Journal of *Imaging*

MDPI

Article

Secure Image Transmission Using Fractal and 2D-Chaotic Map

Shafali Agarwal

20829 Anza Avenue, Torrance, CA 90503, USA; shafali.agarwal@gmail.com; Tel.: +1-91-6693-4645

Received: 14 November 2017; Accepted: 5 January 2018; Published: 10 January 2018

Abstract: A chaos-based cryptosystem has been suggested and investigated since last decade because of its sensitivity to the initial condition, unpredictability and ergodicity properties. The paper introduces a new chaotic map which helps to enhance the security of image transmission by blending the superior fractal function with a new 2D-Sine Tent composite map (2D-STCM) to generate a key stream. A trajectory map of a proposed 2D-STCM show a wider chaotic range implies better unpredictability and ergodicity feature, suitable to design a cryptosystem. A fractal based image encryption increases the key space of the security key up-to hundreds of bits, thus secure the proposed cryptosystem from brute-force attack. The requirement of confusion and diffusion are fulfilled by applying chaotic circular pixel shuffling (CCPS) to change the pixel position repeatedly and the execution of an improved XOR operation i.e., complex XOR, designed to increase the encryption quality. The proposed cryptosystem has been analyzed using statistical analysis, key sensitivity, differential analysis and key space analysis. The experimental result proves that the new scheme has a high security level to protect the image transmission over the network.

Keywords: superior fractal function; 2D-chaotic map; pixel shuffling; pixel diffusion; complex XOR operation

1. Introduction

The art of converting the data (text/image/audio/video) into an unidentifiable form to make it secure from an illegal acquisition, alteration, modification, or unauthorized access while transmitting it over the network is known as cryptography [1]. In 1989, Mathews introduced a chaotic cryptosystem to meet the requirement of image encryption [2]. After that, a lot of research have been done to accomplish a secure and complex cryptosystem by utilizing its cryptographically desirable properties such as ergodicity, randomness and sensitivity to its initial values [3]. The use of chaos or randomness provides a new dimension in the cryptosystem design. 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 [4,5]. Due to simple structure, their chaotic orbits and initial values may be estimated with the least efforts [6–8]. The other group of chaotic maps contains a high-dimensional chaotic map with a rather complex structure and a 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 [9–11]. Although high-dimensional chaotic map based cryptosystem might be vulnerable to the security attacks [12,13]. Therefore, a complex and secure encryption scheme is required to employ it to the real-time applications.

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 chaotic map and the random permutation of the Jigsaw transform are required [14]. The coupled map lattice is used to generate spatiotemporal chaos, which is further encoded with DNA sequence to encrypt the image [15]. The Logistic map function is used three times to scramble the row coordinate, column coordinate and to diffuse the plain image respectively [16]. 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 [17]. The author introduced a new Sine-Logistic modulation map to generate encryption key and a chaotic magic transform method to change the pixel position of a plain image [10]. Similarly, a combined 2D-Logistic-adjusted-Sine map based cryptosystem was introduced by the author by adding a random value to the plain image [9]. Besides all the above well-known chaotic maps, a new Beta chaotic map was introduced to generate key sequence which is based on Beta function [18].

Fractals [19] are non-regular geometric shapes that have the same degree of non-regularity on all scales. Benoit Mandelbrot, in 1979 studied a very complex & perturbed structure that is known as Mandelbrot set [20]. The definition of Mandelbrot set is given in [21] as "The Mandelbrot set is the set of values of c in the complex plane for which the orbit of 0 under iteration of the complex quadratic polynomial $z_{n+1} = z_n^2 + c$ remains bounded." Fractal images exhibit the randomness property, appropriate to design a secure and reliable cryptosystem. Fractal based cryptosystem is designed using a complex number rather than the prime number, thus the generation of a private key and a public key is carried out using complex numbers arithmetically. The chaotic nature of the fractal leads to the sensitiveness of the key value towards initial value, makes it difficult to produce an accurate key by the intruder. An additional advantage of using fractal as a key is the key size, which generally impacts on the number of guesses that an attacker would need to make to find the key e.g., brute force attack i.e., it determines the feasibility of a collision attack. In the case of using the fractal key, the exchange key space depends on the size of the keys, which extend the key space, shrink the key size and make it more complex [22].

A project was carried out in 2003 to encrypt a message with the help of random numbers and Mandelbrot set fractal. In 2004, USA navy published the patent which highlights the importance of fractal as an encryption/decryption key in a cryptosystem [23]. A new approach to encryption using fractal geometry is discussed by the author in which a fractal is generated by using some initial parameters and then use it to encrypt a predetermined length of the message by using fractal orbits to corresponding alphabet mapping [24]. Although Suthikshn [25] encrypts the message using RSA in which an encryption key was generated using Mandelbrot set. A stream cipher encryption algorithm implemented on a compressed image in which a fractal dictionary encoding method is used in the image compression to achieve good quality image reconstruction [26].

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 [27]. A cryptosystem will be relatively more secure if a set of different keys is used to encrypt the plain image on each iteration [28]. In [29], multiple fractal images were used to generate key stream. The method showed an improved performance by adding several parameters: feedback delay, multiplexing and independent horizontal and vertical shifts. To encrypt image pixels, Diaconu [30] applied a bit level circular shift for pixel shuffling along with diffusion using two ciphering matrices. Later in [31] concluded that the concept proposed in [30] is vulnerable to the known/chosen plaintext attacks. To enhance the security of the original scheme, author modified the method by shifting each row of the plain image randomly and appending double crossover diffusion at the end of the original scheme. A pseudo random key stream using fractal generated in [32] by involving a non-linear network and a delay element. A non-transitional key cryptosystem using superior Mandelbrot set and relative superior Mandelbrot set was discussed in [33], which generates its private key at its own site with the help of other's public key.

The proposed encryption scheme utilized the randomness and sensitivity towards its initial condition properties of both chaotic map and fractal function. A fractal image can be generated by using unlimited online resources. The convergence point of the fractal function becomes the initial value to the newly proposed 2D-Sine Tent composite map to generate a key sequence. The algorithm used the key to shuffle and then to diffuse the plain image by applying chaotic circular pixel shuffling (CCPS) and complex XOR operation respectively. To strengthen the system, the whole process is repeated three times using three different keys on feedback mechanism. The performance analysis of the proposed scheme is done by performing various tests such as histogram distribution, correlation coefficient, Shannon entropy, differential attack measure and key sensitivity.

The rest part of the paper is organized into sections: Section 2 discusses about the methodologies used in the proposed encryption scheme. In Section 3, a detailed description of an encryption process is presented. Section 4 will show the simulation results and performance analysis statistics of the method to prove the efficiency and effectiveness towards the real-time applications. Finally, Section 5 covers discussion and conclusion of the paper by briefing the findings of the proposed scheme.

2. The Methodology

The proposed image cryptosystem consists of four following sections to achieve a secure system: (1) an Initial value generation using superior fractal function; (2) a key stream generation using 2D-STCM; (3) a plain image shuffling process using CCPS; (4) an image diffusion process by applying complex XOR operation. The further sections will discuss in detail the function of each method.

