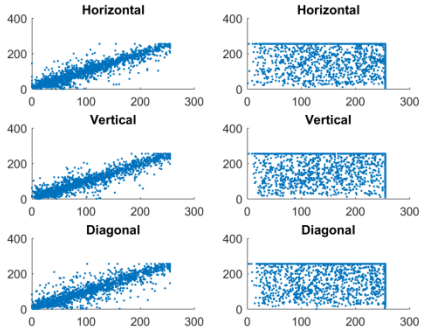


(e) Adjacent pixels correlation of bitmap image

Fig. 7 Henon map scrambling method analysis



(f) Mean value analysis of color image



(g) Adjacent pixels correlation of color image

Fig. 7 Henon map scrambling method analysis (continued)

- Calculate the updated values of x_{n+1} and y_{n+1} by iterating the Henon map function up to *the M*N number* of times (size of gray/color image).
- Sort the resultant values of the previous step and save its corresponding index in an array.
- Finally, to get the scrambled image, update the position of the intensity values of an original image according to the tabulated index values.
- At receiving end, regenerate the same random sequence using the Henon map to get back the sorted index.
- Apply reverse sorting to the elements to get back the original image pixel position.

2.8. Logistic map based transformation

The Logistic map is a 1D or 2D chaotic map and extensively researched because of its complex chaotic behavior. To analyze the scrambling behavior, a 1D Logistic map is being considered to scramble a plain image. The 1D Logistic equation could be defined as:

$$x_{n+1} = r * x_n (1 - x_n) \tag{18}$$

where x_{n+1} is an output sequence of the previous value x_n and r with range (0, 4]. A noted constraint of this method is its chaotic range, i.e. limited to [3.57, 4]. To enhance the range, the author used a combined function (Logistic and Tent map) as a seed map which is known as a Logistic-Tent system(LTS).

The equation for the Tent map can be expressed as:

$$x_{n+1} = \begin{cases} \dfrac{ux_n}{2}, & x_i < 0.5 \\ u(1 - x_n)/2, & x_i \geq 0.5 \end{cases} \tag{19}$$

Here the range of u is (0,4] but the chaotic range improves to [2, 4] which is better than the range of the Logistic chaotic map.

The combined seed map (Logistic and Tent map) function is used in the paper [50] to enhance the performance and security of the proposed cryptosystem. See the used equation below:

$$x_{n+1} = \begin{cases} (r * x_n (1 - x_n) + \dfrac{(4 - r)x_n}{2}) \\ (r * x_n (1 - x_n) + \dfrac{(4 - r)(1 - x_n)}{2}) \end{cases} \tag{20}$$

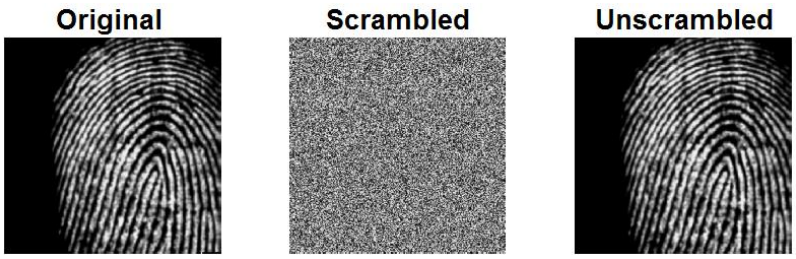
The proposed cryptosystem executed into four rounds, in which each round has five steps:

1. A single pixel insertion with random value at the beginning of each row of an original image.
2. Divide an original image into row by row 1D matrices.
3. Substitute data values in the 1D matrices using a given function.

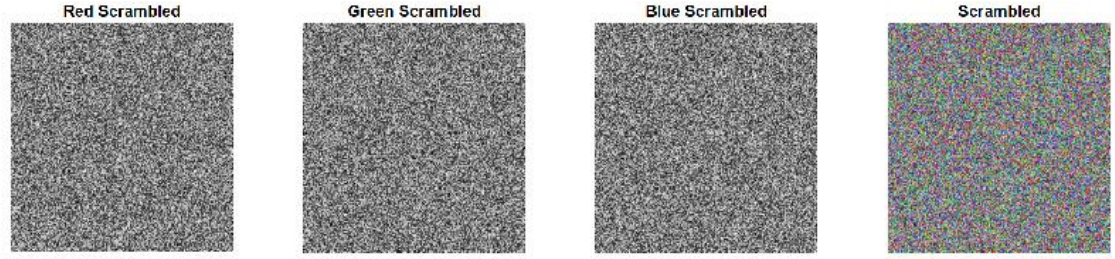
4. The next step is just the reverse of the previous step which combined all 1D matrices with the removal of the first pixel in each row.
5. Finally, rotate the 2D image matrix 90° counter-clockwise.

For the next round, obtained 2D image will be inputted as an initial image (like an original image) and repeat all above given steps for the next three rounds.

