Journal of Engineering Science and Technology
Vol. 16, No. 2 (2021) 1319 - 1341
© School of Engineering, Taylor's University

AN EFFECTIVE COLOR IMAGE ENCRYPTION SCHEME BASED ON DOUBLE PIECEWISE LINEAR CHAOTIC MAP METHOD AND RC4 ALGORITHM

DALAL W. AHMED[1, *], TALIB M. JAWAD[2], LAHIEB MOHAMMED JAWAD[2]

[1]Iraqi Commission for Computers and Informatics, Informatics Institute for Postgraduate Studies, Baghdad, Iraq
[2]College of Information Engineering, Al-Nahrain University, Baghdad, Iraq
*Corresponding Author: dalal_w_a@yahoo.com

Abstract

Fast and secure data stored and transmission through a modern communication and information system are the core objective in this area. With regard to this study, efficient and simple algorithm has been suggested that takes advantage of double piecewise-linear chaotic map method and RC4 algorithm. Each algorithm is utilized as Pseudo Random-Number-Generator (PRNG). The proposed piecewise generator produces series of M × N random sequence keys that are used as an initial key for the RC4 algorithm that has been utilized for generating another M × N key sequence. At each round, an image is converted into one-dimension, then the Fisher-Yates Shuffle algorithm has been utilized for achieving confusion operation, controlled by chaotic matrix created by double piecewise algorithm as well as diffusion is achieved by performing a complex (XOR) operation between confused image and matrix generated by RC4 algorithm. The experimental results on speed of encryption/decryption process, keys-pace analysis, entropy value, key sensitivity, plain text sensitivity, cipher-text sensitivity, and statistical analysis indicated that the suggested approach has good resistance against all known attacks like brute force, plain-text, cipher-text, statistical and differential attacks. In addition, the generated sequences passed the statistical tests on the NIST suite. So that this scheme is applicable for transmitting digital image securely over any communication system.

Keywords: Chaos methods, Fisher-yates shuffle algorithm, Piece wise linear chaotic map (PWLCM), RC4 algorithm, Stream cipher.

1. Introduction

Currently, many different types of information are transmitted through the internet. This involves not just text files, but also digital audios and images [1], it is necessary to ensure that this information is transmitted securely. The technical way to secure this information is by cryptography [2]. For text information, the majority regarding classical encryption algorithms including AES, IDEA, and DES are very suitable. However, digital images, in comparison to texts are having certain features, like huge data volume, many redundant data and a very strong correlation among adjacent pixels [3-5]. So that, such algorithms are not effective for image data because they need much more processing power, bandwidth, and long time during the encryption process, resulting in low performance and substantial latency [6-8].

Meanwhile, the chaotic systems first discovered by EN Lorenz in the 1970s have some properties that make them fundamental components to construct cryptosystems such as sensitivity to initial values and control parameters, ergodicity, and non–periodicity. These properties somehow are related to the Image encryption system's properties, such as sensitivity to the secret key, sensitivity to plain text, and so on. Therefore, chaotic structures are commonly used in image encryption algorithms for generating the key sequences used for encryption [9].

Ideally, a chaos-based algorithm is mainly divided into two stages: permutation (confusion) and diffusion. For reducing correlations among the adjacent pixels, each pixels' location in original image is changed through the confusion stage. For spreading the impact regarding specific pixels of image, the pixels value is modified during the diffusion stage [10]. The chaotic maps can be classified into two types, one dimension (1D) and multi dimensions. One-dimension chaotic maps are simple structure, fast, and easy to implement. Despite these advantages, the main limitation of the 1D chaotic maps is that when using them for generating the PRNG, the key space is usually small and leads to the development of weak encryption algorithms. On the other hand, multi dimension chaotic maps have larger key space, but with difficulties in hardware/software implementation. Moreover, the computational complexity is increased when these chaotic maps are used for encryption algorithms [11].

The methods of image encryption on the basis of the chaotic maps were of high importance to a lot of studies due to their significant properties including pseudo-randomness, ergodicity, unpredictability, and extremely sensitive dependence on the control and initial conditions [12]. That is meeting the requirements of Shannon associated to diffusion and confusion in case of image encryption [7]. Such excellent properties making the algorithms of chaos-based encryption very important with regard to security and speed [13].

Recently, several pseudorandom number generators (PRNG) are proposed based on the chaotic stream cipher systems, like logistic map or tent map. The majority regarding PRNGs based on chaos have been directly acquired through sampling the trajectory that is related to chaotic map. With regard to such condition, certain information related to chaotic map has been possibly exposed, that result in certain loopholes. Furthermore, approaches to predict the chaotic time series were provided including the fuzzy techniques [14, 15]. In addition, complex chaotic systems could be taken into account for the purpose of preventing attackers from violating the PRNG by predicting chaotic series. In view of totally utilize the characteristics regarding complex systems, algorithms satisfying PRNG's security requirements. Yet, in the case of producing pseudorandom numbers algorithm, additional computation are necessary and further analysis of PRNG trade-offs between safety and efficiency was also required [16].

Concerned about the above issue, the presented work presenting novel colour image encryption method by combining double piecewise linear chaotic map with RC4 algorithm, and the reason behind this combination is that RC4 is very simple and fast compared to other encryption algorithms and requires only byte-length manipulations so it is suitable for embedded systems. Although RC4 has vulnerabilities related to the initial permutation of the S array and the permutation process of the S array as well as of being reversible [17], we combined it with chaotic systems to make it almost impossible to break. The major motivation for this article is the new encryption scheme based on generating a set of key sequences using cross-coupled PWLCM combined with RC4 stream cipher to encrypt M×N coloured image

i. The key generation of RC4 is accomplished in a very innovative way by using a chaotic map sequence key and the use of complex XOR operation makes the algorithm impossible to be broken or reversed.

ii. This simple method has been computationally proved to be secure and efficient for transmitting coloured images through the internet compared with other algorithms and can be considered fast when implemented on computers with high-speed and advanced specifications.

iii. The rest of this study has been provided in the following way: Section 2 providing related works. Section 3 providing preliminaries. Section 4 providing the suggested PRNG generators, decryption and encryption stages related to the suggested algorithm. Section 5 providing security as well as simulations. Section 6 summarizing the scheme’s conclusion.

2. Literature Survey

In recent years, many researchers are focusing on chaotic map methods in image encryption techniques, which are considered as modern techniques. In addition, traditional encryption algorithms are still suitable for protecting the image. Therefore, this section describes several image encryption techniques that are proposed in previous years.

Hanchinamani and Kulkarni [10] presented an effective approach for image encryption utilizing Peter De Jong chaotic map during the permutation operation and for generating the initial key of RC4 stream ciphering generator, which was utilized as PRNG to generate another key used for rotating the pixel values and in diffusion stage. The encryption process is consisting of three main stages. In the first stage, every row and column are scrambled with circular rotation in alternate orientations while in the second stage; every pixel is rotated using the M x N PRNG key. The last stage implements double diffusion on the diffused pixels by scanning the image at forward and backward orientations using two PRNG keys. The testing results proved that this scheme is computationally fast and secured.

A novel algorithm was employed by Wu et al. [18] employed A novel algorithm for enhancing the robustness and the security of coloured image encryption on the basis of the coupled-map lattices in addition to fractional-order chaotic map. For making the process of encryption more complicated, the process of image division-shuffling is proposed, that divides the plain-image in to 4 sub-images, after that shuffling the positions that are relate to all the image pixels. Furthermore, a 280-bit secret key has been utilized for generating control parameters and the initial conditions of the two chaotic systems. Results revealed that the suggested approach had excellent robustness against certain image geometric attacks.

A study by Wang et al. [19] suggested encryption approach with regard to coloured images through the use of spatiotemporal chaotic system. Firstly, a matrix has been created with the use of R, G and B channels of a colour plain-image. After that, such matrix has been scrambled with the use of zigzag scan. Then a substitution process is performed on the resultant matrix to produce the ciphered coloured image.

Zhu et al. [20] provided new approach of image encryption on the basis of novel 2D Composite Discrete Chaotic System (CDCS) that consists of two parts: in the first part, they proposed and analysed the chaotic behaviours of the new 2D CDCS, then introduced bit-level confusion with pixel-level diffusion encryption scheme. With regard to diffusion procedure, random values as well as information from plain image were added for improving the suggested scheme's security.