2.1. Generation of Initial Values Using Superior Fractal Function

Fractals are an infinitely complex pattern that is self-similar across different scale [20]. The Mandelbrot and Julia set are a kind of escape time fractal and constructed using same function i.e., $z^2 + c$. The only difference between two is that the Mandelbrot set is a set of points in complex c-plane starting at $z = 0$ whereas Julia set is an image for a fixed c value starting at non-zero z.

A Superior Mandelbrot set SM for a function of the form $Q_c(z) = z^n + c$, $n = 2, 3, ...$, is defined as the collection of $c \in C$, for which the superior orbit of the point 0 is bounded [34],

SM = $\{c \in C: \{Q_c^k(0): k = 0, 1, \ldots \ldots\}$ is bounded in SO$\}$.

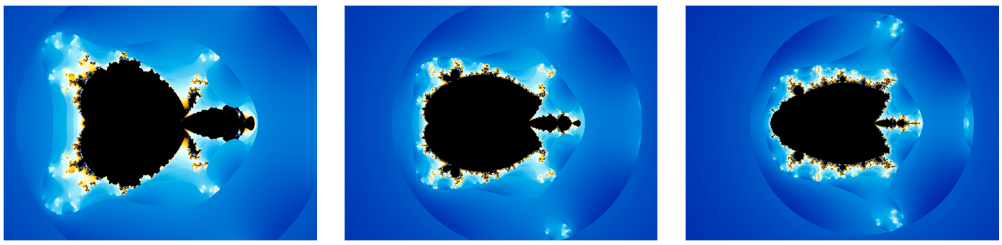
The sequence x_n constructed above is called the Mann sequence of iteration or superior sequences of iterates. We denote it by SO(x_0, s, t). Mann essentially gave this procedure in 1955 [35].

Let X be a non-empty set of real numbers and f: X->X. For x_0 belongs to X, construct a sequence $\{x_n\}$ in the following manner [36]:

$x_1 = \beta_1 f(x_0) + (1 - \beta_1)x_0$
$x_2 = \beta_2 f(x_1) + (1 - \beta_2)x_1$
...
...
...
$x_n = \beta_n f(x_n - 1) + (1 - \beta_n)x_{n-1}$

where $0 < \beta_n \leq 1$. The sequence $\{x_n\}$ constructed this way is called a superior sequence of iterates, denoted by $SO\ (f, x_0, \beta_n)$ At $\beta_n = 1$, $SO\ (f, x_0, \beta_n)$ reduces to $O\ (f, x_0)$.

The generation of fractal for $sin(z^n) + c$ is much like the standard quadratic equation of the Mandelbrot set but it consists of repeated iterations up to the n times with respect to sine function. The paper used the Mann iterated sine fractal function with respect to different β values to obtain initial values of key sequence. The corresponding generated fractal images are shown in Figure 1.



(a) (b) (c)

Figure 1. Superior Fractal images for (**a**) β = 0.3; (**b**) β = 0.5; and (**c**) β = 0.7.

2.2. Key Stream Generation Using 2D-Sine Tent Composite Map

A new 2D chaotic map is introduced to generate a key stream to encipher the plain image. The chaotic map is a combination of modified forms of 1D Sine map and 1D Tent map. A mathematical description of both 1D maps is defined in Equations (1) and (2) respectively:

$$x_{n+1} = \mu \times \sin(\pi x_n) \tag{1}$$

$$x_{n+1} = \begin{cases} rx_n, & 0 \le x < 0.5 \\ r(1-x_n), & 0.5 \le x \le 1 \end{cases} \tag{2}$$

Here μ and r are control parameters lies in the range (0, 4].

These are recursive functions, generate a repetitive set of values for each x. Earlier a combined 1D Sine-Tent chaotic map was introduced by the author for image encryption which has a better chaotic characteristic than the individual map [37]. Although the above given 1D chaotic maps are quite effective and show a complex chaotic behavior, still their orbits can be predicted by using chaotic signal estimation technologies [38,39]. To overcome the limitations of 1D chaotic map, the paper suggested a 2D form of combined Sine and Tent map. It can be defined as:

$$x_{n+1} = \begin{cases} ((\sin(\pi y_n) + 3) \times x_n/2) \; mod \; 1, & x_i < 0.5 \\ ((\sin(\pi y_n) + 3) \times (1-x_n)/2) \; mod \; 1, & x_i \ge 0.5 \end{cases} \tag{3}$$

$$y_{n+1} = \begin{cases} ((\sin(\pi x_{n+1}) + 3) \times y_n/2) \; mod \; 1, & y_i < 0.5 \\ ((\sin(\pi x_{n+1}) + 3) \times (1-y_n)/2) \; mod \; 1, & y_i \ge 0.5 \end{cases} \tag{4}$$

where control parameter r lies in (0, 4].

Trajectory

Trajectory is a pictorial representation of the sequence of values calculated by iterating application of a mapping function to an element of its source [40]. A trajectory map is shown to ensure that the used 2D Sine-Tent composite function has excellent chaotic behavior. A Figure 2 compares the trajectories of 2D-STCM, 2D-SLMM [10] and 2D Logistic map [41] defined in the given functions respectively.

2D-SLMM Function:

$$\begin{aligned} x_{n+1} &= \propto (\sin(\pi y_n) + \beta) x_i (1 - x_i) \\ y_{n+1} &= \propto (\sin(\pi x_{n+1}) + \beta) y_i (1 - y_i) \end{aligned} \tag{5}$$

2D-Logistic map:

$$\begin{aligned} x_{n+1} &= r(3y_n + 1) x_i (1 - x_i) \\ y_{n+1} &= r(3x_{n+1} + 1) y_i (1 - y_i) \end{aligned} \tag{6}$$

The graph shows that the trajectory of 2D-STCM covers a large region in the phase plane as compared to other two. Hence it proves that the proposed chaotic function has better randomness and

ergodicity property. In this paper, the chaotic map 2D-STCM receives initial values from a superior fractal function and will generate a secure key sequence to encrypt the given plain image.