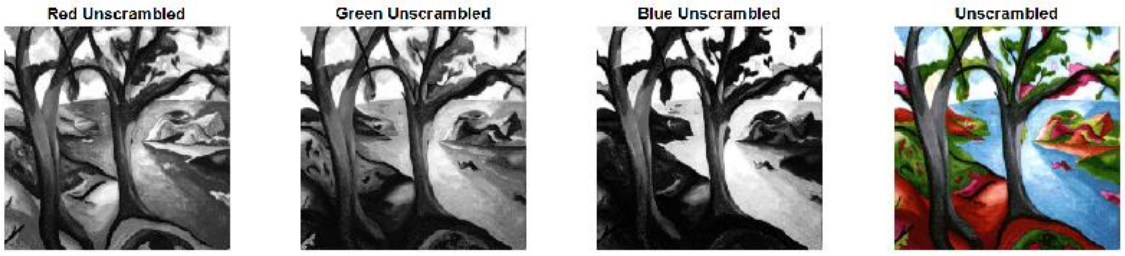
To decrypt, the receiver would need to apply an inverse of image encryption process with the right set of security keys. The results shown in Fig. 8 conclude a complex scrambling method.



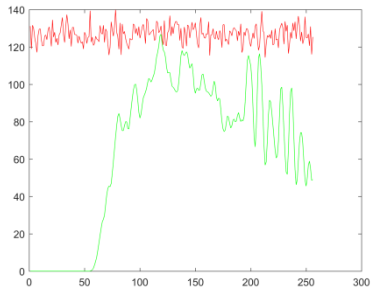
(a) Effect of scrambled and unscrambled process of bitmap image



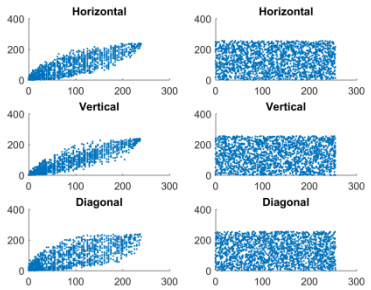
(b) Effect of scrambled process of color image



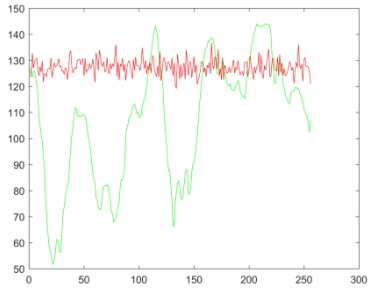
(c) Mean value analysis of bitmap image



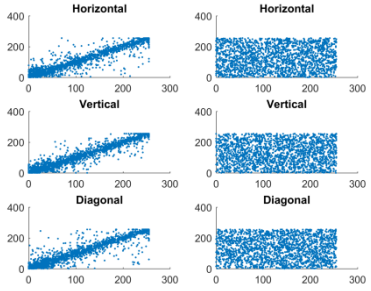
(d) Mean value analysis of bitmap image



(e) Adjacent pixels correlation of bitmap image



(f) Mean value analysis of color image



(g) Adjacent pixels correlation of color image

Fig. 8 Logistic map scrambling method analysis

3. Performance

Several tests have been carried out to measure the effectiveness and efficiency of the various scrambling methods described in the previous section. The statistical and differential analysis of the given chaotic maps is carried out for a color image "tree.png" and a bitmap image "fingerprint.bmp" each of size 256*256.

3.1. Adjacent pixel correlation analysis

The Correlation coefficient represents the relationship between two adjacent pixels in an image. The image pixels are very close to each other hence has a high correlation value between adjacent pixels. In contrary, a scrambled image should have a low correlation value between the adjacent pixels to make it difficult to identify the relationship between the image pixels by an unauthorized user.

The formula to calculate the correlation coefficient is as follows:

$$cc = \frac{cov(x, y)}{\sigma x * \sigma y} \tag{21}$$

where $\sigma x = \sqrt{var(x)}$ and $\sigma y = \sqrt{var(y)}$

$$var(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{22a}$$

$$cov(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{22b}$$

Here x and y are adjacent pixels in a plain image or scrambled image of size $M*N$. The paper shows the horizontal, vertical and diagonal correlation coefficient value of adjacent pixels of plain image and its corresponding scrambled image.

The distribution of adjacent pixel sequence (horizontal, vertical and diagonal) in the plain image and corresponding scrambled image has already been shown in the above-given figures (see Fig. 1-Fig. 8). However, the below Table 1 has the values of the correlation coefficient of all adjacent pixels to represent the relation between plain image pixels and in the corresponding scrambled image pixels.

3.2. Mean value analysis

Mean value analysis of image pixels deals with the horizontal and vertical distribution of the average intensity of pixel values of an image. In an original image, the mean intensity value of pixels distributed along the width of the image, whereas the same values of the scrambled image remain consistent along the width of the image. It represents the consistent distribution of scrambled image pixels along the vertical lines of an image. The relation between the pixel distribution of a plain image and its scrambled image has already been shown by the agreeing method explanation (see part (d) and (f) in Fig. 1-Fig. 8).

