Parallelizable cipher of color image based on two-dimensional chaotic system

Sawsen Abdulhadi Mahmood[1], Khalid Ali Hussein[2], Yaseen Naser Jurn[3], Ekhlas Abbas Albahrani[4]
[1,2,4]Al Mustansiriyah University, College of Education, Computer Science Department, Iraq
[3]University of Information Technology and Communications, College of Engineering,
Mobile Communication and Computing engineering Department, Iraq

Article Info

Article history:

Received Aug 1, 2019
Revised Sep 15, 2019
Accepted Oct 17, 2019

Keywords:

Chaotic system
Diffusion
Parallel encryption
Permutation
Security analysis

ABSTRACT

This paper aims to present a parallel implementation-based color image encryption using non -linear chaotic system. The adopted chaotic system was suggested and approved in our previous work which generates key streams with chaotic behavior. In this paper, pixel level permutation algorithm based on chaotic map generation is investigated and analyzed. The encryption-decryption schemes are achieved in parallel and composed of three main phases: chaotic keys generation, pixel-level permutation and bit-level diffusion phase. Both permutation and diffusion processes are achieved according to the chaotic keys. The parallel implementation of the proposed image encryption system is realized and inspired with parallel computing library offered by Matlab 2018, which equips highly performance than the pipeline ones and would be helpful to utilize in image encryption/decryption for real time application. Security and statistical analysis in addition to the main differential attacks analysis are specified to evaluate the performance of the proposed image encryption algorithm with parallel implementation. From the experimental results, the output image of the encryption task shows a higher randomness of the encrypted image which can be effectively resistant to attacker. Furthermore, the run time of encryption process is faster than other research works.

Corresponding Author:

Sawsen Abdulhadi Mahmood,
Department of Computer Science,
Al Mustansiriyah University Baghdad, Iraq.
Email: sawsenhadi@gmail.com

1. INTRODUCTION

Recently, media communication in secure manner has attracted significant interesting as an growing of images, video transmission over the networks, shared in mobile phones with assist of cloud storage space. Different Image encryption methods were adopted to provide the privacy authenticity, integrity and security through transmission [1]. Chaotic systems based cryptosystem have been applied and approved due to the main properties of chaotic systems including: periodicity, sensitivity to initial conditions, dynamics estimation property and structural intricacy considered to get the confusion, diffusion, pseudo number generation exploited in cryptography systems [2-4]. To achieve and meet the challenge of real time transmission of secure image, many research has been suggested to enhance the performance of chaotic-based image encryption methods. A suitable image encryption methods in recent security systems should have a higher efficiency and performance for being applied in real time applications [5]. Meanwhile, parallel computing is a paradigm of computation in which multiple tasks are implemented simultaneously. It works on the context that large problems can be spited into smaller ones, which are then disbanding in parallel manner (simultaneously) [6]. Therefore, parallel computing are realized and adopted in recent encryption methods in order to reduce the consuming run time [7]. This paper address the main advantages of adopting parallel implementation of imag encryption based on two dimensional chaotic system mentioned in our previous work [8]. The rest of this paper is organized as follows; Section 2 introduces the related works. Section 3 presents an overview of the proposed image encryption algorithm including; chaotic map generation, permutation scheme, diffusion scheme and parallelism implementation based encryption scheme. Section 4 presents the results and illustrates the processing of security analysis including; sensitivity to initial conditions, key space analysis, histogram, correlation coefficients, entropy information, differential attacks and speed time. Finally, Section 5 concludes this paper.

2. RELATED WORKS

The chaos encryption algorithm presented for the first time by Matthews in 1989 [9] for the encryption purposes. This algorithm was applied for the image encryption based on 2D chaotic map by one of the most important papers [10]. Since then a lot of researcher presented many chaos algorithms for image encryption.

On the bases of the 2D chaotic maps and reversible integer wavelet transform an efficient scheme for image encryption was presented [11]. The initial values of chaotic maps and different parameters were generated based on the cipher key related plain image. The order of chaotic maps was utilized to permute the plain image. Then, the integer wavelet transform was used to process the plain-image. The orbits of chaotic maps used to diffuse a part of transform coefficient. Finally, the cipher image was obtained by inverse integer wavelet transform based on the diffused coefficient. Another scheme of chaotic image encryption was presented by Moatsum A. et al. in [12] using two hybrid chaotic system composed of tent, logistic and sine maps. The diffusion and confusion processes were specified based on disturbing the chaotic states and control parameters. Both of confusion and diffusion processes exploited the pixel intensities from one half of the plain image to encrypt the second half. Zenggang X. et al. [13] introduced a chaos encryption algorithm for color image. In this work, the combination of Cyclic Redundancy Check (CRC) and nine palace map were adopted to shuffle and encode image pixels respectively. The randomness of chaos such as Henon map and Arnold cat map were utilized by G.Chaitanaya et al. [14] for image encryption. The randomness of chaotic elements were generated by Arnold's cat map and Henon Map are used as key values in order to shuffle the pixel position. These key values were used for encryption and decryption image for internet image encryption. The Xuncai Zhang et al. [15] presents a hyperchaos digital image encryption scheme by applied the bit permutation and dynamic DNA encoding. The diffusion of pixel values and the scrambling transformation of the pixel locations are achieved based on the bit permutation, chaos mapping, and the dynamic DNA encoding technique. Real time applications, the lightweight 1D chaotic algorithm was presented to encrypt grayscale images and implemented on LPC2148 32-bit microcontroller with MCB2140 ARM development board [16]. This chaotic encryption system can be used in different small payload applications for the purpose of real-time embedded systems. This algorithm uses a primary symmetric key with length 128-bit. The pixels of a plain-image were processed in a sequential order during the encryption process. The weakness of this algorithm was restriction the input image size.