A technique presented by Abbas and Mohammed [21] to extract the secret key that has been utilized for encryption from image's contents, in this way no need to search for a secured channel for exchanging the key. This key was calculated from the entropy values of random blocks, which have been chosen from plain image. The evaluation of strength as well as the performance of the technique showed that the suggested approach has been efficient to be utilized in image authentication and image security.

Sahari and Boukemara [22] proposed a new 3-D chaotic map approach on the basis of coupling piecewise as well as the logistic maps that have been utilized for implementing novel Chaotic Pseudo-Random Number Generator (CPRNG). Moreover, coloured image encryption application has been suggested where encryption key has been associated to plain images in addition to being utilized in diffusion and confusion. Such random key successfully passed the randomness test suite of NIST SP 800 - 22; also, the experimental results indicated that the suggested cryptosystem showed effective performance with regard to robustness, security as well as sensitivity.

Agarwal [23] provided novel scheme of chaotic map for generating keystream through merging superior fractal function with novel 2D-Sine Tent Composite Map (2D-STCM) for the purpose of enhancing the image transmission's security over network. The process of encryption includes 3 rounds related to diffusion and confusion with the use of 3 distinctive key sequences for the purpose of obtaining extra security. Furthermore, the confusion stage has been achieved through utilizing Chaotic Circular Pixel Shuffling (CCPS) on the original-image. With regard to diffusion stage, complex XOR operation has been made on confused image. Experimental results indicated the adequacy regarding the suggested algorithm to be utilized for encrypting the digital images that are transmitted through the networks with high-security levels.

Herbadji et.al [13] proposed a colour image encryption algorithm using logistic-map and quadratic map by initially dividing the image into 3 components (R, B, and G), then these components are combined as a single grey scale image. Two permutation index vectors have been used for the process of confusion and diffusion for both rows and columns of the plain-image. The experimental results showed that this scheme could minimize the computed work and time.

Wang and Sun [24] proposed an improved Joseph traversal method by adding parameter spaces to the standard Joseph traversal and binding other parameters with the plain-image, which increase the randomness and efficiency of the scrambling effect on the ciphered image. Moreover, this article proposes a technique of dynamic cyclic shift to increase the randomness of the no. of bits of the image's binary pixel sequences.

3. Preliminaries

The proposed scheme for encryption utilizes a double piecewise linear map and RC4 stream cipher for generating the PRNG as well as the Fisher-Yates shuffling algorithm.

3.1. Piecewise linear map

This can be considered as the simplest and major 1D chaotic map that used to generate pseudo-random sequences and it was extensively applied in variety image encryption algorithms for having good statistical properties such as uniformed invariant distributing process and excellent ergodicity, and determinacy. Piecewise map is defined by the following equations [25]:

$$X_{i+1} = \begin{cases} \frac{X_i}{p} & if\ 0 \le X_i < p \\ \frac{X_i - p}{0.5 - p} & if\ p \le X_i < 0.5 \\ \frac{(1-p-X_i)}{0.5-p} & if\ 0.5 \le X_i \le 1-p \\ \frac{(1-X_i)}{p} & if\ 1-p \le X_i \le 1.0 \end{cases} \tag{1}$$

where X_i represent the initial condition of the map, $X_i \in [0, 1]$, while p represents the control parameter ‘probability’ and $p \in [0, 0.5]$.

3.2. RC4 stream cipher generator

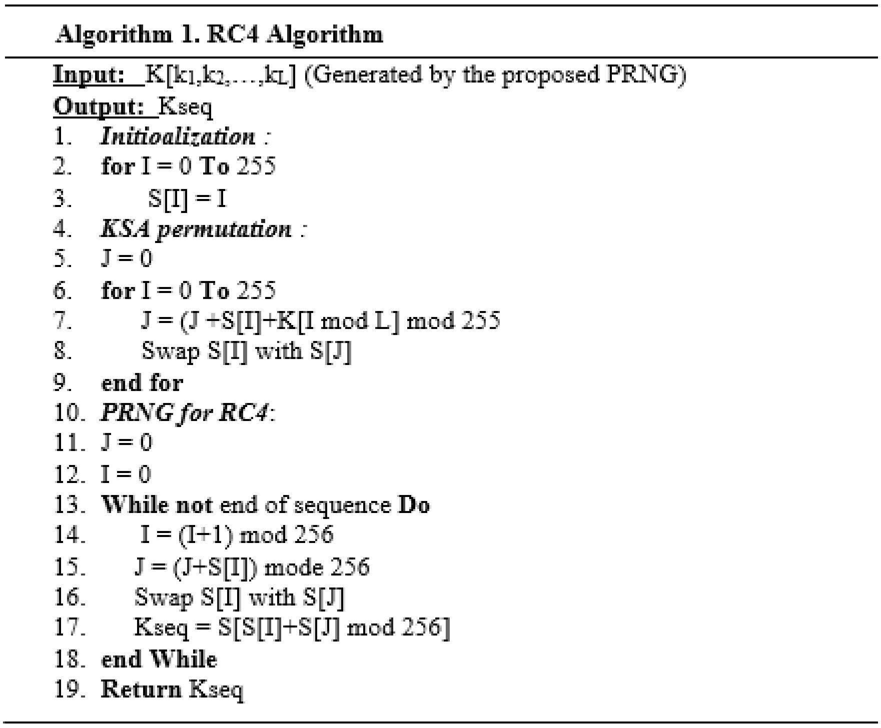
RC4 is considered an acronym regarding to “Rivest Cipher 4”, Ron Rivest developed this algorithm in the year 1987. Furthermore, it has been referred to as “Ron’s Code 4” [26]. Because of its simplicity, speed and ease of generating PRNG sequence the algorithm can be used effectively in both software and hardware.RC4 is majorly applied in various internet protocols like WEP, Skype, WPA, as well as SSL/TLS for file encryption, confidentiality purpose, and communication security [27].

This algorithm has variable key length that is in the range of (0 and 255) bytes for initializing the 256-byte array during the initial state. As suggested by [28, 29] RC4 should utilize key, which is longer than 128 bytes. RC4 Key has been initialized through Key Scheduling Algorithm (KSA), whereas Pseudo Random Number has been generated through PRGA as shown in Algorithm 1.

3.3. Fisher-yates shuffle algorithm

According to literatures [30, 31], the Fisher-Yates shuffling algorithm is for generating a random key sequence that is used for permuting finite set elements. The most important properties of this mechanism are:

- Producing unbiased results, which means every permutation of the set is equally likely.
- It is efficient because it shuffles the element in place, so no additional storage is required, and the execution time depends on the number of elements to be shuffled.



Algorithm 1. RC4 Algorithm

```
Input: K[k1,k2,…,kL] (Generated by the proposed PRNG)
Output: Kseq
Initioalization :
for I = 0 To 255
    S[I] = I
KSA permutation :
J = 0
for I = 0 To 255
    J = (J +S[I]+K[I mod L] mod 255
    Swap S[I] with S[J]
end for
PRNG for RC4:
J = 0
I = 0
While not end of sequence Do
    I = (I+1) mod 256
    J = (J+S[I]) mode 256
    Swap S[I] with S[J]
    Kseq = S[S[I]+S[J] mod 256]
end While
Return Kseq
```

4.The Suggested Colour Image Encryption/Decryption Scheme

This section explains the main stages of the image encrypting process, which consists of PRNG generation and 2 rounds regarding confusion as well as diffusion. Figure 1 shows the proposed encryption/decryption scheme.

The suggested approach has been divided into 3 stages. The first stage represents the key generation: Based on the three components of the coloured images (Red, Green, and Blue). Eight initial conditions and parameters are used as input values to the PCLM generator. This generator produces six random key sequences with size M × N utilized to scramble the positions of the image pixels in the confusion stage at each round and as an input key for the RC4 key generator.

The RC4 generator produces another six key sequences utilized to change pixel values in diffusion stage at each round operation. The second stage is the first round of encryption: Each image component as well as first 3 chaotic key sequences are converted from two-dimension matrices into one dimension by scanning the matrices row by row and apply the Fisher-Yates Shuffle algorithm to achieve the confusion process.

After reconverting the shuffled matrices into two-dimension a complex XOR is performed between the first three RC4 key sequences and the shuffled matrices.

The last stage is the second round of encryption: The output matrices from the first round and the second three chaotic key sequences are converted into one dimension by scanning each component matrix column by column and the rest is the same as the first round. The next section explains these stages in detail.