3.3. NPCR and UACI Tests

NPCR (Number of pixel change rate) and UACI (Unified average changing intensity) are standardized tests to analysis a plain image sensitivity [51]. NPCR value is used to test the influence of the number of pixels change between plain image and scrambled image. Let's consider a plain image "IM" and its corresponding scrambled image "SCR" of size $M*N$, then the value of a bipolar array D to the same size as "IM" or "SCR" will be calculated as:

$$D(i, j) = \begin{cases} 0, & if\ IM(i, j) = SCR(i, j) \\ 1, & if\ IM(i, j) \neq SCR(i, j) \end{cases} \tag{23}$$

The NPCR is defined as:

$$NPCR = \frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i, j) * 100\% \tag{24}$$

The greater the value of NPCR results better plain image sensitivity. Ideally, the NPCR value of a true scrambled image should be around 99.

The UACI is defined as:

$$UACI = \frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{IM(i, j) - SCR(i, j)}{255} * 100\% \tag{25}$$

UACI test calculates the average intensity change between plain image and scrambled image. Table 1 shows the NPCR and UACI values for color image and bitmap image respectively.

3.4. Peak signal noise ratio (PSNR)

PSNR used to measure the image encryption quality. It helps to calculate the change in the pixel value of a plain image P and in the cipher image C of size $M*N$.

$$PSNR = 10 * log_{10} \left[\frac{M * N * 255^2}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (P(i, j) - c(i, j))^2} \right] \tag{26}$$

The lower value of PSNR reflects the better encryption quality. The corresponding PSNR values for a color image as well as for bitmap image can be seen in Table 1.

3.5. Entropy analysis

The information entropy was given in 1949 by Shannon and is a statistical measure to estimate the randomness and unpredictability of an information source [52]. The message entropy $H(s)$ of message source s is defined as:

$$H(s) = -\sum_{i=0}^{2^N - 1} P(s_i) log_2 P(s_i) \tag{27}$$

Here $P(s_i)$ is the probability of symbol s_i and N is the number of bits to represent a symbol s_i. The entropy value is N if a random information source consists of 2^N symbols. In Table 1, the entropy values have given for both types of images. Ideally, the entropy value of a scrambled image with 256 gray levels should be 8.

Table 1 The scrambling performance parameters measure for both RGB images as well as bitmap image with resolution 256*256

		RGB Image				Bitmap Image			
Affine Map	Correlation Coefficient		Hori.	Vert.	Diag.		Hori.	Vert.	Diag.
		Orig. Image	0.9430	0.9457	0.9180	Orig. Image	0.9145	0.9455	0.8304
		Scram. Image	0.2750	0.5663	0.2028	Scram. Image	0.3738	0.4758	0.3184
	Diff. Attack Measure		NPCR%	UACI %			NPCR%	UACI %	
		Scram. Image	99.2905	31.4854		Scram. Image	90.0558	29.9275	
	PSNR	Scram. Image	6.8528			Scram. Image	8.1577		
	Entropy	Scram. Image	7.6140			Scram. Image	5.1963		

Table 1 The scrambling performance parameters measure for both RGB images as well as bitmap image with resolution 256*256 (continued)

		RGB Image				Bitmap Image			
Logistic Map	Correlation Coefficient		Hori.	Vert.	Diag.		Hori.	Vert.	Diag.
		Orig. Image	0.9430	0.9457	0.9180	Orig. Image	0.9145	0.9455	0.8304
		Scram. Image	0.0008	-0.0011	-0.0017	Scram. Image	0.0035	-0.0018	0.0005
	Diff. Attack Measure		NPCR%	UACI %			NPCR%	UACI %	
		Scram. Image	99.6338	33.6859		Scram. Image	99.6399	38.8606	
	PSNR	Scram. Image	7.0642			Scram. Image	6.5757		
	Entropy	Scram. Image	7.9991			Scram. Image	7.9970		
Arnold Map	Correlation Coefficient		Hori.	Vert.	Diag.		Hori.	Vert.	Diag.
		Orig. Image	0.9430	0.9457	0.9180	Orig. Image	0.9145	0.9455	0.8304
		Scram. Image	0.1139	0.0295	-0.0722	Scram. Image	0.3876	0.1886	0.1687
	Diff. Attack Measure		NPCR%	UACI %			NPCR%	UACI %	
		Scram. Image	99.3759	31.6361		Scram. Image	90.3717	29.8647	
	PSNR	Scram. Image	6.7550			Scram. Image	8.1939		
	Entropy	Scram. Image	7.6140			Scram. Image	5.1963		
Baker Map	Correlation Coefficient		Hori.	Vert.	Diag.		Hori.	Vert.	Diag.
		Orig. Image	0.9430	0.9457	0.9180	Orig. Image	0.9145	0.9455	0.8304
		Scram. Image	-0.0045	0.0017	-0.0044	Scram. Image	0.0091	0.0069	0.0007
	Diff. Attack Measure		NPCR%	UACI %			NPCR%	UACI %	
		Scram. Image	87.0468	25.4157		Scram. Image	91.1316	33.2744	
	PSNR	Scram. Image	6.4775			Scram. Image	5.6321		
	Entropy	Scram. Image	7.7926			Scram. Image	6.2967		
Henon Map	Correlation Coefficient		Hori.	Vert.	Diag.		Hori.	Vert.	Diag.
		Orig. Image	0.9430	0.9457	0.9180	Orig. Image	0.9145	0.9455	0.8304
		Scram. Image	0.0163	0.1245	0.0182	Scram. Image	0.0028	0.1293	0.0008
	Diff. Attack Measure		NPCR%	UACI %			NPCR%	UACI %	
		Scram. Image	98.0850	41.0883		Scram. Image	91.4764	40.0931	
	PSNR	Scram. Image	5.5379			Scram. Image	5.8233		
	Entropy	Scram. Image	6.0184			Scram. Image	6.0366		
Fibonacci Sequence	Correlation Coefficient		Hori.	Vert.	Diag.		Hori.	Vert.	Diag.
		Orig. Image	0.9430	0.9457	0.9180	Orig. Image	0.9145	0.9455	0.8304
		Scram. Image	-0.0527	-0.0740	-0.0211	Scram. Image	0.2373	0.1419	-0.0060