Three dimensional chaotic and four dimensional hyper chaotic systems were presented for encryption the color and gray images [17]. These systems were applied based on the key generation which is depending on the initial conditions for encryption and decryption. The proposed systems provide sensitivity to very small change in the initial conditions [18]. The main object of these systems is pixel position permutation only without changing its value. The same key of encryption is used for implementing the decryption process. The 2D chaotic map and information entropy were utilized to present an image encryption algorithm [18]. This algorithm contains permutation, modulation and diffusion (PMD) operations. The generation of key stream had been influenced by the information entropy. This process made the initial keys that used in the permutation and diffusion were interacted with each other. In order to avoid the shortcoming of unchangeable gray distribution before diffusion, the modulation operation is introduced between the permutation and diffusion processes.

3. THE PROPOSED PARALLEL IMAGE ENCRYPTION

A 2D non- linear quadratic equations system based parallel image encryption will be adopted in this work to handle the main aspects such as: chaotic map generation, pixels image permutation using chaotic sequences X and Y, image diffusion using chaotic sequences X and Y. The proposed color image encryption is supposed to be symmetric key encryption based algorithm. To meet the essential requirements of NPCR and UACI measurements (NPCR > 0.995, UACI > 0.334), more than one round are needed for both permutation and diffusion processes. An overview of the proposed encryption scheme is illustrated in Figure 1. In this paper, the plain image is assumed as color image with W×H×3 dimensions.

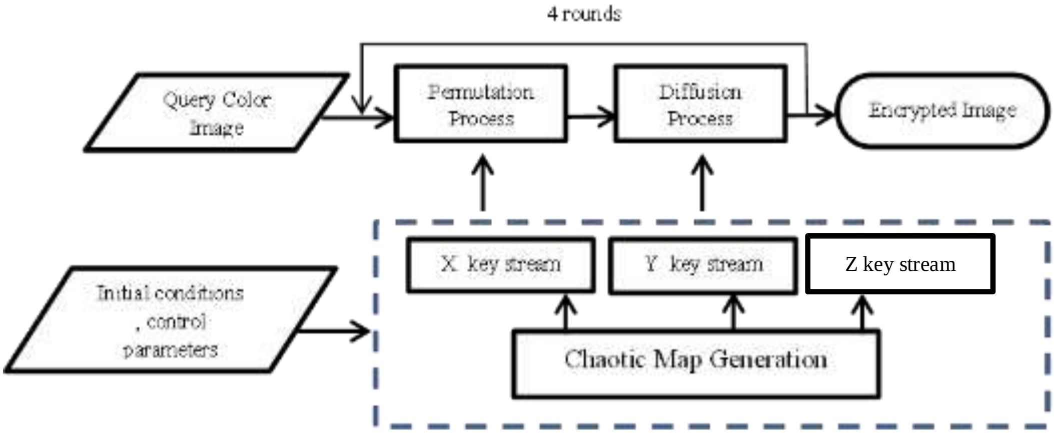


Figure 1. An overview of the proposed image encryption method

3.1. Chaotic Map Generation

The keys stream generation process is achieved based on a new two-dimensional chaotic system characterized by two quadratic nonlinearities equations with two initial values and five control parameters, as given [8]:

$$X_{i+1} = ay_i^2 - bx_i^2 - c \qquad (1)$$
$$Y_{i+1} == dx_i y_i - ex_i$$

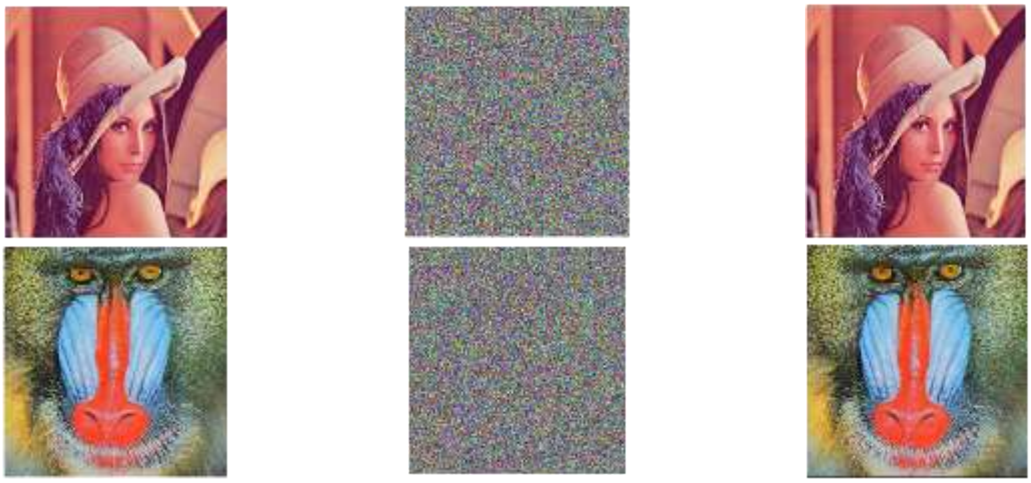
where x, y are real numbers called the states variables and a, b, c, d, e are positive control parameters selected as: a=4, b=1.1, c=4.4, d=0.1, e=8 to obtain the phase portraits of chaotic behavior. The initial values of the proposed chaotic system are two real number $x_0 = 1.2$ and $y_0 = 0.8$, in which $x_0, y_0 \in [1.1, 2.3]$.