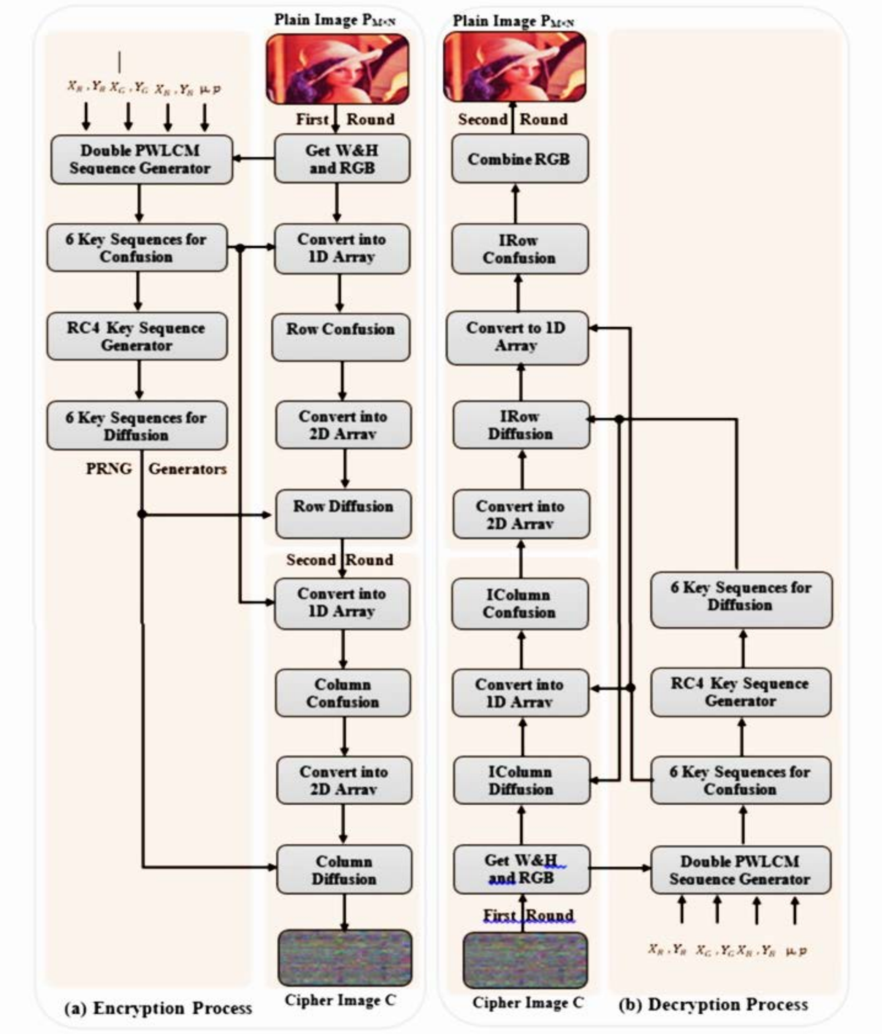


Fig. 1. The proposed cryptosystem.

4.1. The proposed PRNG

Figure 2 shows the generalized block diagram related to proposed PRNG, which is based on piecewise map chaotic method.

4.1.1. PRNG generator based on piecewise map method

The proposed algorithm in this work has two main stages: initialization and chaotic numbers generating. These stages described as follow:

Stage 1: ***Initialization***

This stage consists of four steps as follows:

Step 1: Input initial values for the first piecewise map X_0 in the range [0, 1], and ρ in the range [0, 0.5].

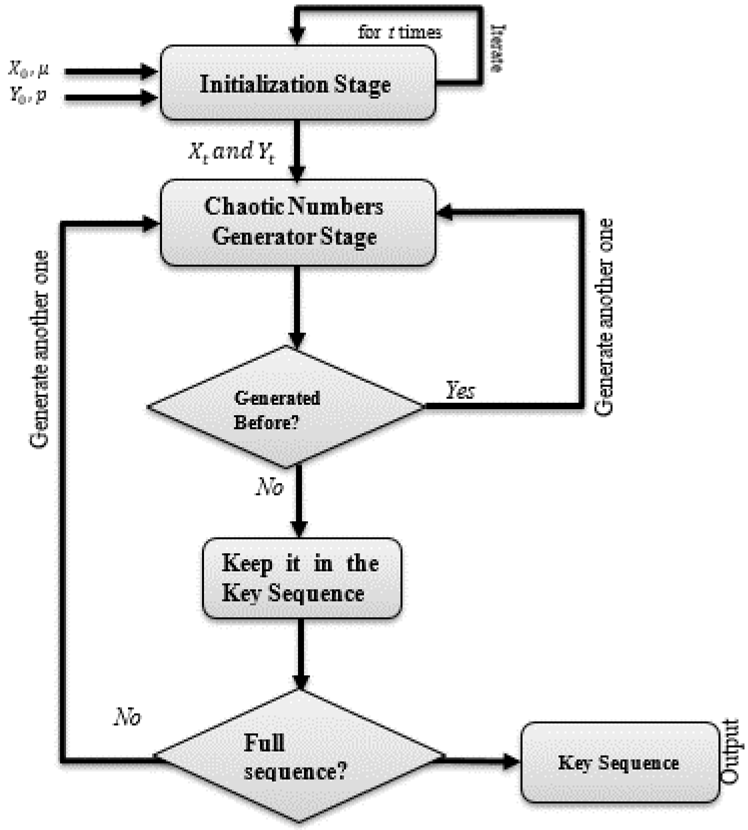


Fig. 2. The generalized block diagram of proposed PRNG.

Step 2: Input initial values for the second piecewise map Y_0 in the range [0, 1], and μ in the range [0, 0.5].

Step 3: Iterate both chaotic maps for *t* times (such as 1000) for eliminating the transient effects.

Step 4: The output X_t and Y_t will be the input value to the next stage for both chaotic maps.

Stage 2: ***Chaotic numbers generator***

This stage contains two phases of chaotic systems. The first phase is composed of two piecewise logistic maps for generating a sequence, which will be used as an initial for each iteration between the phases. In the second phase, the values generated by the first phase are XORed. Figure 3 shows the main structure of this stage.

The output of each map is a real number, which is converted into an integer in the range [0, N-1], by applying the Eq. (1).

$$A = Round\ (V * (N - 1\ ) \qquad (2)$$

where N represents the number of the pixels in width or height of the image, V represents the generated real value by the chaotic map. The details of the suggested PRNG are shown in Algorithm 2.

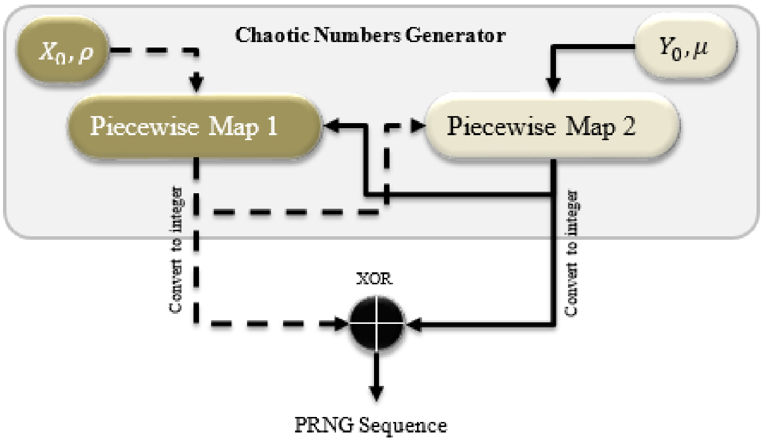
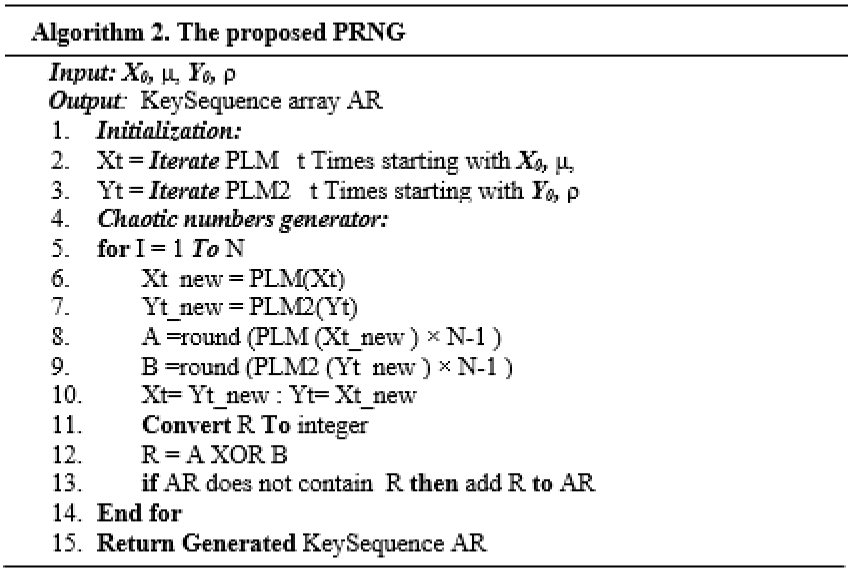