Table 1 The scrambling performance parameters measure for both RGB images as well as bitmap image with resolution 256*256 (continued)

		RGB Image				Bitmap Image			
Fibonacci Sequence	Diff. Attack Measure		NPCR%	UACI %			NPCR%	UACI %	
		Scram. Image	99.2935	31.4864		Scram. Image	90.3061	29.9601	
	PSNR	Scram. Image	6.7775			Scram. Image	8.1755		
	Entropy	Scram. Image	7.6140			Scram. Image	5.1963		
Fibonacci -Lucas Series	Correlation Coefficient		Hori.	Vert.	Diag.		Hori.	Vert.	Diag.
		Orig. Image	0.9430	0.9457	0.9180	Orig. Image	0.9145	0.9455	0.8304
		Scram. Image	0.0645	-0.0071	-0.0701	Scram. Image	0.4038	0.2345	0.0630
	Diff. Attack Measure		NPCR%	UACI %			NPCR%	UACI %	
		Scram. Image	99.2889	31.6799		Scram. Image	90.3290	30.0312	
	PSNR	Scram. Image	6.7383			Scram. Image	8.1539		
	Entropy	Scram. Image	7.6140			Scram. Image	5.1963		
Fibonacci -P Cose Sequence	Correlation Coefficient		Hori.	Vert.	Diag.		Hori.	Vert.	Diag.
		Orig. Image	0.9430	0.9457	0.9180	Orig. Image	0.9145	0.9455	0.8304
		Scram. Image	0.2217	0.0934	-0.0199	Scram. Image	0.0577	0.3840	-0.0589
	Diff. Attack Measure		NPCR%	UACI %			NPCR%	UACI %	
		Scram. Image	99.6216	28.4619		Scram. Image	98.0209	28.5384	
	PSNR	Scram. Image	7.8529			Scram. Image	8.9636		
	Entropy	Scram. Image	7.7025			Scram. Image	7.4838		

3.6. *Encryption/decryption speed analysis*

In this paper, all analyzed scrambling methods have been implemented in MATLAB (2016) using command tic and toc. The encryption/decryption speed mentioned here is carried out on Intel® Atom™ x7-z8700 CPU @1.60GHz with 4 GB RAM. The average scrambling/unscrambling time is measured for 256*256 sized color image as well as for bitmap image of the same size. The below figures show the execution speed result of all scrambling methods discussed in the previous section.

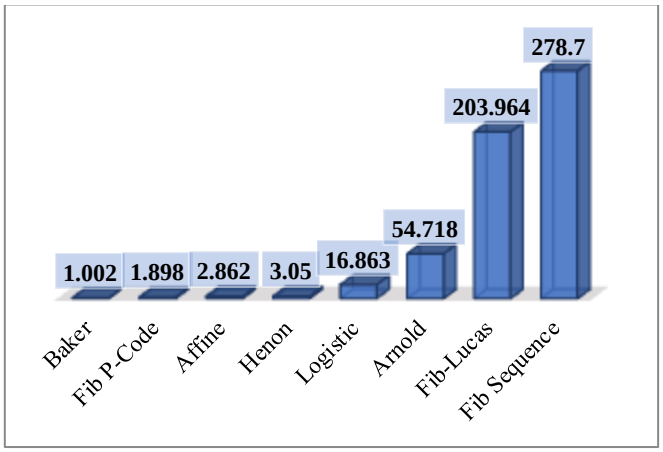


Fig. 9 Scrambling time for color image "tree.png"