Practically, algorithm (1) relates to the essential steps of keys generation map based on the two-dimensional chaotic system. The outputs from chaotic map generation procedure are three key stream vectors X, Y, Z each of size ($W\times H\times 3$).

```
Algorithm 1. Chaotic Keys Generation
Input: a, b, c, d, e, $x_o$, $y_o$, W, H
Output: X, Y, Z // keys stream vectors of dimension (1,N)
Begin
Processing:
Step1: $x_1 = x_o$, $y_1 = y_o$, $z_o = x_1 \oplus y_1$
Step2: N = $W\times H\times 3$
Step3: Iterate N-1 times
 $X_{i+1}$ = mod ( a $y_i^2$ – b $x_i^2$ – c, $N$ ) // i=2,3……N
 $Y_{i+1}$ = mod ( d $x_i$ $y_i$ – e $x_i$, $N$ ) // i=2,3……N
 $Z_{i+1} = X_{i+1} \oplus Y_{i+1}$
 $x_i = X_{i+1}$
 $y_i = Y_{i+1}$
 End Iteration
End
```

3.2. Permutation Scheme

The permutation process is considered a crucial issue in image encryption methods due to its ability for minimizing the correlation between image pixels. It changes the pixels positions only in the original plain image in order to mitigate the correlation between adjacent pixels. In this paper, the pixels positions are scrambled (reordered) according to the ascending order of the two key stream vectors X and Y generated by system (1). Each key stream vector has the size (W×H×3). The color plain mage I_m is decomposed into three basic colors matrices R_m, G_m, B_m of same size (W×H). Then, one dimension vector of size (W×H×3) named I_O represents the image pixels is created which composed of the three matrices R_m, G_m, B_m. The pixels positions in Io vector are shuffled according to the new indexing of the two chaotic vectors X and Y. An illustration of the permutation procedure for one round is depicted in Algorithm (2). The simulation results of permutation process are shown in Figure 2. The invers permutation task is conducted based on the same chaotic key vectors X and Y generated from system (1). Then, ascending sorting of both X and Y chaotic key vectors is performed in order to exploit vectors indexing in re-scrambling procedure (getting the recovered original image).



(a) (b) (c)

Figure 2. Simulation results of permutation process for lena and baboon color images with size (512×512×3), (a) The original image, (b) The permutated image, (c) The recovered image after invers permutation

Algorithm 2. Chaotic Map Indexing Based Permutation Scheme

```
Input: plain Image $I_m$, W, H, $X_{W\times H\times 3}$,$Y_{W\times H\times 3}$
Output: $I_P^{\sim}$ //Permutated Image
Begin
Processing:
Step1: $X_1 \leftarrow$ Ascending-sort(X),
Step2: $X_2 \leftarrow$ Ascending-sort(Y)
Step3: Decompose $I_m$ into three matrices $R_m$, $G_m$, $B_m$
Step4: $I_o \leftarrow$ Composed the matrices $R_m$, $G_m$, $B_m$ into one dimension vector of size (W×H×3)
Step5: $I_P \leftarrow$ Reorder pixels positions of $I_o$ using $X_1$
Step6: $I_P \leftarrow$ Reorder pixels positions of $I_P$ using $X_2$
Step7: $I_P^{\sim} \leftarrow$ Reshape $I_P$ size into (W, H, 3)
Step8: Return $I_P^{\sim}$
End
```

3.3. Diffusion Scheme

In this paper, a random chaotic map generation is realized using system (1). Its generate key stream of real values with arithmetic precision reached to 10^{-15} precision. As illustrated in algorithm (1), the adopted chaotic system generates three floating key vectors $X_{W\times H\times 3}$, $Y_{W\times H\times 3}$ and $Z_{W\times H\times 3}$. The original image I_m is permutated according to algorithm (2) in order to de-correlate the image pixels using two dimension vectors $X_{W\times H\times 3}$ and $Y_{W\times H\times 3}$. Then, the permutated image $I_P^{\sim}$ is divided into n sub-blocks with equal size, such as $I_P^{\sim}$={ $I_P^{\sim}$1, $I_P^{\sim}$2,…, $I_P^{\sim}$n} in order to carry out the diffusion process.

Related to the chaotic key vectors X and Y, another key stream vector Z is constructed by applying the bitwise XOR logical operation between X and Y. Subsequently, we divide each key vector into n sub-vectors, X={x_1, x_2, ….x_n}, Y={y_1, y_2, ….y_n} and Z={z_1, z_2, ….z_n}. In this way, each subblock image is feeds to single core along with its corresponding sublocks of chaotic key streams x_i, y_i, z_i where (i=1,…n) to perform the diffusion process simultaneously. In this way, the diffusion process is performed over $I_P^{\sim}$ subblocks in parallelism implementation. Afterwards, each pixel in sublock $I_P^{\sim}{}_i$ is diffused by applying XOR logical operation with the corresponding bytes in chaotic key subvectors x_i, y_i, z_i simultaneously. Figure 3 shows the simulation results of permutation – diffusion based image encryption algorithm.

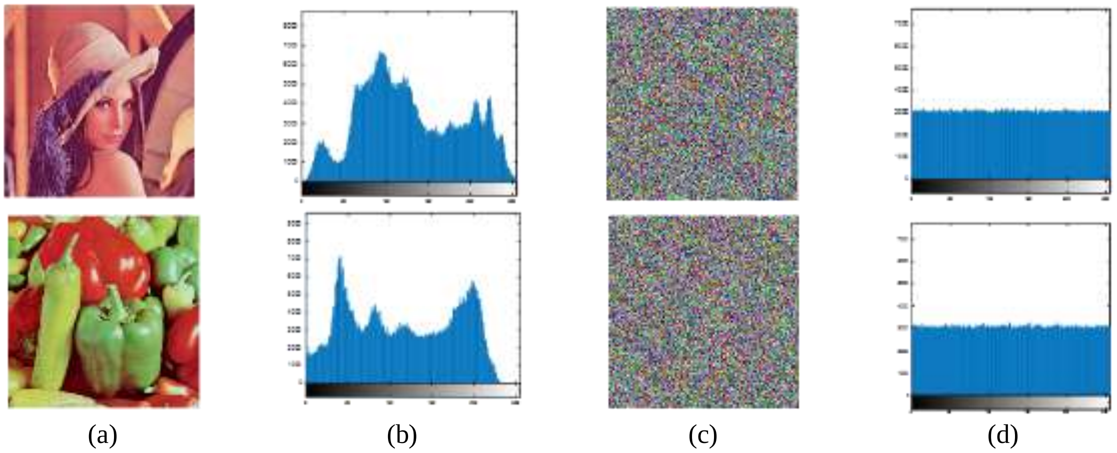


Figure 3. Simulation results of encryption processes, (a) Original image, (b) Histogram of original image, (c) Encrypted image, (d) Histogram of encrypted image