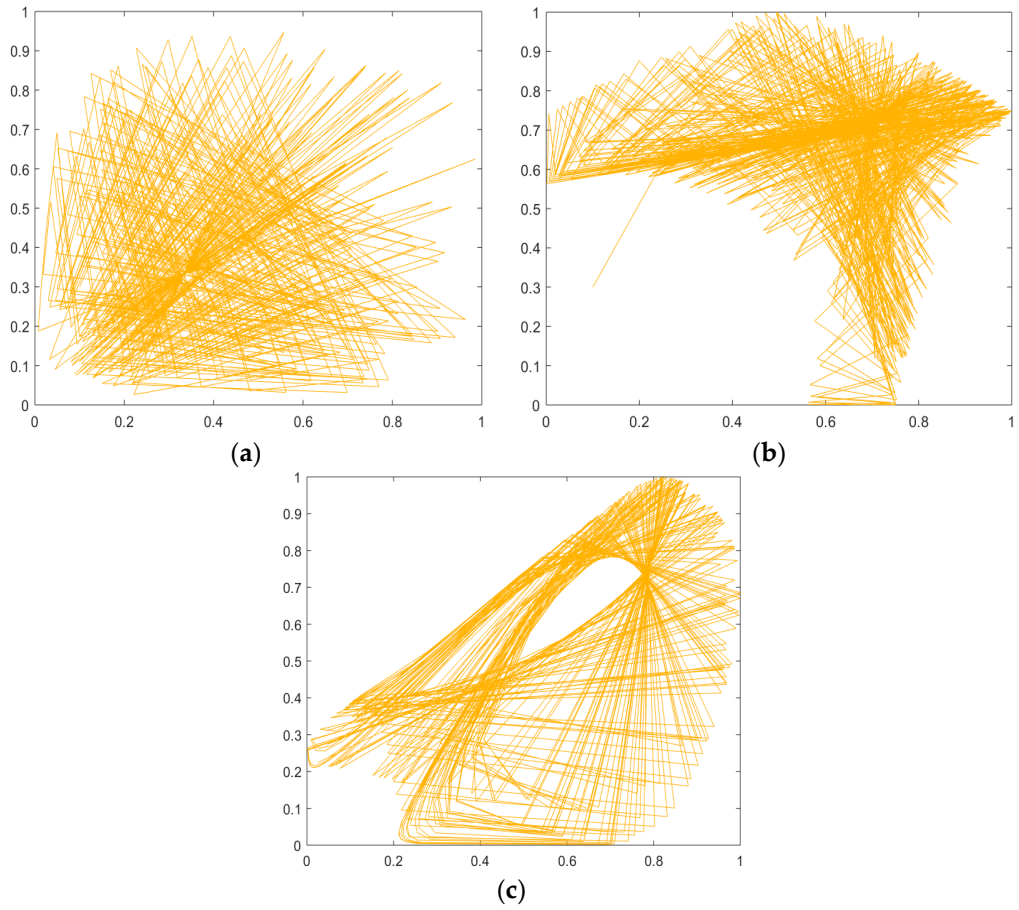


Figure 2. Trajectories of (**a**) the proposed 2D-STCM Map; (**b**) 2D-SLMM Map and (**c**) 2D-Logistic Map.

2.3. Chaotic Circular Pixel Shuffling (CCPS)

This phase deals with the pixel position shuffling within the image. It helps to reduce the pixel correlation present due to the high redundancy in the image. A chaotic circular pixel shuffling (CCPS) method used the chaotic key sequence generated in the previous step to randomly shuffle the pixel positions in the plain image. The idea derived from the chaotic magic transform method [8] but CCPS iterate the whole process multiple times based on a parameter 'w' to increase the efficiency of the method.

Let P be a plain image of size $M \times N$ and CS is a chaotic key sequence generated by 2D-STCM of the same size. The resultant shuffled matrix SM will be:

$$SM = CCPS(P, CS)$$

CCPS method changed the pixel position within the plain image randomly and circularly. The Algorithm 1 describes the image shuffling process using CCPS method.

Algorithm 1. Plain image shuffling using CCPS method

Input: A plain image P of size $M \times N$ and a chaotic key sequence CS of the same size

1. Sort the chaotic key sequence matrix column-wise and obtain the index matrix IM.
2. Set $w = 2$.
3. For $i = 1$ to M do
4. Shuffle the pixel positions in a plain image using index matrix according to the locations ($IM_{i,1}$, 1), ($IM_{i,2}$, 2), ... ($IM_{i,N}$, N) in circular manner
5. End For
6. Compute $w = w \times 2$
7. Repeat steps (3) to (6) until $w < M$

Output: Shuffled plain image (SM)

In Figure 3, an example is shown with the help of an 8×8 sample matrix. The process starts with the sorting of chaotic key sequence column-wise and save its index matrix for further use. In the next step, permute the location in P using the index matrix IM. For example, consider the first row of IM as (2, 8, 5, 4, 2, 3, 3, 6) then the pixel locations of P ((2,1), (8,2), (5,3), (4,4) ... (6,8)) will be connected into the circle and will shift 1pixel positions to the left. Similarly, on second iteration, the pixel positions will get permuted with the left shift of 2 pixels and so on.

164	82	155	155	91	164	247	82
155	82	91	164	7	82	155	73
155	82	155	164	164	7	82	164
155	82	91	73	247	91	82	164
155	82	73	155	164	82	155	164
155	82	82	82	164	73	164	7
155	73	82	73	164	155	164	155
164	0	73	91	164	247	91	82

(**a**)

2	8	5	4	2	3	3	6
3	7	8	7	1	6	4	2
4	1	6	6	3	2	8	1
5	2	7	8	5	5	2	8
6	3	2	1	6	4	5	7
7	4	4	5	7	7	6	3
1	5	1	2	8	1	7	4
8	6	3	3	4	8	1	5

(**b**)

128	215	69	167	107	60	205	191
103	132	76	119	58	143	176	117
110	148	150	166	54	69	200	116
92	98	133	72	76	203	130	115
74	84	58	133	154	133	146	90
88	31	76	120	91	180	69	39
133	25	89	219	16	95	96	77
93	29	49	75	96	41	67	53

(**c**)

115	54	84	88	130	96	205	76
29	133	77	69	69	92	89	25
49	146	150	166	191	200	39	95
120	116	133	58	76	76	110	96
154	128	72	98	74	132	167	90
203	31	143	67	148	117	16	103
69	219	176	180	133	133	60	91
93	58	107	53	119	41	215	75

(**d**)

Figure 3. A sample of plain image shuffling using CCPS. (**a**) Chaotic key sequence (CS); (**b**) Index matrix (IM); (**c**) Plain image (P) and (**d**) Plain shuffled image (SM).

2.4. Complex XOR Operation

A variation of XOR operation is introduced to enhance the security of the cipher image. A complex XOR operation works on the even/odd pattern of the pixel indexes. For every even pixel position, the computation of pixel value will be changed from the pixels present on the odd position. It helped to increase the complexity of the computation, subsequently difficult to de-cipher the cipher image. See Algorithm 2 for detailed description of complex XOR operation method.

Algorithm 2. Complex XOR operation

Input: A shuffled plain image SM of size $M \times N$ and a chaotic sequence CS of the same size

1. for k = 1 to 3 do
2. Set CS(k) = mod(floor(CS(k) × 2^32), 256)
3. for I = 1 to M do
4. for j = 1 to N do
5. If (i%2 == 0 and j%2 == 0)
6. Compute Re = XOR(SM(i, j), CSk(i, j))
7. Compute Re = (Re (i, j) + CSk (i, j)/2) mod M
8. Else
9. Compute Re = XOR(SM(i, j), CSk(i, j))
10. Compute Re = (Re(i, j)+ CSk(i, j) × 2) mod M
11. End if
12. End for (inner j loop)
13. End for (i loop)
14. End for (outer k loop)
15. Result = ~RE

Output: A cipher image (Result)

3. A New Cryptosystem Using Fractal Function

So far, the paper discussed about the methodology to design a secure image cryptosystem. Generally, to generate a cipher image, one may need an encryption key and a plain image. According to Figure 4, our key generation process consists of two steps:

1. Obtain initial values using a superior fractal function.
2. Generate a chaotic key sequence using proposed 2D-STCM.

Next step is to encrypt the plain image using the key stream generated in the previous step. This phase again consists of two steps:

3. Plain image pixel shuffling by CCPS
4. Perform complex XOR operation to shuffled image using chaotic key sequence.

The Algorithm 3 represents step by step execution of proposed image encryption process.