Fig. 3. The structure of PRNG.



```
Algorithm 2. The proposed PRNG
Input: X0, μ, Y0, ρ
Output: KeySequence array AR
Initialization:
Xt = Iterate PLM   t Times starting with X0, μ,
Yt = Iterate PLM2   t Times starting with Y0, ρ
Chaotic numbers generator:
for I = 1 To N
    Xt  new = PLM(Xt)
    Yt_new = PLM2(Yt)
    A =round (PLM (Xt_new ) × N-1 )
    B =round (PLM2 (Yt  new ) × N-1 )
    Xt= Yt_new : Yt= Xt_new
    Convert R To integer
    R = A XOR B
    if AR does not contain  R then add R to AR
End for
Return Generated KeySequence AR
```

4.1.2. PRNG Generator based on RC4 algorithm

Figure 4 shows the structure of RC4 algorithm in which the output key sequence from the previous chaotic algorithm is used as the initial key of length L for RC4 (L is the image's height or the width). The pseudo-code for both RC4 algorithm parts was explained in Algorithm 1. This method generates another key sequence of length *L* to be used in the diffusion process.

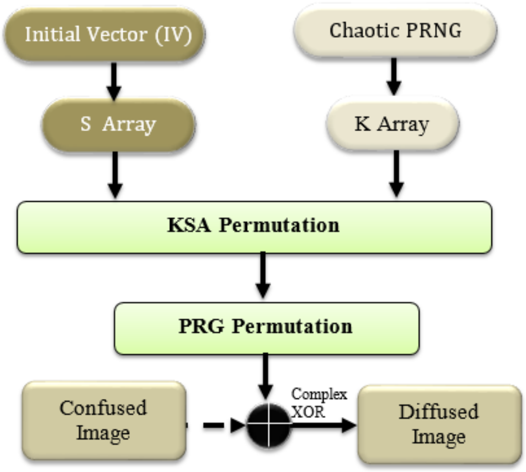


Fig. 4. The structure of RC4 algorithm.

4.2. The encryption process

In this section, the encryption process is presented, which is consists of two rounds. These rounds are explained in detail below:

4.2.1. The first round

This round consists of the following stages:

Stage 1: Input

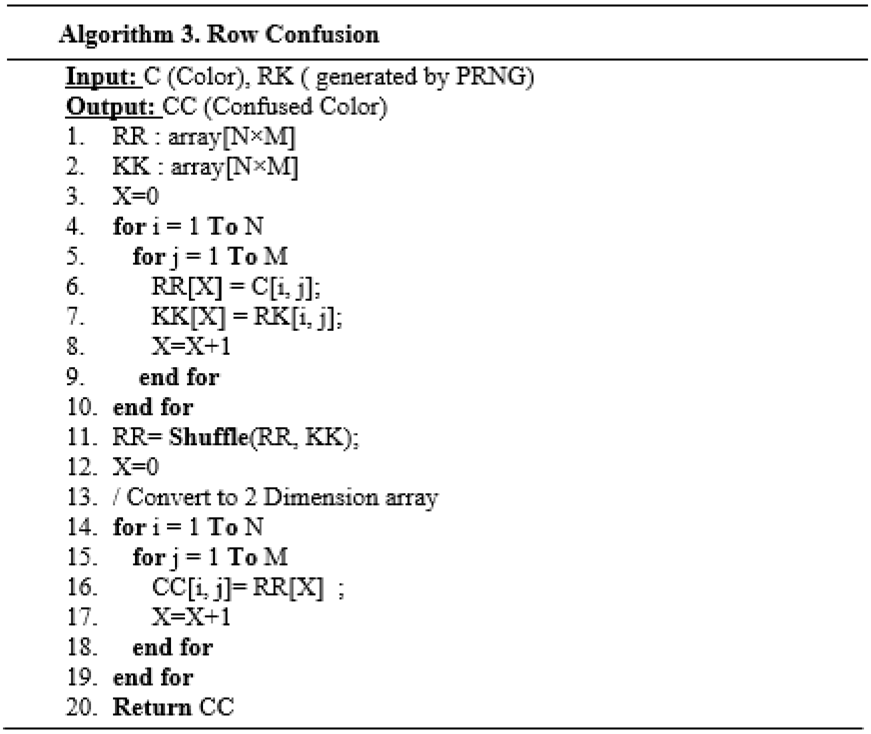
1. Read a colored image as the plain image of size N × M.
2. Get image's width or height.
3. Get the R, G, B components for the plain-image.
4. Read RX, RY, GX, GY, BX, BY, µ and p as the initial values for generating six random sequences of length N by using proposed PRNG. The first three sequences are used for the first round and the second key sequence are used in the second round.
5. Create a new key matrix of size N × M by left shifting each key sequence generated from step 4 according to every key value.
6. Use the key sequences generated from step 4 to generate another six random sequences of length N by using the RC4 algorithm.
7. Create a new key matrix of size N × M by left shifting each key sequence generated from step 6 according to every key value.

Stage 2: Row Confusion Stage

In this stage, all positions of the pixels of the three colors are re-arranged based on the generated random sequences of step 5. The main steps of this stage are as follows:

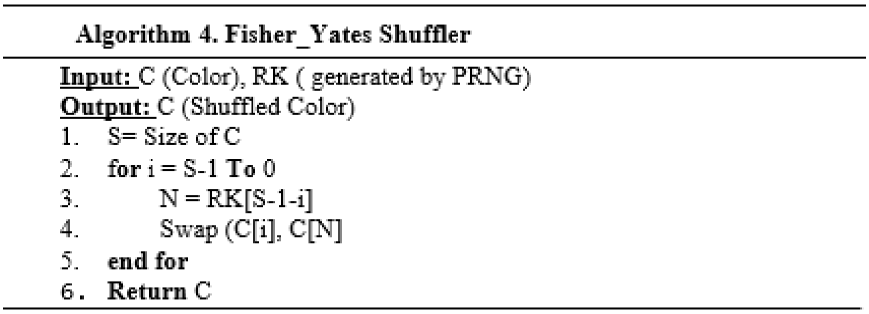
i. Convert both the N × M channels from step 3 and step 5 into 1-D array of N × M size through the scanning of the matrix in a row pattern.
ii. Re-arrange the channels by using Fisher-Yates Shuffling algorithm confusion algorithm and generate three confused colors. Covert the confused one-

dimension array into a two-dimension array again. The confusion and shuffling pseudo code are shown in Algorithms. 3 and 4.