3.4. Parallelism Implementation Based Image Encryption

The most critical issue in chaotic system based cryptography methods is their high implementation time. From this perspective, the proposed encryption scheme is designed to implement image encryption in parallel manner in order to obtain an advantage on recent parallel architecture with fast encryption process. In order to exploit the parallelism property of CPU, the parallel computing toolbox in MATLAB offer a functional methods to perform the Parallel Processing of specific task using Parallel Data Processing method (SPMD). A single program with n different sub encrypted block as input on *n* cores is achieved to manage the parallelism implementation procedure. The (n) cores are worked in parallel manner to outperform the sub-encrypted blocks Se_i. The final encrypted image EI_m is obtained from integrating the n sub encrypted blocks Se_i as illustrated in Figure 4. Each subblock $I_P\tilde{}i$ in the permutated image $I_P\tilde{}$ is diffused at each core simultaneously using the corresponding three subblocks of the key stream vectors x_i, y_i, z_i. The encryption process based on permutation-diffusion processes at each core is presented in details in algorithm (3) and clarified in Figure 5.

```
Algorithm 3. Parallel Implementation Based Image Encryption
Input: Permutated Image $I_P$, $W, H, X, Y, Z, n$
Output: $EI_m$ // Encrypted Image
Begin
Step1: Processing
 Blocksize = [$(W/n)\times 2$, $(H/n)\times 2$]
 Divid $I_P$ image into n sub-blocks of size Blocksize
 Divid $X, Y, Z$ into $n$ sub-blocks of size Blocksize
Step2: In parallel Iterate for each subblock $I_P^i$ in $I_P$ // where $i=1,\ldots.n$
 Upload $I_P^i$, $x_1^{i,}$ $y_1^i$, $z_1^i$ to Graphic Processor Unit GPU
 $x_1i$ = $float\text{-}binary(x_1i)$     // to convert each element in the chaotic
 $y_1i$ = $float\text{-}binary(y_1i)$     subblocks $x_1i$, $y_1i$, $z_1i$ into 48 binary digits
 $z_1i$ = $float\text{-}binary(z_1i)$
 $x_1i$ = $mod(x_1i, 256)$              // to bound each element in the chaotic
 $y_1i$ = $mod(y_1i, 256)$              subblocks to 256 graylevel
 $z_1i$ = $mod(z_1i, 256)$
 $C1 = I_Pi \oplus x_1i$
 $C2 = C1 \oplus y_1i$
 $Se_i = C2 \oplus z_1i$
 Download $Se_i$ to CPU
 End iteration $i$
Step3: Integrating $Se_i$ subblocks into encrypted image $EI_m$
End
```

It's worth mentioning, that each floating point number generated from chaotic system (1) is converted into binary format composed of 48 bit. The decryption procedure is realized based on the same encryption chaotic keys produced from system (1). Its start by getting the encrypted image EI_m, then apply XOR logical operation between key stream vectors X, Y, Z and EI_m in parallel manner as illustrated in algorithm (3). Finally, apply invers permutation over the output image to get the original I_m.