Algorithm 3. The proposed encryption algorithm

Input: A superior fractal function and the plain image P of size $M \times N$

1. Obtain three sets of initial values (x, y) by iterating superior fractal function three times for different values of s (i.e., s = 0.3, 0.5 and 0.7) using Equations (7) and (8).
2. Generate three chaotic key sequences CS1, CS2 and CS3 using 2D-STCM (Equations (3) and (4)) for all three sets of initial values mentioned in step (1).
3. for i = 1 to 3 do
4. Apply CCPS discussed in Section 2.3 to shuffle the plain image P using the chaotic key sequence CSi.
5. Execute complex XOR operation discussed in Section 2.4 to diffuse the shuffled image using chaotic key sequence CSi.
6. End For

Output: An encrypted image (C)

The flowchart in Figure 4 depicts the proposed cryptosystem with the various schemes used in the encryption process.

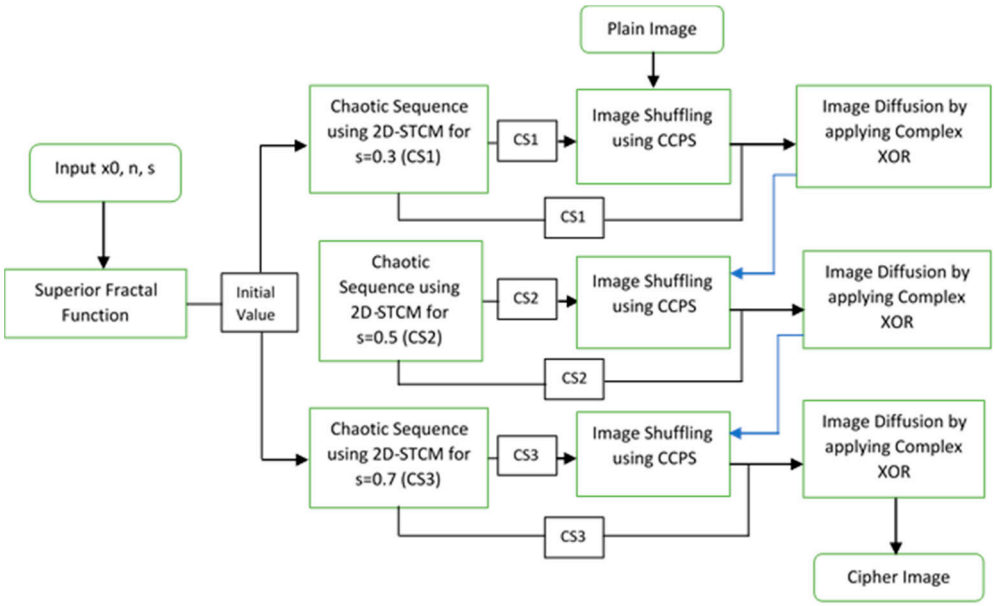


Figure 4. The proposed encryption processes.

3.1. Key Sequence Generation

The encryption/decryption key of the proposed method is generated by using a new 2D-STCM function by inputting the initial values derived from the superior fractal set. A fractal function works on a feedback system in which an output becomes input to the function on each iteration. While iterating the function, gradually it converges to a fixed point. The initial values used in 2D-STCM is nothing but the obtained fixed point after iterating the given function for certain times.

$$z_{i+1} = s \times f(z_i) + (1 - s) \times z_i \tag{7}$$

where

$$f(z) = \sin(z^n) + c \tag{8}$$

z and c both are complex numbers and n is a real number. A s is a control parameter, ranges $0 < s < 1$ and convergent to a non-zero number.

The above Mandelbrot fractal equation with sine function was used to create beautiful fractal images as well as analyzed in the paper [42] to calculate its convergence rate. The sample fractal images have been shown in Figure 1. The analysis results showed that for a value of parameter c, it converges very fast, hence suitable to fasten the key generation speed with much complexity. The paper uses three different values of control parameter s to generate three sets of initial values. For each s, fractal function will converge to a fixed point for different number of iterations. The implementation results depict that the function converges rapidly as s increases, consequently, less time is required to generate initial values. For example, obtained set of initial values are I1 = {0.1880586, 0.0352573, 0.3, 256, 256}, I2 = {0.1899738, 0.0367336, 0.5, 256, 256} and I3 = {0.1908194, 0.0374243, 0.7, 256, 256} which were obtained after 165, 88, 58 number of iterations respectively.

The set of initial values (x, y, s, M, N) passes to the 2D-STCM for generating a chaotic key sequence. As a result, three chaotic key sequence for each set of initial values will be generated by the chaotic map. A sample matrix of 4×4 is shown in the Figure 5 as follows.

0.3594	1.2309	0.5869	1.0313
0.6029	0.3846	1.1521	1.1658
1.0393	0.7182	0.424	0.3696
0.4971	1.4171	0.7903	0.5954

(CS1)

0.3657	1.1551	0.4921	0.6354
0.6156	0.9566	0.8166	1.2224
1.0627	0.3746	0.4061	0.3888
0.4993	0.6529	0.7638	0.7272

(CS2)

0.3686	1.0872	0.9907	0.3937
0.6214	1.102	0.6135	0.6825
1.0734	0.3925	1.0537	1.1992
0.5002	0.642	0.4774	0.5795

(CS3)

Figure 5. Sample chaotic key sequences generated using 2D-STCM.

3.2. Encrypting the Plain Image

An encryption algorithm with confusion property achieved through changing the pixel positions in the image. In the paper, image shuffling is achieved by applying a chaotic circular pixel shuffling method (CCPS). The image pixel shuffling helps to reduce the correlation between the adjacent pixels within the image. An Algorithm 1 explained the process in detail and a sample output of shuffled image was also given in Figure 3.

The final step of converting a plain image into a cipher image is the execution of a complex XOR operation to change the pixel values of shuffled plain image. The process of changing pixel values is known as diffusion, which could resist the chosen-plaintext attack. A detailed XOR operation execution has already been discussed in Algorithm 2.

For an ideal security performance of the proposed scheme, the image shuffling process and the image diffusion process have repeated three times using different chaotic key sequence CS1, CS2 and CS3. The result of each operation in a sample matrix of size 4×4 is shown in Figure 6.