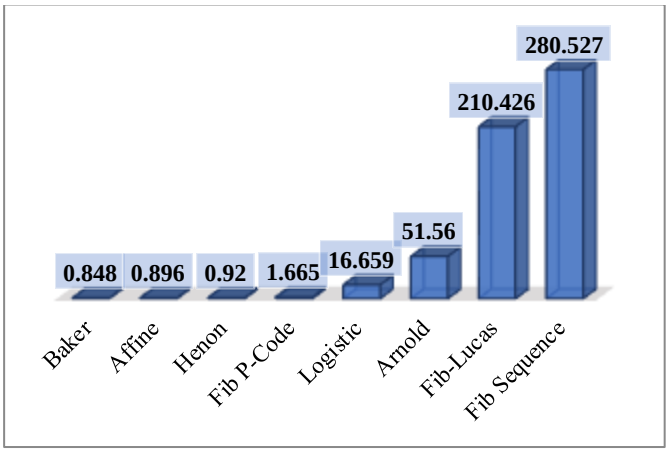


Fig. 10 Unscrambling time for color image "tree.png"

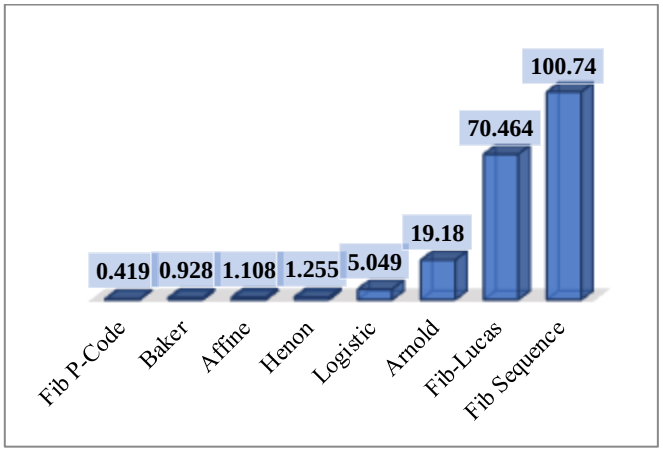


Fig. 11 Scrambling time for bitmap image "finger.bmp"

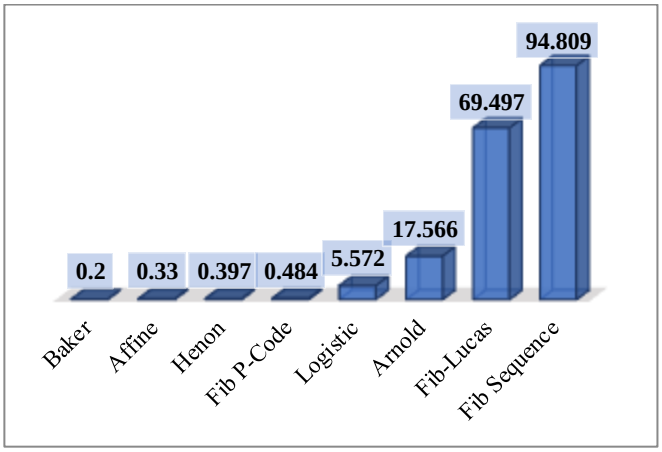
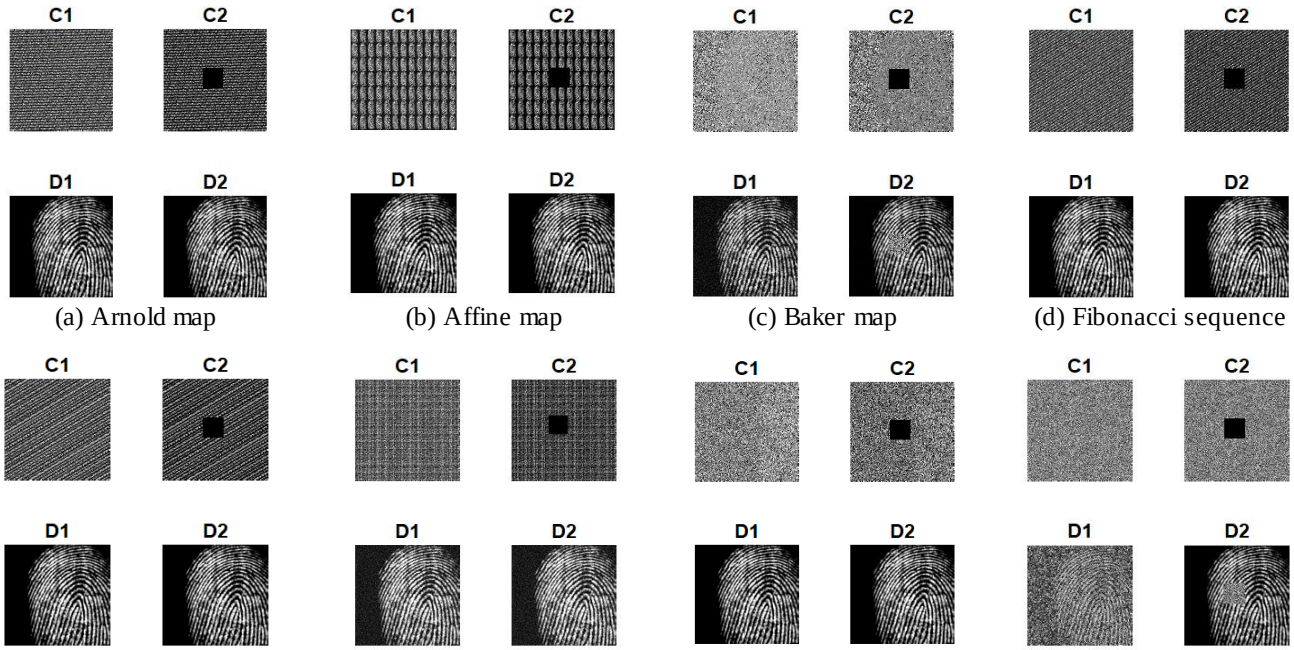