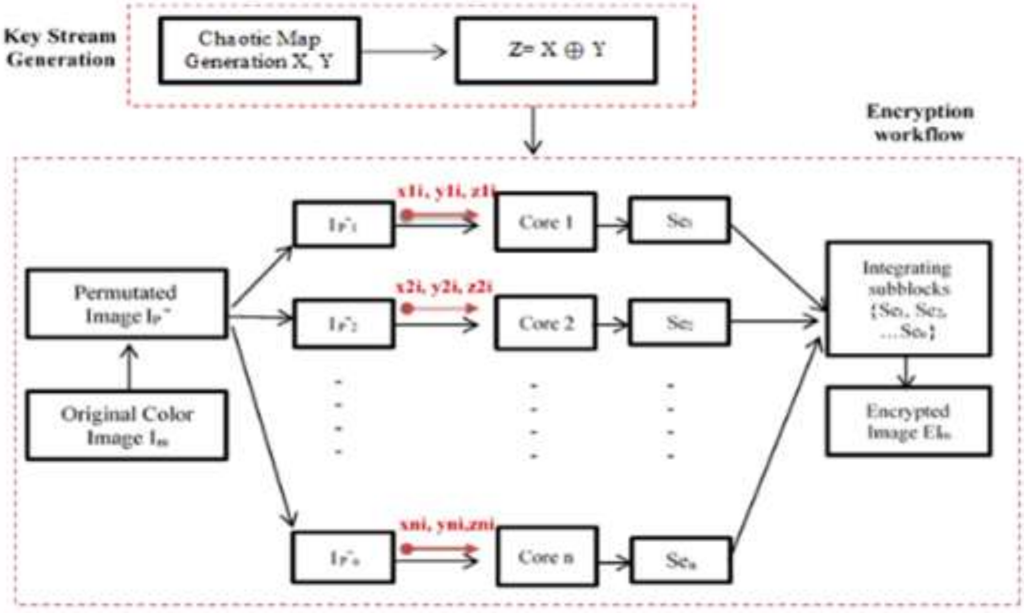


Figure 4. The workflow of parallel implementation based image encryption scheme

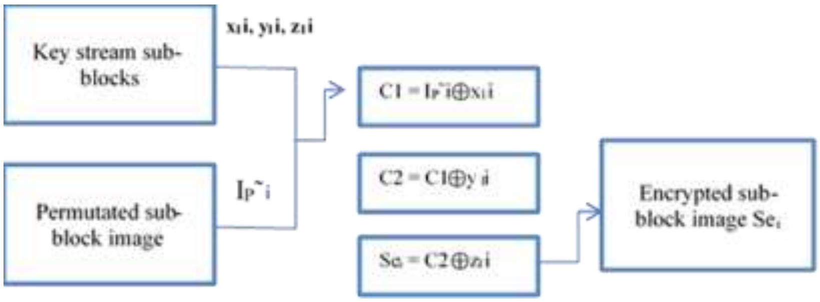


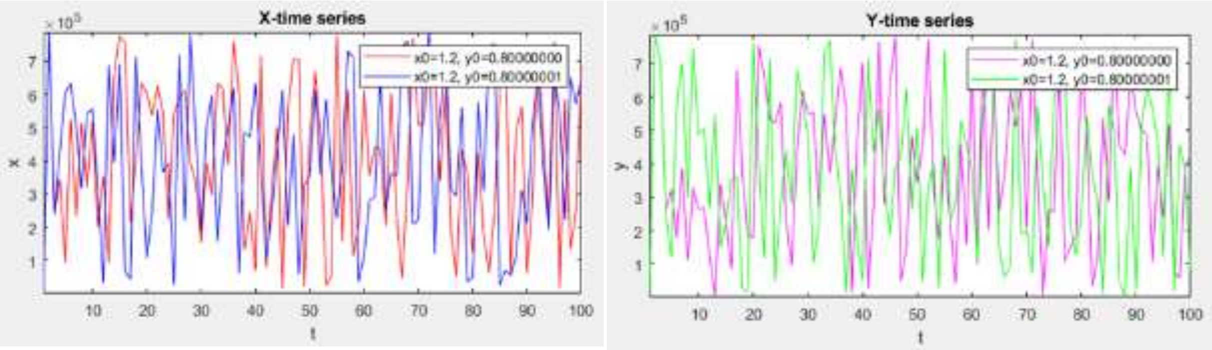
Figure 5. The workflow of sub-block image encryption process at each core

4. RESULTS AND SECURITY ANALYSIS

In this paper, a two-dimensional nonlinear chaotic system with seven terms is adopted to achieve image encryption with parallel implementation. The dynamical characteristics of the proposed system were analyzed, investigated and approved as mentioned in [8] which is based on several tools such as maximum Lyapunov exponents, Kaplan–York dimension, time series waveform, phase portraits, sensitivity to initial conditions. Furthermore, a parallel implementation based image encryption is presented using chaotic key streams generated from system (1). The numerical calculations were conducted and simulated in Matlab 2018. Based on the results, the proposed system is able to produce a wide range of chaotic map numbers x(t), y(t) with floating point precision 10^{-15}. Moreover, a parallel implementation of color image encryption based on chaotic map generation process is performed in this paper in order to minimize the computation time and programming efforts required for encryption computation. We have investigated the speed up of the proposed chaotic system on Cor i7 CPU, 2.20 GHz; RAM 6GB, OS win.8- 64bit using Matlab 2018 platform.

4.1. Sensitivity to Initial Conditions Analysis

The main property of chaotic system demonstrated by its long-term unpredictability [19]. practically, the chaotic system sensitivity based on its initial keys. Sensitivity to initial keys is analyzed in this paper using two different keys for encryption and decryption process over the same plain image. The two key utilized in this experiment are differs from each other in the 10^{-8} precision. Two different initial keys leads to change the chaotic map behavior. Figure 6(a) and Figure 6(b) exhibits that the evolution of the chaos trajectories is very sensitive to the initial conditions.



(a) (b)

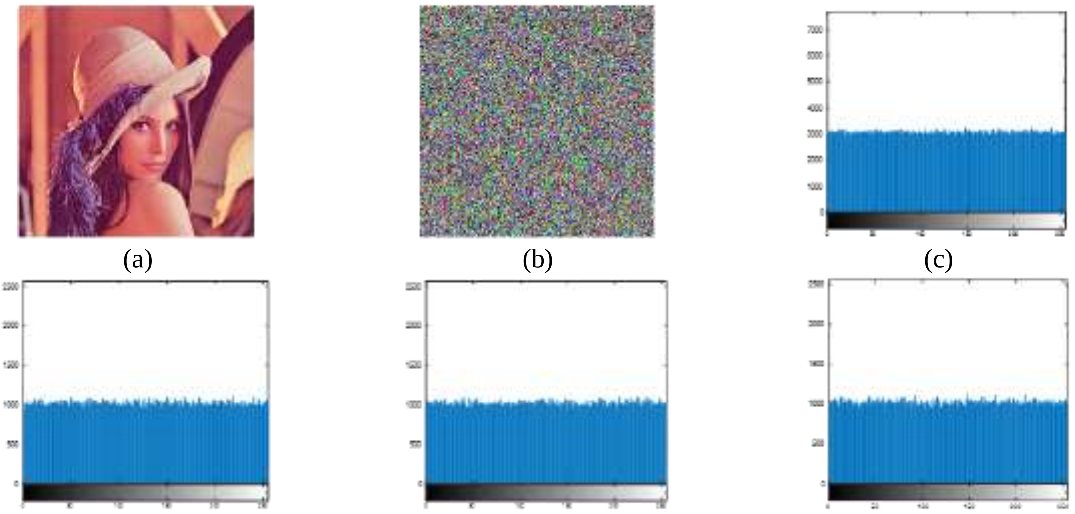
Figure 6. Simulation results of keys sensitivity, (a) X-time series with initial conditions (xo=1.2, yo= 0.800000000) in red line and (xo=1.2, yo=0.800000001) in blue line, (b) Y-time series with initial conditions (xo=1.2, yo=0.800000000) in magenta line and (xo=1.2, yo=0.800000001) in green line

4.2. Key Space Analysis

The key space is the aggregate number of various keys utilized in the encryption and decryption schems. The IEEE-754 floating-point standard [20] is based for floating computation precision with 64-bit double precision. For efficient cryptographic system and more secure, the key space should be large enough to make robust against the brute force attack [21]. In order to realize a larger key space and conquer the lower complexity of secret keys in one dimensional chaotic system, a two dimensional chaotic system was employed in this paper for color image encryption. Furthermore, the permutation-diffusion process are iterated four rounds to achieve an adequate a perturbing in image pixels positions and values. The computational precision of the 64-bit double-precision number is around 10^{-15}. Thus, the considered key space for the initial conditions x_o, y_o (used as a part of the key) is about $10^{15}\times10^{15} \approx 2^{100}$. The chaotic keys employed for encryption task composed three keys X,Y and Z, each has a key space of six byte, i.e the key space for the chaotic keys is $2^{48}\times2^{48}\times2^{48}=2^{144}$. Therefore, the total key space for the proposed encryption system is: $2^{100}\times2^{144.}= 2^{244}$. For robust cryptosystem, the key space that (greater or equal 2^{100}) should be resist the brute-force attack [22].

