Journal of Engineering and Applied Sciences 14 (19): 6954-6958, 2019
ISSN: 1816-949X

Image Encryption Based on Chaotic Mapping and Random Numbers

Salah T. Allawi
Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq
Salah.Taha@uomustansiriyah.edu.iq

Abstract: Many researches called for the producing new encryption algorithms as a method of protecting the information from unauthorized persons when stored or transmitted over the internet. A new method to encrypt the RGB image is presents by this research. This method is relied on three levels to make security better and provide an encrypted image with a few correlation pixel, high degree of entropy and histogram has a distributed uniform, first level, dividing the image into two equal parts. Second level, encrypting the information of each part using a secret key generated by using 1D logistic mapping. Third level, permutation the pixels position using random numbers generated by using LFSRs. Several tests done to verify that the suggested encryption method is safe.

Key words: Image encryption, LFSR, permutation, logistic map, RGB image, encryption method

INTRODUCTION

Now, the trend toward transferring data has increased through the use of the computer network, especially, since, the internet is the fastest way to transfer data (Khanzadi *et al*., 2014). The internet has many vulnerabilities and therefore, information transmitted through it is vulnerable to attack by unauthorized persons (Aziz and Ahmed, 2015). The security is one of the important objectives to be achieved when transmitting data over the internet (Loukhaoukha *et al*., 2012). For this reason, several encryption methods have been proposed to protect information such as substitution, transposition, RSA, DS, AES, etc. (Dewangan and Kamargaonkar, 2015; Ail and Alobaidy, 2016). There are two causes makes these methods do not fit image encryption. First, the size of the image is larger than the text and the method of encoding the image in traditional ways requires a lot of time, secondly, the original text should be equal to the encoded text (Dhannoon *et al*., 2011). These reasons led to produce a new methods of encryption is capable of providing the protection for the information, therefore, chaotic maps consider one of the most important new ways in this field (Loukhaoukha *et al*., 2012).

Literature review: Ahmad and Alam (2009) suggest a new way for image encryption depended on more than two levels from chaotic maps. This algorithm includes: divided the plain-image into blocks the size of each one 8×8, generated random numbers by using 2D cat map to shuffling the blocks and the chaotic sequence that generated by one dimensional logistic map used to encrypt the shuffled image.

Alshibani (2015) suggest a new method using a blend of 2D chaotic maps for RGB image encryption. The maps of chaotic Henon, chaotic Burger and chaotic ginger bread man are used to face the needs of the protected image transmission. The suggested image encryption project is consisted of two transformation and two substitution processes.

Banthia and Tiwari (2013) suggest method depended on two techniques for image encryption. First, include process of generate random numbers to encrypted the image by using linear congruential generator. These numbers are used as key for reordering the positions of pixels, columns and rows in the image. Second, include generate a sequences of random numbers which is used as a key for rearrangement the positions of the pixels, columns and rows in the image. These numbers are generated by using logistic maps.

Chaotic mappings used

2D cat mapping: The research of periodic theory by V.I. Arnold presented the 2D cat map. Let, a 2D cat map have two positive number represent control parameters and the coordinates of positions of pixels in the plain image represent by Z = {(q, p)| q, p = 1, 2, 3, ..., M} (Ail and Alobaidy, 2016; Ahmad and Alam, 2009) can be seen in:

$$q' = (q+a*p) \bmod (m) \quad (1)$$

$$p' = (b*q+(a*b+1)\ p) \bmod (m) \quad (2)$$

Where:

(a, b) = Positive numbers which represent control parameters
(M*M) = The size of the original image
(q, p) = The position of the original pixel
(q', p') = The new position of the pixel when the cat map is done once