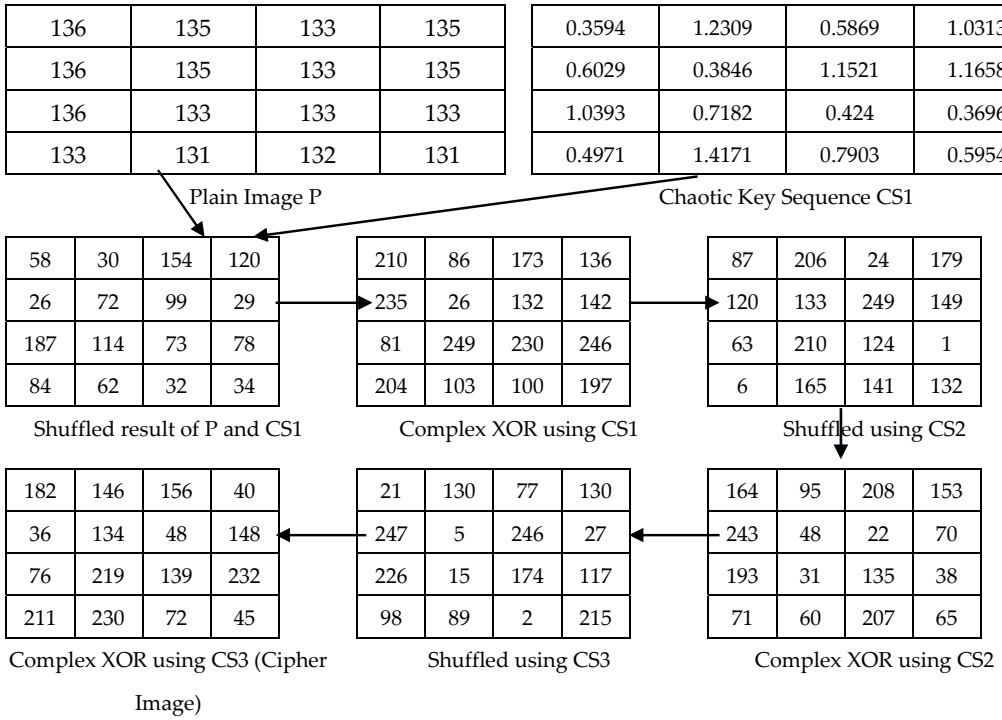


Figure 6. An example of the proposed image encryption algorithm.

After performing three rounds of image pixel confusion and diffusion using three different chaotic key sequence, a plain image is encrypted into a complex and unidentifiable form of image known as cipher image.

3.3. Decryption Process

Decryption is the reverse order of an encryption process, noting that the iteration and the operations reversed accordingly. To encrypt an image, the method executes two major steps:

1. Image shuffling by changing the pixel position
2. Image diffusion by changing the pixel values

The decryption process will start in reverse order so to get an original image first do image diffusion by performing complex XOR and then image shuffling using CS3 in both. Repeat the process of diffusion and confusion three times by using each chaotic key sequence in reverse.

A confusion/diffusion process in encryption starts with the first pixel of plain image which presents on top-left corner of the image with the directions left to right and top to bottom. Whereas, in decryption process, confusion process will start from the last pixel of a cipher image with the direction right to left and bottom to top.

4. Experimental Results and Analysis

The proposed algorithm encrypt/decrypt the plain image by applying confusion and diffusion process using multiple steps. To verify the efficiency of the given method, various tests have been carried out using MATLAB software with system configuration Intel® Atom™ x7-z8700 CPU @1.60 GHz and 4 GB RAM.

4.1. Encryption and Histogram

An implementation of the algorithm is carried out to represent the result of an encryption and decryption process. The proposed method is applied to various images of same size i.e., 256 × 256. Figure 7 shows the simulation results of execution in the form of a plain image and cipher image with its corresponding histogram.

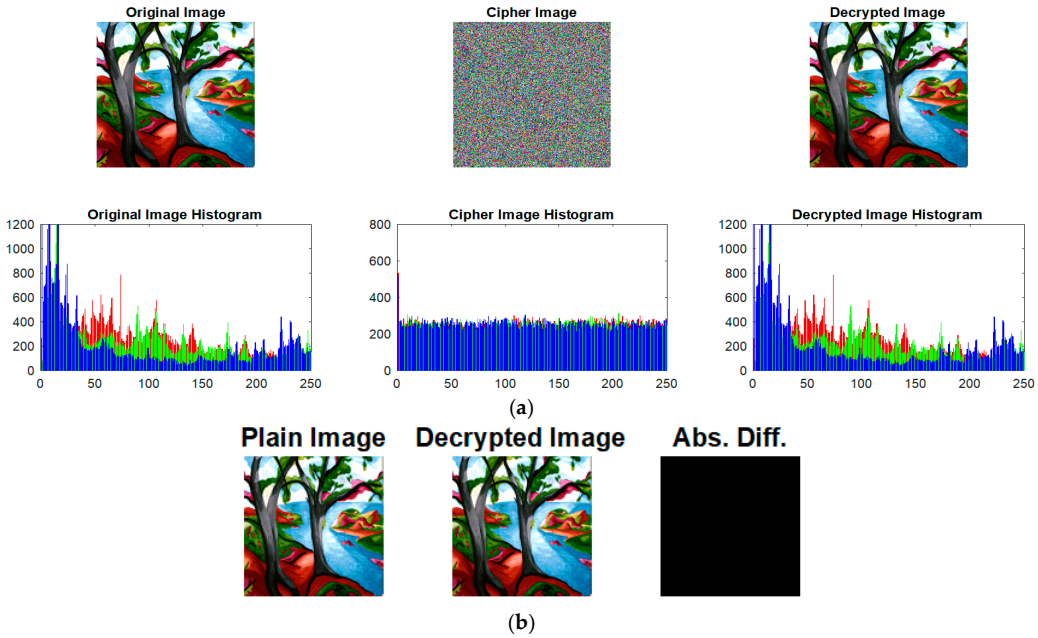


Figure 7. (**a**) Plain image, Cipher Image and Decrypted Image and their Histogram (**b**) Absolute Matrices Difference between Plain Image and Decrypted Image.

4.2. Key Space Analysis.

The effect of proposed method can be realized by seeing the cipher image and its histogram. A histogram of a plain image has a specific pattern which gives an idea about the image structure. An evenly and randomly distributed pixel values of the cipher image prove the randomization of the output value. Therefore, a histogram generated by uniform pixel value distribution makes it difficult to obtain any information by the hacker. Also, an absolute matrices difference between the plain image and the decrypted one is also shown in the below figure.

A chaos based cryptosystem has a wide range of key space, helps to resist the brute-force attack. As suggested in [43], a key of size 2^{100} is enough to resist brute force attack. The security key is generated with the help of 2D-STCM by inputting initial values derived from a superior fractal function. A key space consists of the size of parameters of the fractal function. A function has three parameters at a time i.e., x_0 (starting value), n (number of iterations) and s (control parameter). However, the method generates three chaotic key sequence using control parameter values 0.3, 0.5 and 0.7. In that case, total five keys have been used in the implementation. They all are stored in double data type and the required memory space of one parameter is 8 bytes or 64 bits. Hence, the key space size of the proposed cryptosystem will be 2^{320} which is quite sufficient to resist the brute force attack.

4.3. Encryption/Decryption Computational Time Analysis

The proposed scheme is implemented on the multiple color images of size 256 × 256. The system configuration used in the experiment is equipped with Intel® Atom™ x7-z8700 CPU @1.60 GHz and 4 GB RAM running Windows 10 (64-bit). A fractal based key required multiple rounds of iteration to achieve a fixed point. In the proposed scheme, a fixed point achieved through the superior fractal function convergence and then inputted into the 2D-STCM to generate a key sequence. Further, image shuffling and diffusion steps required three consecutive iterations to generate a cipher image in encryption and a plain image in decryption process. The approximate time to encrypt/decrypt the 256 × 256 color image is calculated 6.5–7.0 s.

4.4. Randomness Test

The information entropy was given in 1949 by Shannon and is a statistical measure to estimate the randomness and unpredictability of an information source [44]. The message entropy $H(s)$ of message source s is defined as:

$$H(s) = -\sum_{i=0}^{2^N-1} P(s_i) log_2 P(s_i), \quad (9)$$

Here $P(s_i)$ is the probability of symbol s_i and N is the number of bits to represent a symbol si. Ideally, the entropy value of a random image having $2N$ symbols, is expected to be $H(s) = N$. Accordingly, a ciphered image with 256 gray levels should be 8. An entropy value, less than 8 depicts a certain degree of predictability of the cipher image. The entropy of various cipher images is shown in the Table 1. Results showed that the entropy of the output ciphered images is nearby to the standard entropy value.