4.3. Histogram

The histogram analysis reflects the image pixels distribution graphically, it's often used to exhibit the uniform distribution of the encrypted image pixels. Thus, the attacker cannot inferred any valuable information via adopting the statistical attack. Figure 7(a)–7(f) illustrate the original image I_m, encrypted image EI_m, histogram of encrypted image EI_m, histogram of red layer of EI_m, histogram of green layer of EI_m, histogram of blue layer of EI_m. Obviously, the histogram demonstration of the encrypted image and its three components red, green, blue are almost uniformly distributed. The encrypted image EI_m considerably differ from that of the original image, subsequently do not offer any evidence for exploiting the statistical attack. Furthermore, the permutation scheme adopted in this work is specifically shuffles the pixels locations only; i.e there is no changing of pixel intensity. Consequently, the histogram of the permutated image is the same as that of the original image.

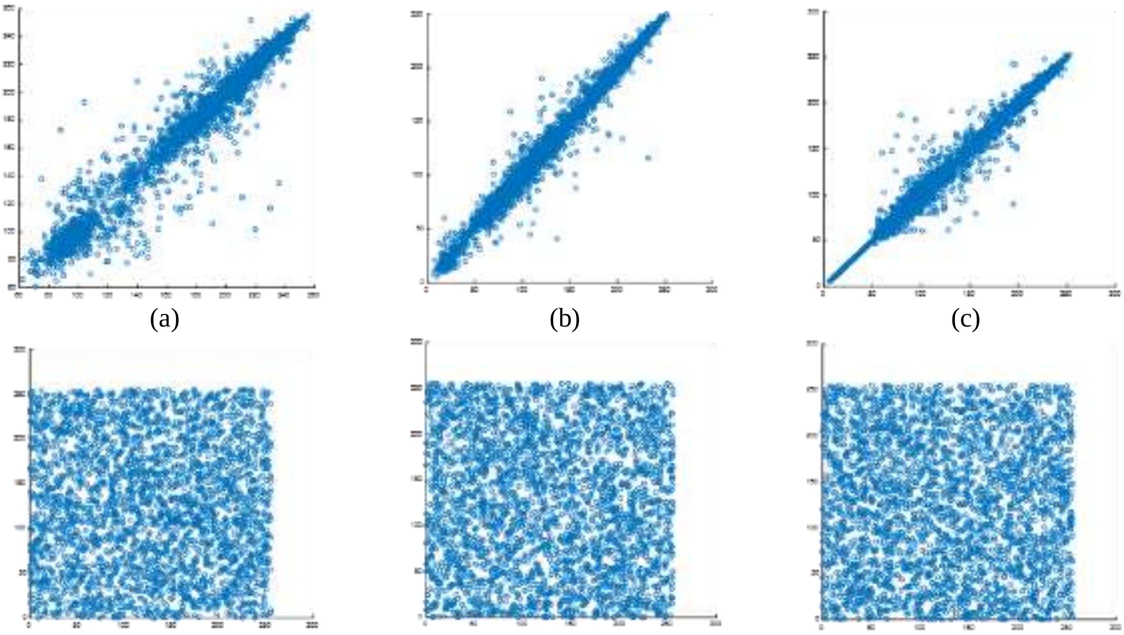

(a) (b) (c)

(d) (e) (f)

Figure 7. Histogram Demonstration of the encrypted Lena image (a) Original Image I_m, (b) Encrypted image EI_m, (c) Histogram of Encrypted image EI_m, (d) Histogram of Red layer, (e) Histogram of Green layer, (f) Histogram of Blue layer

4.4. Correlation Coefficients Analysis

Correlation coefficient refers the pixels distribution and their (dependency or independency) relationships in the image. The linear relationship between adjacent pixels indicates a higher correlation (correlation value near to 1). While the non-linear relationship between adjacent pixels implies low correlation value (correlation value near to 0) [23]. The experimental results of correlation coefficients related to standard images for both original and encrypted images are listed in Table 1. Figure 8 shows the graph demonstration for correlation coefficients of Lena image.


(a) (b) (c)

(d) (e) (f)

Figure 8. Simulation results of correlation analysis of lena image, (a) Diagonal, (b) Vertical, (c) Horizental, (d) Encrypted diagonal, (e) Encrypted vertical, (f) Encrypted horizontal

In addition, the contrast, homogeneity and energy measurements are computed for the encrypted images as shown in Table 2. The contrast measurement enables the viewer to determine the objects in an image. The higher value of contrast measurement referred to robust encryption method. Energy measurement is used to depict the information amount in a given image. For efficient encryption, energy measurement should have a low value as much as possible. The homogeneity measure obtained from Gray level co-occurrence matrix (GLCM) is used to describe the potency of combinations of pixel brightness results in tabular form. For better encryption process, the homogeneity measurements of the encrypted images should be small values.