Algorithm 3. Row Confusion

```
Input: C (Color), RK ( generated by PRNG)
Output: CC (Confused Color)
RR : array[N×M]
KK : array[N×M]
X=0
for i = 1 To N
  for j = 1 To M
    RR[X] = C[i, j];
    KK[X] = RK[i, j];
    X=X+1
  end for
end for
RR= Shuffle(RR, KK);
X=0
/ Convert to 2 Dimension array
for i = 1 To N
  for j = 1 To M
    CC[i, j]= RR[X]  ;
    X=X+1
  end for
end for
Return CC
```



Algorithm 4. Fisher_Yates Shuffler

```
Input: C (Color), RK ( generated by PRNG)
Output: C (Shuffled Color)
S= Size of C
for i = S-1 To 0
      N = RK[S-1-i]
      Swap (C[i], C[N])
end for
Return C
```

Stage 3: Diffusion

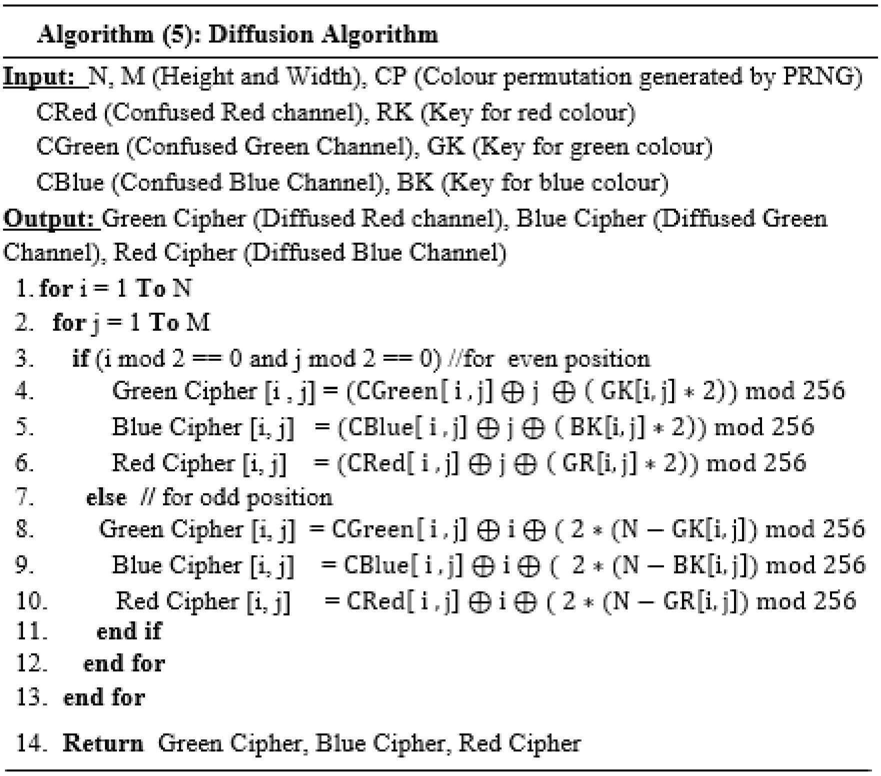
In this stage, a complex XOR operation is implemented between the confused matrices generated by the previous stage and the key sequence generated by the RC4 algorithm. The equation used for even positions is different from odd positions as shown in Algorithm 5.

4.2.2. The Second Round

This round consists of the same stages of the first round except for some point that will be explained hereafter:

Stage 1: Input

It begins from step 5 of round one in which the second PRNG key sequence is used for confusing the image and the rest steps are the same.



Algorithm (5): Diffusion Algorithm

Input: N, M (Height and Width), CP (Colour permutation generated by PRNG)
CRed (Confused Red channel), RK (Key for red colour)
CGreen (Confused Green Channel), GK (Key for green colour)
CBlue (Confused Blue Channel), BK (Key for blue colour)

Output: Green Cipher (Diffused Red channel), Blue Cipher (Diffused Green Channel), Red Cipher (Diffused Blue Channel)

```
for i = 1 To N
 for j = 1 To M
   if (i mod 2 == 0 and j mod 2 == 0) //for even position
      Green Cipher [i , j] = $(CGreen[i, j] \oplus j \oplus (GK[i, j] * 2))$ mod 256
      Blue Cipher [i, j] = $(CBlue[i, j] \oplus j \oplus (BK[i, j] * 2))$ mod 256
      Red Cipher [i, j] = $(CRed[i, j] \oplus j \oplus (GR[i, j] * 2))$ mod 256
   else // for odd position
      Green Cipher [i, j] = $CGreen[i, j] \oplus i \oplus (2 * (N - GK[i, j])$ mod 256
      Blue Cipher [i, j] = $CBlue[i, j] \oplus i \oplus (2 * (N - BK[i, j])$ mod 256
      Red Cipher [i, j] = $CRed[i, j] \oplus i \oplus (2 * (N - GR[i, j])$ mod 256
   end if
 end for
end for
Return Green Cipher, Blue Cipher, Red Cipher
```

Stage 2: Columns Confusion Stage

It contains the same steps of the input stage in the first round only the diffused image generated from the first round is scanned by columns and the rest is the same.

Stage 3: Diffusion

This stage performs the same XOR operation on the confused image and produces the ciphered image.

4.3. The decryption process

Includes the generation of the PRNG key streams with the use of the proposed method and apply the two rounds, but in reverse order starting with the last stage in the second round and end with the row confusion stage to obtain the decrypted image.

5. Experimental Results and Analyses

In the present section, the experimental tests and results are provided, for the sake of showing the strength of the suggested algorithm of image encryption in the face of the majority of the common attack types. These tests include key-space analysis, sensitivity analysis, information entropy, PSNR, and statistical analysis that includes histogram analysis, correlation coefficient. The testing images have been selected from USC-SIPI image database (sipi.usc.edu / database.php), which are Lena, House, and Peppers of size 512×512 colour- image. The tests are carried out using the MATLAB (R2018b) Soft-ware on a computer having system configuration 4-GB RAM with 2.30 GHz processor in windows 10 operating system. The programming language used to program the proposed algorithm was C# based on Visual Studio 2017 version 15.9.3. The initial key parameters are explained in Table 1, which are used in the following tests.

Table 1. Parameters values for the experiments.

Parameter	Symbol	Value
Control Parameter “PCLM1”	μ	0.3
Control Parameter “PCLM2”	P	0.4
Initial Value for R “PCLM1”	X_R	0.856856
Initial Value for G “PCLM1”	X_G	0.965965
Initial Value for B “PCLM1”	X_B	0.745745
Initial Value for R “PCLM2”	Y_R	0.123123
Initial Value for G “PCLM2”	Y_G	0.456456
Initial Value for B “PCLM2”	Y_B	0.789789

5.1. Key space analysis

Based on the principle of Kerckhoff, the level of the security of a method of image encryption is dependent upon the random the encryption keys are. A key space size must contain more than 2^{100} possible keys. Attackers can execute a brute force attack if the key-space is too small [32]. Nevertheless, the sequence generated by the proposed system is dependent upon the initial states, in addition to control parameters [33]. The encryption algorithm with the PRNG is applied in C#. Based on the floating-point standard of the IEEE [3], the 64-bit double-precision number’s computational precision is approximately 10^{-15}. For the PRNG method, there are two initial values $X_0 \in [0,1]$ and $Y_0 \in [0,1]$ for two chaotic maps for each channel, and two control parameters $\mu \in [0,0.5], P \in [0,0.5]$. Therefore, the key-space for the presented scheme may be calculated using the following equations:

$$\left(PV(X_0) \times PV(Y_0)\right) * 3 \times PV(\mu) \times PV(P) \tag{3}$$

$$(10^{15})^8 = 10^{120} \cong 2^{400}$$

where PV represents the number of possible values for a specific variable. The calculation above proofed that the proposed PRNG has enough size of key-space as compared with required.

5.2. Analysis of sensitivity

The effective image encryption algorithm has to be sensitive towards the plain-image, and secrete key such that a minor change in the plain-image or the initial key parameters causes a significant changing in the ciphered-image [10]. In this test, two important measures are utilized for the evaluation of the image encryption algorithms’ strength against the differential attacks. The first one is the rate/number of pixel changes (NPCR), which is used for measuring the percentage of different pixels between the images and the second measure is Unified Averaged Changed Intensity (UACI), for measuring the average intensity differences between the images, they can be calculated as follows [19]:

$$NPCR_{R,G,B} = \frac{\sum_{ij} D_{R,G,B}(i,j)}{W \times H} \times 100\% \tag{4}$$

where $D(i, j)$ is defined as:

$$D(i,j) = \begin{cases} 0, & if\ (C_1(i,j) = C_2(i,j) \\ 1, & if\ (C_1(i,j) \neq C_2(i,j) \end{cases} \tag{5}$$

$$UACI_{R,G,B} = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_{R,G,B}(i,j) - \bar{C}_{R,G,B}(i,j)|}{512}\right] \times 100\% \tag{6}$$

For two arbitrary images having the same size, the ideal values of UACI and NPCR and must be around 33.4635%, and 99.6094% respectively [9].

5.2.1. Key sensitivity

An algorithm of generating PRNG should be considered sensitive to the smallest changes in the generated keys. To test the proposed PRNG method for sensitivity, the following test cases have been used to verify the sensitivity:

Case1: Both X_0 and Y_0 equal 0.123456789

Case2: X_0 changes from 0.123456789 to $X_0 + 2^{-48}$

Case3: Y_0 changes from 0.123456789 to $Y_0 + 2^{-48}$

Case 4: X_0 and $Y_0$are changed from 0123456789 to $X_0 + 2^{-48}$ and $Y_0 + 2^{-48}$.