Fig. 12 Unscrambling time for bitmap image "finger.bmp"

3.7. *Impact of data loss and noise*

The images are most likely prone to the data loss or interrupted by noise induced during transmission through the network. An image scrambling method should be robust enough to resist the effect of data loss or noise. A chaotic map based scrambling method is sensitive to the change in the pixel value. Therefore, making a small change in original image causes to a huge difference in the corresponding cipher image. However, in the decryption process, the change in one pixel affects few pixels in the recovered image. Thus, an efficient scrambling method must be able to reproduce the cipher image with data loss or noise. The Fig. 13 shows the recovered original images from the cipher image with noise or data loss. The resulting images depict that few scrambling methods are strong enough to recover an image completely while some of them are not. Still, the image information can be recognized.



(a) Arnold map (b) Affine map (c) Baker map (d) Fibonacci sequence

(e) Fibonacci-Lucas sequence (f) Fibonacci P-code (g) Henon map (h) Logistic map

Fig. 13 Result of impact of data loss or noise on "fingerprint.bmp" image in the form of cipher image and its corresponding decrypted image using various scrambling method

3.8. *Summary*

The paper is focused on analyzing the performance and security measure of 8 chaotic maps which are used to scramble the image. The quality of a technique depends on several factors like execution time, accuracy and memory cost. The values in Table 1 show the correlation coefficients are very small between the image pixels except for the affine transformation.

Hence the pixels of a color image as well as of bitmap image are almost unrelated in all directions. The part (e) and (g) of Figs. (1)-(8) reflect the adjacent pixels sequence pair for all said methods. The observation from those figures illustrates the uniform distribution of image pixels after scrambling. Only the Baker map behaves differently by showing scrambled image pixel distribution diagonally.

The execution time of the above-studied cases would depend on the system configuration. Processing speed is a necessary but not sufficient factor for measuring the quality and complexity of the algorithm. The scrambling time for Affine, Baker, Henon and Fibonacci p-code have best execution time as compared to others. Although the Logistic map has high processing time, its complexity and security measures are significant due to the other parameters listed in the study. The Fibonacci sequence and Fibonacci-Lucas series are requiring a very high execution time, so it can be utilized in multimedia applications for authenticated and accurate transmission.

Industry and communication system require highly reliable and unpredictable cryptosystem for secure transmission of information. The differential attack measure can be estimated with the help of NPCR and UACI values listed in Table 1. The outcome displayed in the table verifies that the all studied chaotic maps have standard NPCR and UACI values except Baker map. Therefore, these scrambling techniques can be used in a secure and efficient image encryption algorithm.

The sensitivity analysis is needed for a secure scrambled image and could be measured using entropy value. Table 1 also depicts the obtained entropy values of each studied scrambling method, which are close to 8 except the Henon map. The quality of the scrambling method can also be quantified using mean value analysis of image pixels. The part (d) and (f) of Figs. (1)-(8) reflect the mean value analysis of the plain image (fluctuated in both horizontal & vertical directions) and scrambled image (fluctuated parallel to x axis) of the corresponding scrambling techniques. The uniform distributions of scrambled image pixels along the width of an image are nearly consistent and close to each other. Thus, the secrecy, accuracy, and quality of an algorithm can be acknowledged using the various discussed performance parameters in the paper.

4. Conclusion

There are many image encryption research papers being published using a chaotic map based scrambling method. The objective of this review paper is not to suggest a good scrambling method, but to provide the insights of explored chaotic maps already used in the modified form in an image encryption algorithm. The paper covered eight scrambling methods which were already applied in existing image encryption algorithms by various authors. Based on the study of the above-mentioned papers, it is worth noting that the chaotic system plays an important role to provide a faster and a secure cryptosystem as compared to a conventional system. Several characteristics of chaos such as unpredictability, randomness, sensitive to its initial condition, large key space and low memory capacity make the scrambling methods most suited to design an effective, efficient, fast, and secure cryptosystem. The chaos-based scrambling method provides a better trade-off between security and computational complexity, hence identified as an integral part in the design of a reliable and an authenticated cryptosystem.

References

[1] M. Khan and T. Shah, "A literature review on image encryption techniques," 3D Research, vol. 5, no. 4, p. 29, December 2014.

[2] F. S. Abed, "A new approach to encoding and hiding information in an image," International Journal of Computer Science Issues, vol. 8, no. 5, pp. 514-523, September 2011.