Table 1. Simulation Resultes of Correlation Coefficientc

Image	Correlation Coefficients of original image		Correlation Coefficients of Encrypted image	
Lena	Diagonal	0.9754	Diagonal	-0.0176
	Vertical	0.989	Vertical	-0.0356
	Horizontal	0.986	Horizontal	-0.0122
Baboon	Diagonal	0.853	Diagonal	-0,0182
	Vertical	0.858	Vertical	-0.0248
	Horizontal	0.900	Horizontal	-0.0068
Peppers	Diagonal	0,953	Diagonal	-0.0072
	Vertical	0.981	Vertical	0.0011
	Horizontal	0.980	Horizontal	-0.0007

Table 2. The Simulation Results of Contrast, Homogeneity and Energy Measurements

Image	Contrast	Homogeneity	Energy
Lena	10.52	0.3892	0.0156
Baboon	10.51	0.3899	0.0156
Peppers	10.49	0.3893	0.0155

4.5. Entropy Information Analysis

The entropy analysis relates with the randomness measure of image. Significantly, the higher entropy value reflects more uncertain and unintelligible in image information and calculated according to formula bellow [8, 24]:

$$E(S) = \sum_{i=0}^{2^n-1} P(s_i) \frac{1}{log_2 P(s_i)} \tag{2}$$

where n is bits number which represent the symbol si in the Source S. P(s_i) indicates the probability of symbol si, thus the entropy measure represented by bits. For ideal random source send out 2^n symbols, the entropy is realized to E(s)= n. thus, for a encrypted image with 256 color, the entropy should be equal to 8. Table 3. Exhibits the entropy measurement for different encrypted images and the corresponding RGB layers based on the proposed encryption method with entropy values close to 8.

4.6. Differential Attacks Analysis

To hold out the differential attack, a useful encryption method should guaranteed that any small updating in the original image should leads to a noticeable discrepancy in the encrypted image. Two major measures are adopted in this paper to evaluate and analyze the differential attacks such as; NPCR (number of pixels change rate) and UACI (unified average changing intensity). NPCR measurement used to investigate the influence of one pixel change in the whole image. Its computed according the mathematics formula given below [25, 26]:

$$NPCR = \frac{1}{N \times M} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \ \times 100\% \tag{3}$$

$$where\ D(i,j) = \begin{cases} 0\ if\ C1(i,j) = C2(i,j) \\ 1\ if\ C1(i,j) \neq C2(i,j) \end{cases}$$

$$UACI = \frac{1}{N \times M} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C1(i,j) - C2(i,j)|}{255} \times 100\% \tag{4}$$

where C1(i, j) and C2(i, j) are pixel intensity of the two encrypted images C1 and C2 whose plain images have only one pixel difference. The demonstration results of differential attack analysis are illustrated in Table 3.

Table 3. Simulation Resultes of Differaintail Attaks and Enropy Analysis

Image Name	NPCR	UACI	Entropy of original image	Entropy of Red Layer	Entropy of Green Layer	Entropy of Blue Layer
Lena	99.63	33.45	7.99975	7.9994	7.9993	7.9992
Baboon	99.60	33.49	7.99978	7.9994	7.9991	7.9991
Peppers	99.61	33.46	7.99975	7.9993	7.9991	7.9991

Table 4 shows the performance comparison regards NPCR and UACI measurements against the algorithms in [26-28].

Table 4. The Comparision Resultes of NPCR, UACI against Other Methods

Method	NPCR (average)	UACI (average)
Proposed	99.613	33.466
Ref. [26]	99.610	33.463
Ref. [27]	99.219	33.317
Ref. [28]	98.110	32.979

4.7. Speed Time Analysis

The main contribution of this work is to implement a color image encryption task based on parallelism technique in order to minimize the computation time of encryption stages compared with other research. The encryption speed factor is considered a critical issue in real time security applications. In order to validate the proposed parallel implementation based color image encryption algorithm, we determined the average consuming time of encryption task for different images. From this view, a beneficial comparison was carried out in our experiments regards the consuming time of encryption process per round in parallel manner against other researchers works [26-29] described in Table 5.

Table 5. The Computation Time (Sec) of Encryption Process Per Round

The proposed		Ref. [26]		Ref. [28]		Ref. [29]	
0.0235		0.2123		0.3823		0.312	
Platform	i7 CPU, 2.20 GHz RAM 6GB, OS win.8-64bit	Platform	i7 CPU, 2.60 GHz RAM 8GB, OS win7-64bit	Platform	(TM) i7-4770 CPU 3.4 GHz	Platform	(PC) 1.8 GHz Celeronl processor, RAM 0.99 GB 80 GB hard disk capacity

5. CONCLUSION

This paper introduces a parallel implementation based color image encryption algorithm based on two dimension chaotic system presents in our previous work. Practically, the proposed encryption algorithm is based mainly on permutation-diffusion processes. From the experiments, the permutation process is contribute to scramble the image pixels with assist of chaotic key streams in order to provide the randomness distribution of pixels positions. In the diffusion process, the permutated pixel values are then altered in parallel implementation by combining (applying XOR logical operation) with the key stream values that are generated by a two dimensional chaotic map. A small change in the plane image can be spread overall pixels in the cipher image as shown in the experimental results of NPCR and UACI measurements. In this way, the image histogram is significantly changed after permutation process. A proper and ideal values of differential attacks parameters (NPCR, UACI) and statistical analysis (information entropy) are obtained as a result of the proposed encryption algorithm. From the experimental results, the output image of the encryption task shows a higher randomness of the encrypted image which can be effectively resistant to attacker.

ACKNOWLEDGEMENTS

This research was partially supported by computer science department-college of education at Al Mustansiriyah University.

REFERENCES

[1] El-Samie, F.E.A., Ahmed, H.E.H., Elashry, I.F., Shahieen, M.H., Faragallah, O.S., El-Rabaie, E.S.M., and Alshebeili, S.A., "Image encryption: a communication perspective," *CRC Press*, 2013.

[2] Alvarez G. and Li S.,"Some basic cryptographic requirements for chaos based cryptosystems," *Int. J.of Bifur. and Chaos*, vol. 16, pp. 2129-2151, 2006.

[3] Amigo J.M. Kocarev, L.,Szczepanski, J. "Theory and practice of chaotic cryptography," *Phys. Lett.* A 2007, 366, 211-216.

[4] Alvarez, G., Li, S, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurc. Chaos* 2006, 16, 2129-2151.