Table 1. Information entropy of plain images and cipher images.

Name	Chip	House	Pepper	Tree	Mandrill
Actual Entropy	7.4140	7.0686	7.6339	7.6140	6.8178
Ciphered Entropy	7.9942	7.9986	7.9972	7.9988	7.9952

4.5. NPCR and UACI Tests

NPCR (Number of pixels change rate) and UACI (Unified average changing intensity) are standardized tests to analysis a plain image sensitivity so that differential attack can be resist [45].

NPCR value is used to test the influence of the petty change in plain image pixels causes a huge difference in the corresponding cipher images. Let's consider the plain image "PI1" and "PI2" with small change in pixel values and the cipher image "CI1" and "CI2" of size $M \times N$ respectively, then the value of a bipolar array D with the same size as of "CI1" and "CI2" will be calculated as:

$$D(i,j) = \begin{cases} 0, & if\ \mathrm{CI1}(i,j) = \mathrm{CI2}(i,j) \\ 1, & if\ \mathrm{CI1}(i,j) \neq \mathrm{CI2}(i,j) \end{cases}, \tag{10}$$

The NPCR is defined as:

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,\ j) \times 100\%, \tag{11}$$

The greater the value of NPCR results a better plain image sensitivity. Ideally, the NPCR value of a true random cipher image should be around 100.

The UACI is defined as:

$$\mathrm{UACI} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{IM(i,j) - SCR(i,j)}{255} \times 100\%, \tag{12}$$

UACI test calculates the average intensity change between a plain image and a cipher image which is expected to be around 33. The results are shown in Table 2 which indicates that the proposed scheme is secure from the differential attack.

Table 2. NPCR (number of pixels change rate) and UACI (unified average changing intensity) values of cipher images.

Image Name	NPCR (%)	UACI (%)
Chip	99.5580	33.0396
House	99.6099	33.4819
Pepper	99.4207	33.3457
Tree	99.5504	33.2621
Mandrill	99.1608	33.1975

4.6. Key Sensitivity Analysis

A security key plays an important role to decide the security level of the cryptosystem. An encryption method must have a key to large sufficient key space and must be extremely sensitive to the minor change in its value. In an earlier section, a discussion has been carried out about the importance and the available key space. Now a detailed analysis will be done around the key sensitivity property of the key used in the proposed method.

A key sensitivity has a significant role in encryption phase as well as in decryption phase. (1) While doing encryption, a small change in security key must produce a totally different cipher image as compared to a cipher image, encrypting with an exact key; (2) While doing decryption, a small change in security key must recover a totally different plain image than the decrypting with an exact key.

The paper used K: $x_0 = 0.0$, $n = 200$, $s = 0.3$, $s = 0.5$ and $s = 0.7$ as initial values to generate the security key. To verify the key sensitivity, let's do an experiment by changing the initial values of security key slightly. Assume:

1. K1: $x_0 = 0.0001$, $n = 200$, $s = 0.3$, $s = 0.5$ and $s = 0.7$
2. K2: $x_0 = 0.000001$, $n = 200$, $s = 0.3$, $s = 0.5$ and $s = 0.7$
3. K3: $x_0 = 0.00000001$, $n = 200$, $s = 0.3$, $s = 0.5$ and $s = 0.7$

The below I part of Figure 8 shows the encrypted and decrypted images obtained after making a slight change in the security key. The decrypted image 2 is obtained by changing x_0 from 0.0 to 0.0001 and that is totally changed from the expected one. Even after changing the key initial value by 10^{-6}, nobody can guess the actual plain image from the obtained decrypted image 3. The output images are completely different and unrecognizable, which ensure the key sensitivity in both encryption and decryption process.

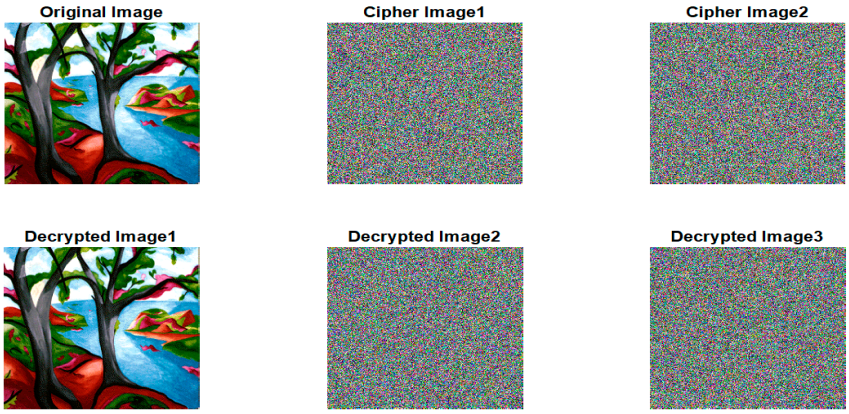


Figure 8. Original image; Cipher Image1 = encrypt(P, K); Cipher Image2 = encrypt(P, K1); Decrypted image using correct key; Decrypted Image2 = decrypt(C, K1); Decrypted Image3 = decrypt(C, K2); (From top to bottom and left to right).

An additional test is executed to prove the key sensitivity in terms of NPCR and UACI calculation. The Table 3 shows the NPCR and UACI between cipher images encrypted with correct key and the modified key assumed in (1), (2) and (3). The results showed a significant difference between two ciphered images, either by the difference of 10^{-4}, 10^{-6} or 10^{-8}. In that case, if an attacker tries to decrypt the image using modified key, it will be completely fail.

Table 3. NPCR and UACI values of cipher images after modifying secret key.

Change in Key Value	NPCR (%)	UACI (%)
x_0 = 0.0001	99.6053	33.4245
x_0 = 0.000001	99.6109	33.3863
x_0 = 0.00000001	99.5936	33.4691

4.7. Adjacent Pixel Correlation Analysis

The Correlation coefficient represents the relationship between two adjacent pixels in an image. The image pixels are highly redundant, that's why have a strong correlation between adjacent pixels. In contrary, a cipher image should have a low correlation 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:

$$cc = \frac{cov(x,y)}{\sigma x * \sigma y}, \tag{13}$$

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

$$var(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2, \tag{14}$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \tag{15}$$

Here, x and y are adjacent pixels in a plain image or cipher image of size $M \times N$. The Figure 9 shows the distribution of adjacent pixel pairs of the plain image and its cipher image in horizontal, vertical and diagonal direction. The pixel pairs of plain image are mostly located nearby the diagonal line in the graph, means they all are close to each other or have equal values. Whereas, in case of cipher image, pixels are randomly distributed covering entire data range, depicts that they are not related to each other. The low correlation between the pixels makes it non-vulnerable by the attacker only by getting the adjacent pixel pair information.

Table 4 shows the horizontal, vertical and diagonal correlation coefficient value of adjacent pixels of plain image and its corresponding encrypted image. Ideally, the coefficient value in case of plain image must be close to 1 and for cipher image, it must be close to 0. The quantitative result of correlation coefficient has the desired values for the plain image and its cipher image as well, hence proved the suitability of the proposed algorithm for the image encryption application.