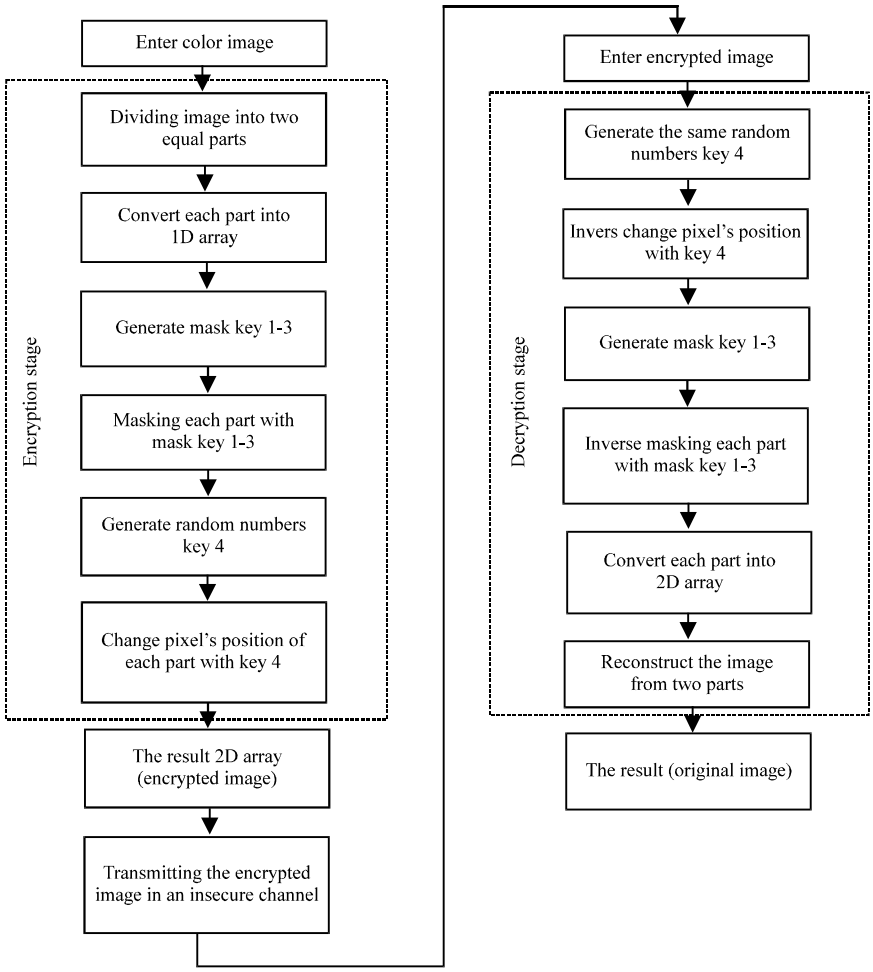

Fig. 1: The general outline of encryption and decryption stage

Cat map rearrange the position of the original pixels by replacing the original position with the new position that were generated. The image appears distorted after a number of iterations because distorted the correlation between the adjacent pixels. But when iterating the work several iterations the plain image will be return, the cat map is periodic. To resolve this problem, 2-D Zaslavskii map is used to generate parameters (a, b) at random (Ahmad and Alam, 2009).

1D logistic mapping: The 1D logistic map is one of the simplest non-linear chaotic methods proposed by R.M. May. Can be defined as follows:

$$x_{m+1} = a*x_m(1-x_m) \quad (3)$$

Where:

m = The number of iterations
a = The system parameter 3.57<a<4
x_o = Initial condition

Image is encrypted by using the random numbers that is generated by Eq. 3 (Ahmad and Alam, 2009).

Suggested method: The propose method depending on three steps: dividing image, encryption step and reordering pixels position step. Figure 1 illustrate the outline of the proposed method.

Encryption stage: The encryption stage includes the following steps. Entering color image (w*h) and then dividing the image into two equal parts (p1, p2) with size (w/2, h). Converting the pixels value of each part into 1-D array with size (w/2*h) = m pixels (i = 1, 2, 3, ..., m). Three color components of pixel i converted to three values of variable x_{red}, x_{green}, x_{blue}. Applying 1D logistic mapping to generating three keys (k_r, k_g, k_b) one for each color (red, green, blue) by using control parameters, k_r = 0.78916, k_g = 0.12345, k_b = 0.45678 and a = 3.99995:

$$k_{r+1} = a*k_r*(1-k_r) \quad (4)$$

$$k_{g+1} = a*k_g*(1-k_g) \quad (5)$$

$$k_{b+1} = a*k_b*(1-k_b) \quad (6)$$

After applying Eq. 4-6 m times, we obtain a decimal values between (0-1). To converting these values to integer value between (0-255) using Eq. 7-9:

$$k_r = (k_r*10^{14}) \bmod 255 \quad (7)$$

$$k_g = (k_g*10^{14}) \bmod 255 \quad (8)$$

$$k_b = (k_b*10^{14}) \bmod 255 \quad (9)$$

To encrypted part 1 and 2 applying XOR operation between pixels value for each color (red, green, blue) with corresponding key (k_r, k_g, k_b) by using Eq. 10-12:

$$C_r = k_r \text{ XOR } x_{red} \quad (10)$$

$$C_g = k_g \text{ XOR } x_{green} \quad (11)$$

$$C_b = k_b \text{ XOR } x_{blue} \quad (12)$$

The process of reordering pixels position for the two part include generate a set of random numbers using LFSR consists of three registers that have different primary lengths (29, 31, 37), respectively with different connecting function as shown in Fig. 2. The length of sequence bits that will be generated by using LFSR and then convert it to random numbers depended on the size of the part of image (w/2*h), therefore, this length will be changed with each image has a new size. These random numbers are stored in 1-D array its size (w/2*h).

Each random number indicates the pixel location that will be called from the two parts, respectively, to be rearranged in a 2D array represent the encrypted image, whose size is w*h as shown in Fig. 3 (Algorithm 1).

Algorithm 1; Illustrates the main steps of encryption image:

1. Entering color image (RGB)
2. Dividing the image into two equal parts ((Width/2), height)
3. Converting each part into to 1-D arrays ((Width/2)* height)
4. Generating three masks key 1-3, one for each color (red, green, blue)
5. Applying XOR operation between part1 pixels (red, green, blue) with the mask key 1-3
6. Applying XOR operation between part 2 pixels (red, green, blue) with the mask key 1-3
7. Generating random numbers key 4 by using LFSRs
8. Reordering pixels position of the 2 parts depending on the random numbers
9. The resulting of reordering step 2-D array represent the encrypted image

Decryption stage: This stage include: load stego-image which is to be decrypted, reconstruct the original position of the pixels in the two parts by generate the same random number, generate same three keys (mask) to decrypted the 2 parts and then reconstructed the original image (Algorithm 2).

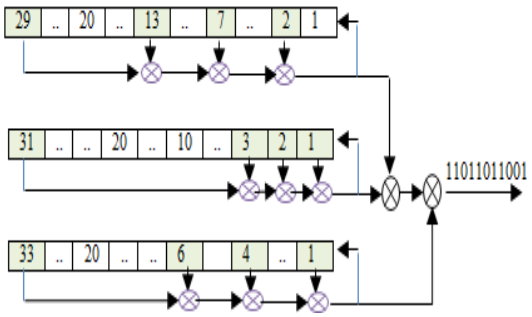

Fig. 2: Structure the generator

R.No.	No.	P1	P2	Encrypted image (w*h)					
3	1	210	88	100	33	65	210		
5	2	130	250						
1	3	100	33						
4	4	90	200						
2	5	65	210						

Fig. 3: Process of reordering pixels position

Algorithm 2; Illustrate algorithm 2 the main steps of decryption image:

1. Entering encrypted image
2. Generating the same random numbers by using LFSRs
3. Using the random number to getting back the data of the two 1-D arrays
4. Generating three masks key 1-3, one for each color (red, green, blue)
5. Applying XOR operation between part1 pixels (red, green, blue) with the mask key 1-3
6. Applying XOR operation between part 2 pixels (red, green, blue) with the mask key 1-3
7. Reconstructing the original image from the two 1-D arrays
8. Save the result image

MATERIALS AND METHODS

Proposed methodes test: In this study, the results of the tests performed on one of the examples will be displayed after applying the proposed method for image encryption. the steps of the proposed method can be seen in Fig. 4.

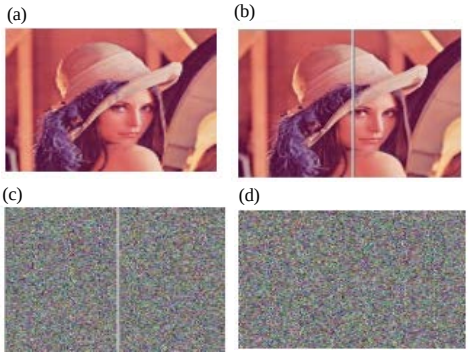

Fig. 4: Result of image encrypted method: a) Original image; b) Dividing image; c) Encrypted two parts image and d) Permutation image

RESULTS AND DISCUSSION

Histogram: The statistical features of images are presented using histogram. To protect this images from the statistical attacks, information must be distributed uniformly in the image. Figure 5 shown the histogram of the plain and stego image.