[5] Hong-Mei Yuan, Ye Liu, Tao Lin, Ting Hu, Li-Hua Gong, "A new parallel image cryptosystem based on 5D hyper-chaotic system," *Signal Processing, Image Communication*,2017.

[6] Abraham F. Vergara, Everardo I. González, Enrique E. Guerrero, Oscar R. López, Eduardo R. Orozco, Juan M. Ontiveros, José R. Valdez and Esteban T. Cuautle, " Implementing a Chaotic Cryptosystem by Performin Parallel Computing on Embedded Systems with Multiprocessors," *MDPI, Entropy journal*, 2019.

[7] Ünal Çavuşoğlu, Sezgin Kaçar, "A novel parallel image encryption algorithm based on chaos," *Springer Science and Business Media*, 2019.

[8] Khalid A. Hussein and Sawsen A. Mahmood, "A parallel Programming for Robust Chaotic Map Generation Based on Two Dimensional Equation System," *Journal of Engineering and Applied Sciences*, vol. 14, pp. 3741-3745, 2019.

[9] R. Matthews, "On the derivation of a 'chaotic' encryption algorithm. Cryptologia," *Journal Cryptologia*, vol. 8, pp. 29-42, 1989.

[10] J. Fridrich, "Symmetric ciphers based on two–dimensional chaotic Maps," *International Journal of Bifurcation and Chaos*, vol. 8, pp. 1259-1284, 1998.

[11] Xiaopeng Wei, Bin Wang, Qiang Zhang and Chao Che, "Image Encryption Based on Chaotic Map and Reversible Integer Wavelet Transform," *Journal of Electrical Engineering*, vol. 65, pp. 90-96, 2014.

[12] Moatsum A., Azman S., Je Sen T., Rami S. Alkhawaldehb, "A New Hybrid Digital Chaotic System with Applications in Image Encryption," *Signal Processing Journal*, February 2019.

[13] Zenggang X., Yuan W., Conghuan Y., Xuemin Z., Fang X., "Color Image Chaos Encryption Algorithm Combining CRC and Nine Palace Map," *Springer Science and Business Media*, 2019.

[14] G.Chaitanaya, B.Keerthi, A.Saleem, A.Trinadh Rao and K.T.P.S.Kumar, "An Image Encryption and Decryption using Chaos Algorithm," *IOSR Journal of Electronics and Communication Engineering*, vol. 10, pp. 103-108, 2015.

[15] Xuncai Zhang, Feng Han, and Ying Niu, "Chaotic Image Encryption Algorithm Based on Bit Permutation and Dynamic DNA Encoding," *Hindawi Computational Intelligence and Neuroscience*, 2017.

[16] Siva Janakiraman, K. Thenmozhi, John Bosco Balaguru Rayappan, engarajan Amirtharajan, "Lightweight chaotic image encryption algorithm for real-time embedded system: Implementation and analysis on 32-bit microcontroller,", *Elsevier Microprocessors and Microsystems: Embedded Hardware Design (MICPRO)*, vol. 56, pp. 1-12, 2018.

[17] S. N. Lagmiri, N. Elalami, and J. Elalami, "Color and gray images encryption algorithm using chaotic systems of different dimensions,", *IJCSNS International Journal of Computer Science and Network Security*, vol.18, pp. 79-86, 2018.

[18] Guodong Ye, "A Chaotic Image Encryption Algorithm Based on Information Entropy," *International Journal of Bifurcation and Chaos*, vol. 28, pp. 1-11, 2018.

[19] Riecke, H., Roxin, A., Madruga, S., Solla, S.A.," Multiple attractors, long chaotic transients, and failure in small-world networks of excitable neurons," *Chaos* 2007.

[20] An American National Standard, "754-1985 - IEEE Standard for Binary Floating-Point Arithmetic," *Standards Committee of the IEEE Computer Society*, 1985.

[21] Fu, C.; Lin, B.b.; Miao, Y.s.; Liu, X.; Chen, J.J. "A Novel Chaos-Based Bit-Level Permutation Scheme for Digital Image Encryption,". Opt. Commun. 2011, vol. 284, pp. 5415-5423.

[22] L. Y. Zhang, C. Li, K.-W. Wong, S. Shu, and G. Chen, "Cryptanalyzing a chaos-based image encryption algorithm using alternate structure," *J. Syst. Softw.*, vol. 85, no. 9, pp. 2077-2085, 2012.

[23] Francisca E. Canales, Iván R. Cambero, Lucio R. Herrera, Cesar C. Bello, "Pseudo-random bit generator using chaotic seed for cryptographic algorithm in data protection of electric power consumption," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 2, pp. 1399-1409, 2019.

[24] Yong Z., "A Chaotic System Based Image Encryption Algorithm using Plaintext-related Confusion," *TELKOMNIKA (Indonesian Journal of Electrical Engineering)*, vol. 12, no. 11, pp. 7952-7962, 2014.

[25] Liu, H., Kadir, A., Gong, P.,"Chaos-Based Image Encryption Algorithm Using Decomposition," *TELKOMNIKA (Indonesian Journal of Electrical Engineering)*, vol.12, no.1, pp. 575-583, 2014.

[26] Ping Ping, Jiny Fan, Yingchi Mao, Feng Xu, and Jerry Gao. "A Chaos Based Image Encryption Scheme Using Digit-Level Permutation and Block Diffusion," *IEEE Access*, vol. 6, pp. 67581-67593, 2018.

[27] G. Ye and X. Huang, "An effcient symmetric image encryption algorithm based on an intertwining logistic map," *Neurocomputing*, vol. 251, pp. 4553, 2017.

[28] Z. Hua and Y. Zhou, "Image encryption using 2D Logistic-adjusted-Sine map," *Inf. Sci.*, vol. 339, pp. 237-253, 2016.

[29] X. Liao, S. Lai, and Q. Zhou, "A novel image encryption algorithm based on self-adaptive wave transmission," *Signal Process.*, vol. 90, pp. 2714-722, 2010.