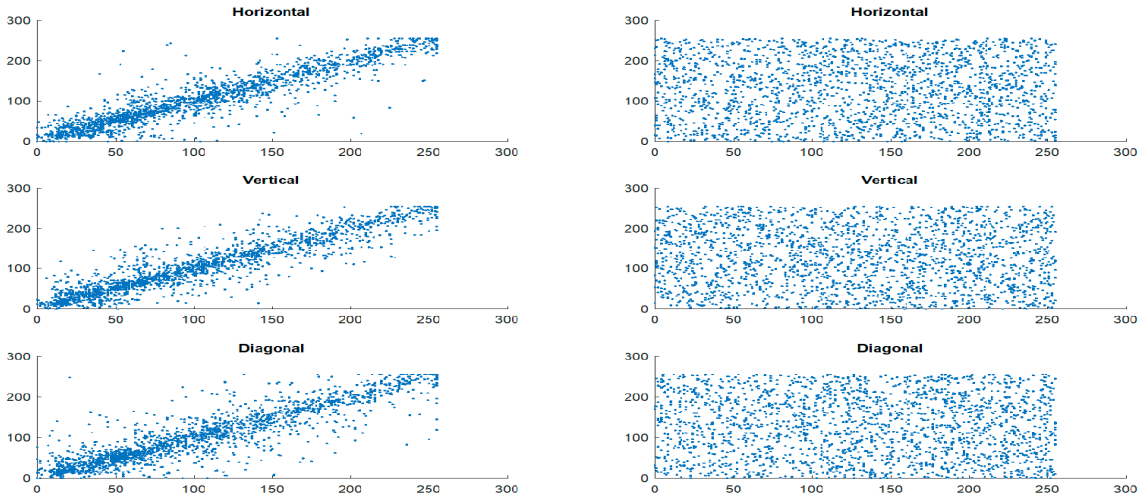


Figure 9. Adjacent pixel pair distribution of a plain image and its cipher image in horizontal, vertical and diagonal direction.

Table 4. Pixel correlation coefficient values of the plain image and its cipher image.

Image Name	Plain Image			Cipher Image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Chip	0.9698	0.9799	0.9627	−0.0029	0.0052	−0.0054
House	0.9670	0.9352	0.9126	0.0033	0.0032	−0.0048
Pepper	0.9943	0.9950	0.9882	0.0019	0.0003	0.0033
Tree	0.9430	0.9457	0.9180	0.0054	0.0013	0.0009
Mandrill	0.9309	0.9187	0.9019	−0.0084	0.0008	−0.0054

4.8. Performance Comparison with Other Image Encryption Techniques

The proposed scheme is compared with the other available color image encryption techniques in terms of information entropy, NPCR, UACI, correlation coefficient and speed in the Table 5. The results showed that the given algorithm performs similar or better than the other mentioned schemes.

Table 5. Performance Comparison between Various Image Encryption Methods.

Performance Parameter	Ref. [46]	Ref. [47]	Ref. [48]	Proposed Method
Information Entropy	7.9992	7.9962	7.9895	7.9972
NPCR	99.6101	99.57	99.7915	99.4207
UACI	33.4008	33.39	49.2191	33.3457
Horizontal Correlation	0.0607	−0.0012	−0.0036	0.0019
Vertical Correlation	−0.0011	0.0022	0.0001	0.0003
Diagonal Correlation	−0.0057	−0.0022	−0.023	0.0033
Speed (s)	10.3959	1.33	1.25	7

5. Discussion and Conclusions

This paper realizes the importance of a fractal function in the combination of a two-dimensional chaotic map to suggest an image encryption scheme. A superior fractal function required fewer iterations to generate initial values as compared to the general fractal function. The initial conditions are then keyed to a newly introduced 2D-Sine Tent composite map (2D-STCM) to generate a chaotic sequence as a key stream. It is designed by combining a 1D Sine map and a 1D Tent map with minor modifications to reduce the limitations imposed by the 1D chaotic maps. The trajectory graph of the proposed chaotic map shows the wider chaotic region in the phase plane. The generated encryption/decryption key is much complicated and has key sensitivity due to the fractal function and chaotic map properties.

The proposed cryptosystem implements both necessary steps, i.e., confusion and diffusion to achieve a high security level. The confusion phase deals with the pixel position change which was accomplished by applying chaotic circular pixel shuffling (CCPS) to the plain image. The final cipher image is obtained after the diffusion process by executing complex XOR operation on the shuffled plain image. In the entire course, three rounds of confusion and diffusion process have performed using three different key sequences to enforce more security. The scheme is tested to prove effectiveness, correctness, secure and robustness by calculating Shannon entropy, NPCR value, UACI value, adjacent pixel correlation, histogram distribution, key space analysis and key sensitivity analysis. The experimental results show the suitability of the proposed method to protect the digital image transmission over the network with high security level.

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

References

1. Stallings, W. *Cryptography and Network Security: Principles and Practice*, 4th ed.; Pearson/Prentice Hall: Upper Saddle River, NJ, USA, 2006; ISBN 978-0-13-187316-2.
2. Matthews, R. On the derivation of a "chaotic" encryption algorithm. *Cryptologia* **1989**, *13*, 29–42. [CrossRef]
3. Baptista, M.S. Cryptography with chaos. *Phys. Lett. A* **1998**, *240*, 50–54. [CrossRef]
4. Hilborn, R.C. *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*; Oxford University Press: Oxford, UK, 2000; ISBN 978-0-19-850723-9.
5. Radwan, A.G.; Abd-El-Hafiz, S.K. Image encryption using generalized tent map. In Proceedings of the 2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), Abu Dhabi, UAE, 8–11 December 2013; pp. 653–656.
6. Wu, X.; Hu, H.; Zhang, B. Parameter estimation only from the symbolic sequences generated by chaos system. *Chaos Sol. Fract.* **2004**, *22*, 359–366. [CrossRef]
7. Rhouma, R.; Belghith, S. Cryptanalysis of a spatiotemporal chaotic cryptosystem. *Chaos Sol. Fract.* **2009**, *41*, 1718–1722. [CrossRef]
8. Li, S.; Zheng, X. Cryptanalysis of a chaotic image encryption method. In Proceedings of the 2002 IEEE International Symposium on Circuits and Systems, Phoenix-Scottsdale, AZ, USA, 26–29 May 2002; Volume 2, pp. II-708–II-711.