The steps below assess the key sensitivity:

Step1: Encrypt the original image with key parameter value in Table 1 *K₁* for producing an encrypted image named *C₁*.

Step2: Produce another key, i.e., *K₂* from the above cases then cipher the original image with the use of the *K₂* for producing the encrypted image *C₂*.

Step3: Compare *C₁* and *C₂* to denote the number of differed pixels by applying Eqs. (4) and (6).

The key sensitivity results for the three cases are presented in Table 2. As can be seen, any small change in the initial values of the key sequence can influence the encryption and decryption process and the computed NPCR and UACI values are almost equal to their theoretical values, which means the fact that the suggested system has a highly sensitive strong key.

Table 2. The calculated key sensitivity for of Lena image.

Image	NPCR				UACI			
	R	G	B	Avg.	R	G	B	Avg.
Case 1	99.616	99.619	99.596	99.610	33.358	33.461	33.406	33.408
Case 2	99.606	99.625	99.601	99.611	33.515	33.527	33.438	33.493
Case 3	99.618	99.599	99.611	99.609	33.493	33.402	33.440	33.445

5.2.2. The analysis of the plain-text sensitivity

Assuming the fact that the original images *P₁* and *P₂* differ in one pixel only. They are encrypted through using the suggested approach with an identical key K for obtaining two encrypted images, which can be represented as *C₁* and *C₂*.

Two plain images P_1 and P_2 (the original plain-image and the image which have been produced by changing [15, 210] R channel's pixel value from '175 to '76) were used to perform these tests. These images have been ciphered using an identical key for generating their respective ciphered images (C_1 and C_2) after two cycles. The test results were obtained via simulation studies (as shown in Table 3.). The NPCR$_{R,G,B}$ was found to be more than 99%, while the UACI$_{R,G,B}$ was over 33%. These results present a high sensitivity of the suggested approach based upon the small changes in the plain-image.

Table 3. NPCR and UACI of Lena and house images with 1 bit different.

Image	NPCR				UACI			
	R	**G**	**B**	**Avg.**	**R**	**G**	**B**	**Avg.**
Lena	99.623	99.621	99.612	99.618	33.474	33.485	33.468	33.475
House	99.619	99.617	99.614	99.616	33.412	33.420	33.418	33.416

5.3. The analysis of information entropy

Entropy is defined as the measurement of uncertainty and may be utilized to portray the level of uncertainties in image information [20]. The colour-level distribution values in an image can also be determined via entropy analysis. If there is more uniformity in the distribution of the grey-level values, there will be a greater entropy, and a higher entropy value portrays better-secured encryption. Entropy is calculated thus [33]:

$$H\ (S) = \sum_{i=0}^{2^M-1} P(S_i)\ log_2 \frac{1}{P(S_i)} \tag{7}$$

where, $P(S_i)$ is the probability of S_i, while 2^M is the state of the total source of information. An image is truly random (*RI*) if its pixel intensities are uniform in the range of [0, 255], i.e., $P(RI)$= 1/256 for all $i \in$ [0, 255] and $H(RI)$= 8 bits. This means that an ideally random image has an entropy information value of 8.

In this study, we computed the entropy information values of the three standard plain-images and their respective cipher images, and the results of the computation are presented in Table 4. The obtained entropy values were close to the theoretical entropy value of H = 8, which suggests the fact that the results of the approach of the encryption in random like cipher-images.

Table 4. Information entropy of the proposed method.

Image	Plain Image				Ciphered Image			
	R	**G**	**B**	**Avg.**	**R**	**G**	**B**	**Avg.**
Lena	7.2531	7.594	6.9684	7.2718	7.9993	7.9994	7.9992	7.9993
House	7.3843	7.1634	6.8060	7.1179	7.9993	7.9993	7.9993	7.9993
Pepper	6.7178	6.7990	6.2138	6.5768	7.9991	7.9992	7.9991	7.9991

5.4. Statistical analysis

The statistical analysis may be obtained by calculating the histograms and relations between adjacent pixels as will explained in the following next section.

5.5. Histogram analysis

The pixel distribution is depicted as the image, which has been generated by drawing the number of the pixel at every level of colour intensity in the histogram [22]. The Lena and house images are shown in Figs. 4 and 5, respectively. The channel illustrations are shown in red, green, and blue for the plain and ciphered images. The illustrations of the ciphered images have been clearly uniform and very distinctive from their corresponding standard image graphs, thus giving no idea of a possible statistical attack on the cryptographic system.

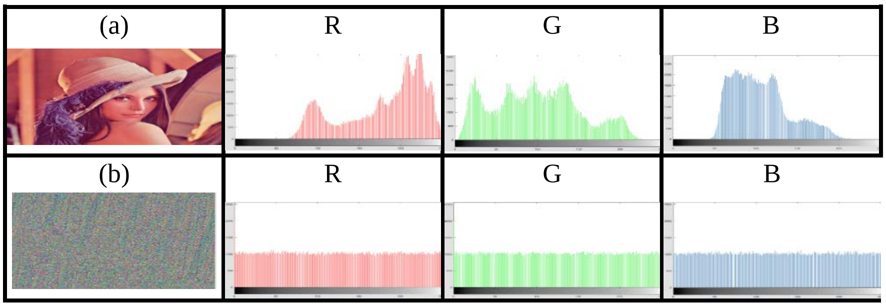


Fig. 4. (a) Lena image and the R, G and B component histograms; (b) The corresponding cipher image of (a) and the R, G and B component histograms.

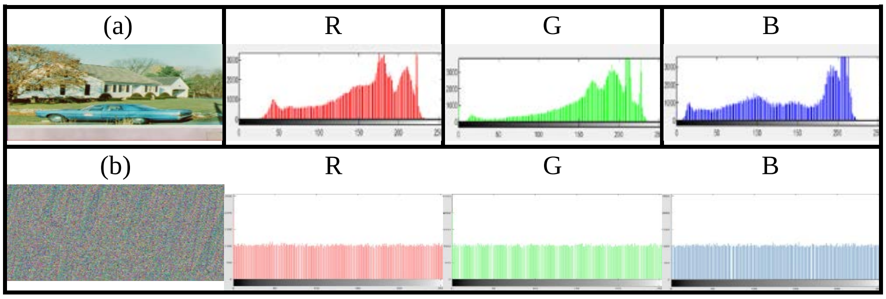


Fig. 5. (a) House image and the R, G and B component histograms; (b) The corresponding cipher image of (a) and the R, G and B component histograms.

The analysis of the correlation

The correlation between two adjacent elements can be determined in three ways, either by looking at vertically contiguous pixels, taking into account horizontal pixels, or looking at diagonal pixels in the encrypted image [10,11]. In this study, the link study was performed using 5,000 pairs of adjacent pixels. The results of the links obtained have been listed in Table 5. The coefficients of the correlation for Lena and house images were calculated as follows [23]:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D_x}\sqrt{D_y}} \tag{8}$$

$$cov(x,y) = E[(x - E(x))(y - E(y))] \tag{9}$$

Here, $E(x)$ and $E(y)$ are the projection and difference of variable x,

$$E(x) = \frac{1}{L}\sum_{i=1}^{L} x_i\ ;\quad D_x = \frac{1}{L}\sum_{i=1}^{L}(x_i - E(x))^2 \tag{10}$$

where x and y represent neighbouring pixel values, while L is the number of the used samples. The value of r_{xy} must lies between -1 and 1. If it is close to 1, this indicating strong positive-correlation, and if it is close to -1, this indicating strong negative-correlation, but if it close to 0, this mean the image is totally uncorrelated and the encryption algorithm is strong enough to resist statistical effects [34]. After running the correlation algorithm on the plain and cipher image in many directions we can see the plain image nears (+1), that means the pixels have a high correlation and in cipher image close form (0) that main the pixels having low correlation between them.

Table 5. The coefficients of correlation of two neighbouring pixels.