[3] J. Fridrich, Method for encrypting and decrypting data using chaotic map, U.S. Patent, 6,064,738, May 16, 2000.

[4] R. C. Hilborn, Chaos and nonlinear dynamics: an introduction to scientists and engineers, 2nd ed. New York: Oxford University Press, 2001.

[5] X. Wu, H. Hu, and B. Zhang, "Parameter estimation only from symbolic sequence generated by chaos system," Chaos Soliton & Fractals, vol. 22, no. 2, pp. 359-366, October 2004.

[6] R. Rhouma and S. Belghith, "Cryptanalysis of a spatiotemporal chaotic cryptosystem," Chaos Solitons & Fractals, vol. 41, no. 4, pp. 1718-1722, August 2009.

[7] L. Shujun and X. Zheng, "Cryptanalysis of a chaotic image encryption method," IEEE International Symp. Circuits and Systems (ISCAS 2002), vol. 2, pp. 708-711, 2002.

[8] R. Parvaz and M. Zarebnia, "A combination chaotic system and application in color image encryption," Optics & Laser Technology, vol. 101, pp. 30-41, May 2018.

[9] Z. Hua, Y. Zhou, C. M. Pun, and C. L. P. Philip, "2D sine logistic modulation map for image encryption," Information Science, vol. 297, pp. 80-94, March 2015.

[10] Z. Hua and Y. Zhou, "Image encryption using 2D Logistic-adjusted-Sine map," Information Science, vol. 339, pp. 237-253, April 2016.

[11] J. M. Vilardy, C. J. Jimenez, and R. Perez, "Image encryption using the Gyrator transform and random phase masks generated by using chaos," Journal of Physics: Conference Series, vol. 850, pp. 012012-1-012012-7, 2017.

[12] X. Y. Wang, S. Wang, Y. Zhang, and K. Guo, "A novel image encryption algorithm based on chaotic shuffling method," Information Security Journal: A Global Perspective, vol. 26, no. 1, pp. 7-16, 2017.

[13] M. Kanafchian and B. Fathi-Vajargah, "A novel image encryption scheme based on Clifford attractor and noisy logistic map for secure transferring images in navy," International Journal of e-Navigation and Maritime Economy, vol. 6, pp. 53-63, April 2017.

[14] R. Zahmoul, R. Ejbali, and M. Zaied, "Image encryption based on new beta chaotic map," Optics and Lasers in Engineering, vol. 96, pp. 39-49, September 2017.

[15] M. Mikhail, Y. Abouelseoud, and G. ElKobrosy, "Two-phase image encryption scheme based on FFCT and fractals," Security and Communication Networks, vol. 2017, pp. 7367518-1-7367518-13, 2017.

[16] H. Oğraş and M. Türk, "A robust chaos-based image cryptosystem with an improved key generator and plain image sensitivity mechanism," Journal of Information Security, vol. 8, no. 1, pp. 23-41, January 2017.

[17] Y. Sun, L. Chen, R. Xu, and R. Kong, "An image encryption algorithm utilizing julia sets and hilbert curves," PLoS ONE, vol. 9, no. 1, January 2014.

[18] Q. Zhang, S. Zhou, and X. Wei, "An efficient approach for DNA fractal-based image encryption," Applied Mathematics & Information Sciences, vol. 5, no. 3, pp. 445-459, 2011.

[19] G. A. Sathishkumar, K. B. Bagan, and N. Sriraam, "Image encryption based on diffusion and multiple chaotic maps," International Journal of Network Security and its Applications, vol. 3 no. 2, pp. 181-194, March 2011.

[20] Y. Xu, H. Wang, Y. Li, and B. Pei, "Image encryption based on synchronization of fractional chaotic systems," Communications in Nonlinear Science and Numerical Simulation, vol. 19, no. 10, pp. 3735-3744, October 2014.

[21] M. Ahmad, U. Shamsi, and I. R. Khan, "An enhanced image encryption algorithm using fractional chaotic systems," Procedia Computer Science, vol. 57, pp. 852-859, 2015.

[22] M. Kumar, P. Powduri, and A. Reddy, "An RGB image encryption using diffusion process associated with chaotic map," Journal of Information Security and Applications, Elsevier, vol. 21, pp. 20-30, April 2015.

[23] N. K. Pareek, V. Patidar, and K. K Sud, "Substitution-diffusion based image cipher," International Journal of Network Security & Its Applications, vol. 3, no. 2, pp. 149-160, March 2011.

[24] J. X. Chen, Z. L. Zhu, C. Fu, H. Yu, and L. B. Zhang, "A fast chaos-based image encryption scheme with a dynamic state variable selection mechanism," Communications in Nonlinear Science and Numerical Simulation, vol. 20, no. 3, pp. 846-860, March 2014.

[25] M. Prasad and K. L. Sudha, "Chaos image encryption using pixel shuffling," Computer Science & Information Technology, pp. 169-179, 2011.

[26] N. K. Pareek, "Design and analysis of novel digital image encryption scheme," International Journal of Network Security & Its Applications, vol. 4, no. 2, pp. 95-108, March 2012.