9. Parvaz, R.; Zarebnia, M. A combination chaotic system and application in color image encryption. *Opt. Laser Technol.* **2017**, *101*, 30–41. [CrossRef]
10. Hua, Z.; Zhou, Y.; Pun, C.-M.; Chen, C.L.P. 2D Sine Logistic modulation map for image encryption. *Inf. Sci.* **2015**, *297*, 80–94. [CrossRef]
11. Hua, Z.; Zhou, Y. Image encryption using 2D Logistic-adjusted-Sine map. *Inf. Sci.* **2016**, *339*, 237–253. [CrossRef]
12. Rhouma, R.; Belghith, S. Cryptanalysis of a chaos-based cryptosystem on DSP. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 876–884. [CrossRef]
13. Solak, E.; Rhouma, R.; Belghith, S. Cryptanalysis of a multi-chaotic systems based image cryptosystem. *Opt. Commun.* **2010**, *283*, 232–236. [CrossRef]
14. Vilardy, J.M.; Jimenez, C.J.; Perez, R. Image encryption using the Gyrator transform and random phase masks generated by using chaos. *J. Phys. Conf. Ser.* **2017**, *850*, 012012. [CrossRef]
15. Wang, X.-Y.; Zhang, Y.-Q.; Bao, X.-M. A novel chaotic image encryption scheme using DNA sequence operations. *Opt. Lasers Eng.* **2015**, *73*, 53–61. [CrossRef]
16. Wang, X.; Wang, S.; Zhang, Y.; Guo, K. A novel image encryption algorithm based on chaotic shuffling method. *Inf. Secur. J. Glob. Perspect.* **2017**, *26*, 7–16. [CrossRef]
17. Kanafchian, M.; Fathi-Vajargah, B. A Novel Image Encryption Scheme Based on Clifford Attractor and Noisy Logistic Map for Secure Transferring Images in Navy. *Int. J. E-Navig. Marit. Econ.* **2017**, *6*, 53–63. [CrossRef]
18. Zahmoul, R.; Ejbali, R.; Zaied, M. Image encryption based on new Beta chaotic maps. *Opt. Lasers Eng.* **2017**, *96*, 39–49. [CrossRef]
19. Pickover, C.A. *Computers, Pattern, Chaos, and Beauty: Graphics from an Unseen World*; Courier Corporation: North Chelmsford, MA, USA, 2001; ISBN 978-0-486-41709-7.
20. Mandelbrot, B.B. *The Fractal Geometry of Nature*; Henry Holt and Company: New York, NY, USA, 1982; ISBN 978-0-7167-1186-5.
21. Crownover, R.M. *Introduction to Fractals and Chaos*; Jones and Bartlett: Burlington, MA, USA, 1995; ISBN 978-0-86720-464-3.
22. Negi, D.; Negi, A.; Agarwal, S. The complex key cryptosystem. *Int. J. Appl. Eng. Res.* **2016**, *11*, 681–684.
23. Huntress, G.B. Encryption Using Fractal Key. Grant Patent 6,782,101 B1, 24 August 2004.
24. Ivo, M.; Jasek, R.; Varacha, P. Analysis of the Fractal Structures For the Information Encrypting Process. *Int. J. Comput.* **2012**, *6*, 224–231.
25. Kumar, S. Public Key Cryptographic System Using Mandelbrot Sets. In Proceedings of the MILCOM 2006—2006 IEEE Military Communications Conference, Washington, DC, USA, 23–25 October 2006; pp. 1–5.
26. Sun, Y.; Xu, R.; Chen, L.; Hu, X. Image compression and encryption scheme using fractal dictionary and Julia set. *IET Image Process.* **2015**, *9*, 173–183. [CrossRef]
27. Mikhail, M.; Abouelseoud, Y.; ElKobrosy, G. Two-Phase Image Encryption Scheme Based on FFCT and Fractals. Available online: https://www.hindawi.com/journals/scn/2017/7367518/abs/ (accessed on 9 November 2017).
28. Oğraş, H.; Türk, M. A Robust Chaos-Based Image Cryptosystem with an Improved Key Generator and Plain Image Sensitivity Mechanism. *J. Inf. Secur.* **2017**, *8*, 23–41. [CrossRef]
29. Abd-El-Hafiz, S.K.; Radwan, A.G.; Haleem, S.H.A.; Barakat, M.L. A fractal-based image encryption system. *IET Image Process.* **2014**, *8*, 742–752. [CrossRef]
30. Diaconu, A.-V. Circular inter–intra pixels bit-level permutation and chaos-based image encryption. *Inf. Sci.* **2016**, *355–356*, 314–327. [CrossRef]
31. Fan, H.; Li, M. Cryptanalysis and Improvement of Chaos-Based Image Encryption Scheme with Circular Inter-Intra-Pixels Bit-Level Permutation. Available online: https://www.hindawi.com/journals/mpe/2017/8124912/ (accessed on 9 November 2017).
32. AbdElHaleem, S.H.; Radwan, A.G.; Abd-El-Hafiz, S.K. Design of pseudo random keystream generator using fractals. In Proceedings of the 2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), Abu Dhabi, UAE, 8–11 December 2013; pp. 877–880.
33. Agarwal, S. Symmetric Key Encryption using Iterated Fractal Functions. *Int. J. Comput. Netw. Inf. Secur.* **2017**, *9*, 1–9. [CrossRef]
34. Rani, M.; Kumar, V. Superior Mandelbrot Set. *Res. Math. Educ.* **2004**, *8*, 279–291.
35. Mann, W.R. Mean Value Methods in Iteration. *Proc. Am. Math. Soc.* **1953**, *4*, 506–510. [CrossRef]

36. Rana, R.; Chauhan, Y.S.; Negi, A. Generation of New Fractals for Sin Function. *Int. J. Comput. Technol. Appl.* **2011**, *2*, 1747–1754.
37. Zhou, Y.; Bao, L.; Chen, C.L.P. A new 1D chaotic system for image encryption. *Signal Process.* **2014**, *97*, 172–182. [CrossRef]
38. Arroyo, D.; Rhouma, R.; Alvarez, G.; Li, S.; Fernandez, V. On the security of a new image encryption scheme based on chaotic map lattices. *Chaos Interdiscip. J. Nonlinear Sci.* **2008**, *18*, 033112. [CrossRef] [PubMed]
39. Cong, L.; Xiaofu, W.; Songgeng, S. A general efficient method for chaotic signal estimation. *IEEE Trans. Signal Process.* **1999**, *47*, 1424–1428. [CrossRef]
40. Trajectory. Available online: https://en.wikipedia.org/wiki/Trajectory (accessed on 11 November 2017).
41. Wu, Y.; Noonan, J.P.; Yang, G.; Jin, H. Image encryption using the two-dimensional logistic chaotic map. *J. Electron. Imaging* **2012**, *21*, 013014. [CrossRef]
42. Agarwal, S.; Srivastava, G.; Negi, A. Dynamics of Mandelbrot set with transcendental function. *Int. J. Adv. Comput. Sci. Appl.* **2012**, *3*, 142–146. [CrossRef]
43. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151. [CrossRef]
44. Shannon, C.E. Communication theory of secrecy systems. *Bell Labs Tech. J.* **1949**, *28*, 656–715. [CrossRef]
45. Wu, Y.; Noonan, J.P.; Agaian, S. NPCR and UACI randomness tests for image encryption. *Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun. JSAT* **2011**, 31–38.
46. Zhu, H.; Zhang, X.; Yu, H.; Zhao, C.; Zhu, Z. A Novel Image Encryption Scheme Using the Composite Discrete Chaotic System. *Entropy* **2016**, *18*, 276. [CrossRef]
47. Wang, X.-Y.; Zhang, Y.-Q.; Bao, X.-M. A Colour Image Encryption Scheme Using Permutation-Substitution Based on Chaos. *Entropy* **2015**, *17*, 3877–3897. [CrossRef]
48. Wu, X.; Li, Y.; Kurths, J. A New Color Image Encryption Scheme Using CML and a Fractional-Order Chaotic System. *PLoS ONE* **2015**, *10*. [CrossRef] [PubMed]

© 2018 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).