Direction	Image	Colour	Original	Cipher
Vertical	Lena	R	0.9908	-0.0002
		G	0.9828	0.0018
		B	0.9581	0.0009
	House	R	0.9828	0.0011
		G	0.9843	0.0021
		B	0.9850	-0.0015
Horizontal	Lena	R	0.9788	0.0115
		G	0.9696	-0.0030
		B	0.9316	-0.0018
	House	R	0.9842	0.0312
		G	0.9893	-0.0048
		B	0.9898	0.0023
Diagonal	Lena	R	0.9680	-0.0003
		G	0.9617	0.0109
		B	0.9218	0.0121
	House	R	0.9708	-0.0119
		G	0.9768	-0.0012
		B	0.9771	-0.0054

5.6. Speed analysis

The speed of the algorithm and the execution time depends mainly on several factors, like a programming language, operating system, hardware specifications, and programming skills. Thus, it is useless to compare the proposed algorithm against two or more encryption algorithms, unless using the same environment. According to the mentioned specification, the speed of the whole process for encrypting Lena's image of size 512*512 was 1.774447 seconds. This speed can be faster by optimizing the hardware and the software of the computer. However, for such speed, we can clearly state that the suggested algorithm is still suited for internet applications where the time of encryption/decryption processes must be short compared to the transmission time.

5.7. Complexity analysis

In our scheme, the time complexity consists of six phases as shown in Table 6. The total time complexity of whole algorithm is:

$$2(O(N)) + 2(O(N)) + O\,(N \times M) + O\,(N \times M) + O\,(N \times M) + O\,(N \times M) \tag{11}$$

where N and M represent the height and the width of the encrypted image. The results of complexity analysis in comparison with other approaches have been listed in Table 10.

Table 6. Time complexity of the proposed algorithm.

No.	Phases	Time complexity
1	**Proposed chaotic PRNG**	$2(O(N))$
2	**RC4 key generation**	$2(O(N))$
3	**Row confusion**	$O(N \times M)$
4	**Row diffusion**	$O(N \times M)$
5	**Column confusion**	$O(N \times M)$
6	**Column diffusion**	$O(N \times M)$

5.8. The analysis of the peak signal to noise ratio (PSNR)

The target encryption method's evaluation may be carried out through the calculation of PSNR. In this analysis, the plain-image can be referred to as the signal, while the encrypted one can be referred to as the noise [35]. The calculation of the PSNR is performed based on [33]:

$$PNSR = 20 \times \log_{10}\left(\frac{255}{\sqrt{MSE}}\right)\ dB \tag{12}$$

where the *MSE* stands for the mean square error and is obtained based on:

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[f(i,j)-f\prime(i,j)]^2}{MN} \tag{13}$$

where N and M represent the height and the width of the image. $f(i, j)$ is the $(i, j)^{th}$ pixel value of the plain-image and $f\ '(i\ ,\ j)$ represents $(i\ ,\ j)^{th}$ encrypted image's pixel value. The low value of PSNR means a larger difference between the plain and the encrypted image. The PSNR values of various images are listed in Table 7, which indicates a good encryption quality as the PSNR values are < 10 dB. The PSNR results comparison against other approaches has been listed in Table 11.

Table 7. PSNR value of various images.

Image	PSNR(*dB*) Space
Lena	8.6675
Pepper	8.1478
House	7.6078

5.9. Randomness test

The generated sequences by the proposed PRNG are tested using the National Institute of Standards and Technology (NIST SP800-22) Test Suite. It is a statistical package consists of 15 tests for measuring the randomness of the output sequences of true-random number generators or pseudo-random number generators [36].

The p-value represents the possibility of a good or poor random number generator. The Testing method compares the p-value to 0.01. If the p-value is greater than 0.01, then the sequence is passed the test and is considered as random, otherwise the sequence is rejected because of non-randomness [37]. The proposed PRNG was evaluated to produce 1500 different key sequences, each of 1,000,000 bits, and then measured the average of the p-values resulting from these tests. All the tests have been successful as the p-value ≥ 0.01 as shown in Table 8, and the sequences generated can be considered random, spread uniformly, and appropriate for cryptography.

Table 8. NIST SP800-22 standard test of the proposed PRNG.

Test No.	**Statistical Test Name**	**Proposed PRNG**	
		p-value	**Conclusion**
1	Approximate entropy	0.499251	SUCCESS
2	Block frequency	0.835226	SUCCESS
3	Cumulative sums (Forward)	0.999278	SUCCESS
4	FFT	0.509529	SUCCESS
5	Frequency	0.707660	SUCCESS
6	Linear complexity	0.275249	SUCCESS
7	Longest Runs	0.603742	SUCCESS
8	Non-Overlapping templates	0.250249	SUCCESS
9	Overlapping template	0.693722	SUCCESS
10	Random excursions	0.055386	SUCCESS
11	Random excursions variant	0.020319	SUCCESS
12	Rank	0.091558	SUCCESS
13	Runs	0.902259	SUCCESS
14	Serial	0.707660	SUCCESS
15	Universal statistical	0.462343	SUCCESS

5.10. Performance comparison of image encryption approaches

The suggested algorithm has been compared against other methods of colour image encryption in literature. Table 9 presents a comparison of suggested algorithm against other algorithms in terms of their Key Space, Entropy, NPCR, UACI and correlation coefficient for Lena image, Table 10 shows the time complexity comparison and Table 11 states the PSNR comparison. The results have shown that the proposed algorithm has better security performance compared to the others.

Table 9. Comparison of the performance between varieties of approaches of image encryption for Lena image.

Performance Parameter	**[10]**	**[18]**	**[19]**	**[20]**	**[23]**	**Proposed Algorithm**
Entropy	7.99727	7.9895	7.9962	7.9992	7.9972	7.9993
NCPR	99.616	99.7915	99.57	99.610	99.4207	99.618
UACI	33.465	49.2191	33.39	33.400	33.3457	33.475
H. Correlation	-0.0003	-0.0036	-0.0012	0.0607	0.0019	0.0002
V. Correlation	0.0012	0.0001	0.0022	-0.0011	0.0003	-0.0012
D. Correlation	0.0052	-0.023	-0.0022	-0.0057	0.0033	-0.0023
Key Space	2^{384}	2^{280}	$>2^{128}$	$>2^{100}$	2^{320}	2^{400}

Table 10. Time complexity comparison.

Algorithm	Time complexity
Proposed	4(O(N))+4(O(N×M
[38]	100(O(N×M)
[39]	12(O(N×M))
[40]	9(O(N×M)

Table 11. PSNR comparison for Lena.

Algorithm	PSNR (*dB*)
Proposed	8.6675
[10]	8.924724
[22]	8.6728
[33]	9.2335
[35]	11.30

6. Conclusions and Future Work

In the present paper, a new scheme of image encryption is proposed based on a double PWLM and RC4 stream ciphering algorithm. It depends on two encryption rounds. In each round, the original image is confused with the use of the PRNG key sequence generated by the double PCWLM algorithm and diffused using a PRNG key sequence generated by the RC4 algorithm. The confusion operation is accomplished by using the efficient and fast Fisher-Yates technique while the diffusion process is done using complex XOR operation.

The experimental and simulation analysis of the proposed scheme proved that this scheme is known for its wanted characteristics like the high level of sensitivity for the small changes in the original image (through the values of UACI and NPCR), low correlation coefficients, low value of the PSNR, and the high entropy of the information.

Furthermore, the proposed scheme presents more security comparing other algorithms mentioned in the literature with acceptable running time. The test images were selected from the image data-base of the USC -SIPI. All those characteristics showed the fact that the suggested scheme is secure, efficient, fast for the encryption of images, and highly robust against cryptanalysis threats such as brute force, cipher-text, plain-text, statistical and differential attacks. Moreover, the generated output sequences of the proposed have been tested by NIST SP800-22 suite; the test results showed that the sequence has perfect statistics performance.

This scheme is suitable to be used in real time image encryption and transmission in web applications. In future, this algorithm can modify by adding compression or scaling techniques to speed up the encryption process and can be modified to be used for video encryption.