[27] H. J. Liu and X. Y. Wang, "Colour image encryption based on one-time keys and robust chaotic maps," Computers & Mathematics with Applications, vol. 59, no. 10, pp. 3320-3327, May 2010.

[28] S. Sathyanarayana, M. Kumar, and K. Bhat, "Symmetric key image encryption scheme with key sequences derived from random sequence of cyclic elliptic curve points," International Journal of Network Security, vol. 12, no. 3, pp. 137-150, 2011.

[29] R. Guesmi, M. A. B. Farah, A. Kachouri, and M. Samet, "Hash key - based image encryption using crossover operator and chaos," Multimedia Tools and Applications, vol. 75, no. 8, pp. 4753-4769, April 2016.

[30] D. I. G. Amalarethinam and J. S. Geetha, "Image encryption and decryption in public key cryptography based on MR," International Conf. Computing and Communications Technologies, IEEE Press, February 2015, pp. 133-138.

[31] Z. Hua, Y. Wang, and Y. Zhou, "Image cipher using a new interactive two-dimensional chaotic map," IEEE International Conf. Systems, Man, and Cybernetics, IEEE Press, October 2015, pp. 1804-1808.

[32] Z. Hua, B. Zhou and Y. Zhou, "Image content-based encryption algorithm using high-dimensional chaotic system," International Symp. Nonlinear Theory and its Applications, December 2015, pp. 554-557.

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

[34] L. Liu and S. Miao, "A new image encryption algorithm based on logistic chaotic map with varying parameter," Springerplus, vol. 5, no. 289, pp. 1-12, March 2016.

[35] Y. Wu, Y. Zhou, S. Agaian, and J. P. Noonan, "2D Sudoku associated bijections for image scrambling," Information Sciences, vol. 327, pp. 91-109, January 2016.

[36] Z. Hua and Y. Zhou, "Design of image cipher using block-based scrambling and image filtering," Information Sciences, vol. 396, pp. 97-113, August 2017.

[37] N. Dwivedi, R. K. Gupta, and S. Agarwal, "Image encryption using curved scrambling and diffusion," International Journal of Engineering and Technology, vol. 8, no. 6, pp. 2990-2996, January 2017,

[38] S. Somaraj and M. A. Hussain, "Image encryption using edge map and key image", Indian Journal of Science and Technology, vol. 10, no. 4, pp. 1-4, January 2017.

[39] J. S. Teh and A. Samsudin, "A chaos-based authenticated cipher with associated data," Security and Communication Networks, vol. 2017, pp. 9040518-1-9040518-15, 2017.

[40] V. I. Arnold, "First steps in symplectic topology," Russian Math. Surveys, vol. 41, no. 6, pp. 1-21, 1986.

[41] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," Chaos Solitons and Fractals, vol. 21, no. 3, pp. 749-761, July 2004.

[42] Z. Tang, and X. Zhang, "Secure image encryption without size limitation using Arnold transform and random strategies," Journal of Multimedia, vol. 6, no. 2, pp. 202-206, 2011.

[43] A. Chopra, M. Ahmad, and M. Malik, "An enhanced modulo-based image encryption using chaotic and fractal keys," International Conf. Advances in Computer Engineering and Applications, July 2015, pp. 501-506.

[44] A. Nag, J. P. Singh, S. Khan, S. Biswas, D. Sarkar, and P. P. Sarkar, "Image encryption using affine transform and XOR operation," Proc. Signal Processing, Communication, Computing and Networking Technologies, IEEE Press, September 2011, pp. 309-312.

[45] Y. Dong, J. Liu, C. Zhu, and Y. Wang, "Image encryption algorithm based on chaotic mapping," Proc. Computer Science and Information Technology, IEEE Press, January 2011, pp 289-291.

[46] S. Kumar, B. Sinha, and C. Pradhan, "Comparative analysis of color image encryption using 2D chaotic maps," Information Systems Design and Intelligent Applications, vol. 2, pp. 79-387, 2015.

[47] D. Qi, J. Zou, and X. Han, "A new class of transform and its application in the image transform covering," Science in China Series E: Technological Sciences, vol. 43, no. 3, pp. 304-312, June 2000.

[48] M. Mishra, P. Mishra, M. C. Adhikary, and S. Kumar, "Image encryption using Fibonacci-Lucas transformation," International Journal on Cryptography and Information Security, vol. 2, no. 3, pp. 131-141, September 2012.

[49] Y. Zhou, S. Agaian, V. M. Joyner, and K. Panetta, "Two Fibonacci p-code based image scrambling algorithms," Proc. SPIE 6812 Image Processing: Algorithms and Systems VI, vol. 6812, March 2008, p. 681215.

[50] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," Signal Processing, vol. 97, pp. 172-182, April 2014.

[51] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," Journal of Selected Areas in Telecommunication, vol. 2, pp. 31-38, April 2011.

[52] C. E. Shannon, "Communication theory of secrecy systems," The Bell System Technical Journal, vol. 28, no. 4, pp. 656-715, October 1949.