Nomenclatures

Greek Symbols

$\mathcal{P}$	Control parameter
μ	Control parameter

Abbreviations

AES	Advanced Encryption Standard
CCPS	Chaotic Circular Pixel Shuffling
CDCS	Composite Discrete Chaotic System
2D-STCM	2D-Sine Tent Composite Map

DES	Data Encryption Standard
IDEA	International Data Encryption Algorithm
KSA	Key Scheduling Algorithm
NIST	National Institute of Standards and Technology
NPCR	Number of Pixel Changes Rate
PRNG	Pseudo Random-Number-Generator
PSNR	Peak Signal to Noise Ratio
PWLCM	Piece Wise Linear Chaotic Map
RC4	Rivest Cipher 4
SSL	Secure Sockets Layer
TSL	Transport Layer Security
UACI	Unified Averaged Changed Intensity
WPA	Wi-Fi Protected Access

References

1. Kumar, M.; Aggarwal, A.; and Garg, A. (2014). A review on various digital image encryption techniques and security criteria. *International Journal of Computer Applications*, 96(13), 19-26.
2. Radwan, A.G.; AbdelHaleem, S.H.; and Abd-El-Hafiz, S.K. (2016). Symmetric encryption algorithms using chaotic and non-chaotic generators: A review. *Journal of Advanced Research*, 7(2), 193-208.
3. Chen, J.; Zhu, Z.; Fu, C.; Yu, H.; and Zhang, L. (2015). A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism. *Communications in Nonlinear Science and Numerical Simulation*, 20(3), 846-860.
4. Zhu, H.; Zhao, C.; and Zhang, X. (2013). A novel image encryption-compression scheme using hyper-chaos and Chinese remainder theorem. *Signal Processing: Image Communication*, 28(6), 670-680.
5. Wang, Y.; Wong, K.W.; Liao, X.; and Chen, G. (2011). A new chaos-based fast image encryption algorithm. *Applied Soft Computing* , 11(1), 514-522.
6. Zhu, H.; Zhao, C.; Zhang, X.; and Yang, L. (2014). An image encryption scheme using generalized Arnold map and affine cipher. *Optik*, 125(22), 6672-6677.
7. Zhang, L.Y.; Li, C.; Wong, K.W.; Shu, S.; and Chen, G. (2012). Cryptanalyzing a chaos-based image encryption algorithm using alternate structure. *Journal of Systems and Software*, 85(9), 2077-2085.
8. Som, S.; and Sen, S. (2013). A non-adaptive partial encryption of grayscale images based on Chaos. *Procedia Technology*, 10(2013), 663-671.
9. Zhang, Y.; and Tang, Y. (2018). A plaintext-related image encryption algorithm based on chaos. *Multimedia Tools and Application*, 77(2), 6647-6669.
10. Hanchinamani, G.; and Kulkarni, L. (2015). An efficient image encryption scheme based on a Peter De Jong chaotic map and a RC4 stream cipher. *3D Research*, 6(3), 1-15.
11. François, M.; Grosges, T.; Barchiesi, D.; and Erra, R. (2014). Pseudo-random number generator based on mixing of three chaotic maps. *Communications in Nonlinear Science and Numerical Simulation*, 19(4), 887-895.
12. Zhen, P.; Zhao, G.; Min, L.; and Li, X. (2014). Optimized key agreement protocol based on chaotic maps. *Journal of Communications*, 9(5), 398-403.

13. Herbadji, D.; Belmeguenai, A.; Derouiche, N.; Youcef, Z.; and Ouchtati, S. (2019). A novel color image encryption scheme using logistic map and quadratic map systems. *International Conference on Mobile, Secure, and Programmable Networking*. Paris, France, 13-23.
14. Xing, F.Z.; Cambria, E.; and Zou, X. (2017). Predicting evolving chaotic time series with fuzzy neural networks. *2017 International Joint Conference on Neural Networks(IJCNN)*. Anchorage, USA, 3176-3183.
15. Gholizade-Narm, H.; and Shafiee Chafi, M.R. (2015). Using repetitive fuzzy method for chaotic time series prediction. *Journal of Intelligent and Fuzzy Systems*, 28(4), 1937-1946.
16. Akhshani, A.; Akhavan, A.; Mobaraki, A.; Lim, S.C.; and Hassan, Z. (2014). Pseudo random number generator based on quantum chaotic map. *Communications in Nonlinear Science and Numerical Simulation*, 19(1), 101-111.
17. Hameed, S.M.; and Mahmood, I.N. (2018). A modified key scheduling algorithm for RC4. *Iraqi Journal of Science*, 57(1), 262-267
18. Wu, X.; Li, Y.; and Kurths, J. (2015). A new color image encryption scheme using CML and a fractional-order chaotic system. *Plos One*, 10(3), 1-28.
19. Wang, X.Y.; Zhang, Y.Q.; and Bao, X.M. (2015). A colour image encryption scheme using permutation-substitution based on chaos. *Entropy*, 17(6), 3877-3897.
20. Zhu, H.; Zhang, X.; Yu, H.; Zhao, C.; and Zhu, Z. (2016). A novel image encryption scheme using the composite discrete chaotic system. *Entropy*, 18(8), 1-27.
21. Abbas, M.; and Mohammed, D. (2017). Image encryption technique based on the entropy value of a random block. *International Journal of Advanced Computer Science and Applications*, 8(7), 260-266.
22. Sahari, M.L.; and Boukemara, I. (2018) A pseudo-random numbers generator based on a novel 3D chaotic map with an application to color image encryption. *Nonlinear Dynamics*, 94(1), 723-744.
23. Agarwal, S. (2018). Secure image transmission using fractal and 2D-chaotic map. *Journal of Imaging*, 4(1), 1-17.
24. Wang, X.; and Sun, H. (2020). A chaotic image encryption algorithm based on improved Joseph traversal and cyclic shift function. *Optic and Laser Technology*, 122(7), 1-12.
25. Hu, Y.; Zhu, C.; and Wang, Z. (2014). An improved piecewise linear chaotic map based image encryption algorithm. *The Scientific World Journal*. 2014, 1-7.
26. Sahib, N.M.; Fadel, A.H.; and Ahmed, N.S. (2018). Improved RC4 algorithm based on multi-chaotic maps. *Research Journal of Applied Sciences, Engineering and Technology*, 15(1), 1-6.
27. Crainicu, B. (2015). On Invariance weakness in the KSAm Algorithm. *Procedia Technology*, 19(3), 850-857.
28. Fluhrer, S.; Mantin, I.; and Shamir, A. (2001). Weaknesses in the key scheduling algorithm of RC4. *International Workshop on Selected Areas in Cryptography*. 1-24.
29. Jindal, P.; Singh ,B. (2015). RC4 encryption-A literature survey. *Procedia Computer Science*, 46(1), 697-705.

30. Hazra, T.K.; Ghosh, R.; Kumar, S.; Dutta, S.; and Chakraborty, A.K. (2015). File encryption using fisher-yates shuffle. *International Conference and Workshop on Computing and Communication(IEMCON)*. Vancouver, Canada, 1-7.
31. Saeed, S.; Umar, M.S.; Ali, M.A.; and Ahmad, M. (2014). Fisher-yates chaotic shuffling based image encryption. *International Journal of Information Processing*, 8(3), 31-41.
32. Oğraş, H.; and Türk, M. (2016). A secure chaos-based image cryptosystem with an improved sine key generator. *American Journal of Signal Processing*, 6(3), 67-76.
33. Sheela, S.J.; Suresh, K.V.; and Tandur, D. (2018). Image encryption based on modified Henon map using hybrid chaotic shift transform. *Multimedia Tools and Applications*, 77(19), 25223-25251.
34. Patro, K.A.K.; and Acharya, B. (2019). An efficient colour image encryption scheme based on 1-D chaotic maps. *Journal of Information Security and Appl*ication, 46(6), 23-41.
35. Younas, I.; and Khan, M. (2018). A new efficient digital image encryption based on inverse left almost semi group and Lorenz chaotic system. *Entropy*, 20(12), 1-22.
36. Hammood, M.M.; Yoshigoe, K.; and Sagheer, A.M. (2013). RC4-2S : RC4 stream cipher with two state tables. *Information Technology Convergence*, 4(7), 13-20.
37. Nepomuceno, E.G.; Nardo, L.G.; Arias-Garcia, J.; and Butusov, D.N. (2019). Image encryption based on the pseudo- orbits from 1D chaotic map. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(6), 1-7.
38. Zhang, Y.Q.; and Wang, X.Y. (2015). A new image encryption algorithm based on non-adjacent coupled map lattices. *Applied Soft Computing Journal*, 26(1), 10-20.
39. Wang, X.Y.; Zhang, H.L.; and Bao, X.M. (2016). Color image encryption scheme using CML and DNA sequence operations. *BioSystems*, 144(6), 18-26.
40. Cheng, G.; Wang, C.; and Chen, H. (2019). A novel color image encryption algorithm based on hyperchaotic system and permutation-diffusion architecture. *International Journal of Bifurcation and Chaos*, 29(09),1-17.