NPCR & UACI: To determine the effect of the change between pixels in the plain and stego image, the rate of change in pixels (NPCR) is calculated by using the following Eq. 13 and 14 (Huang *et al*., 2013):

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M*N} *100 \quad (13)$$

$$D(i,j) = \begin{cases} 0, & \text{if } C_1(i,j) = C_2(i,j) \\ 1, & \text{if } C_1(i,j) \neq C_2(i,j) \end{cases} \quad (14)$$

To calculate the difference rate between the encrypted and original image (UACI) use the following Eq. 15 (Alshibani, 2015; Huang *et al*., 2013):

$$UACI = \sum \frac{|C_1(i,j) - C_2(i,j|)}{255} *100 \quad (15)$$

Table 1 shown the difference of NPCR and UACI value.

Entropy: The level of uncertainties in the system defined as entropy. The entropy value is calculated by using Eq. 16 as (Reyad *et al*., 2016):

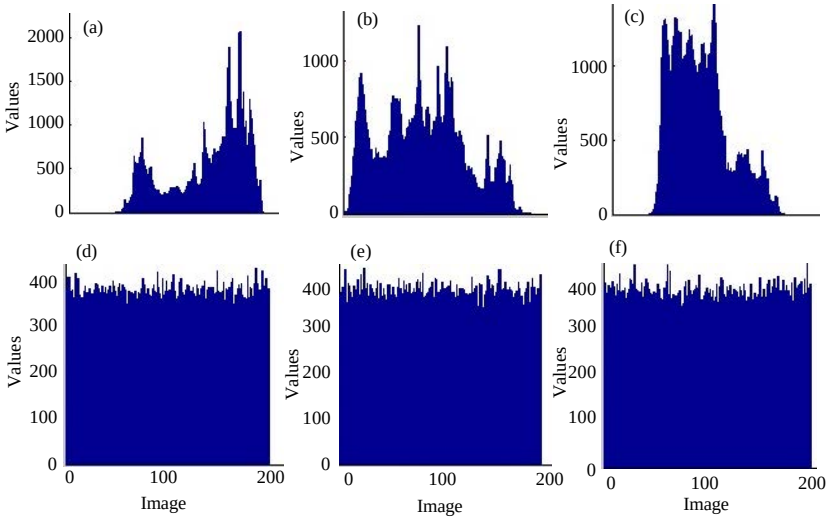

Fig. 5: Histogram for plain image and Histogram for encrypted image: a) Red; b) Green; c) Blue; d) Red; e) Green and f) Blue

Table 1: The difference of NPCR and UACI value

Encryption method	NPCR	UACI
Proposed method	99.6260	30.6600
Abraham and Daniel (2013)	99.6410	28.6200
Huang *et al.* (2013)	99.5400	28.2700
Sivakumar and Venkatesan (2014)	98.5460	28.8065
Loukhaoukha *et al.* (2012)	99.5850	28.6210

Table 2: The difference of entropy value

Encryption method	Entropy
Proposed methods	7.9980
Abraham and Daniel (2013)	7.9930
Huang *et al.* (2013)	7.9967
Sivakumar and Venkatesan (2014)	7.9920
Loukhaoukha *et al.* (2012)	8.9960

$$H(m) = -\sum_{i=0}^{255} P(m_i) \log_2 P(m_i) \quad (16)$$

where, $P(m_i)$ represents the probability of symbol m_i and the entropy is expressed in bits. The best value of entropy for the image encrypted should ideally 8. Table 2 shown the difference of entropy value.

CONCLUSION

A new suggested method given in this research, for image encryption. This way depends on fragmentation the image into two parts, encryption these parts and permutation pixels position through generated random numbers by using LFSRs. The count of random numbers that are generated represent 1/2 the size of the image. One of the benefits of this method, the result image can be sent to the receiver with JPEG form and the receiver can recover the original image. Another benefit of this method consists of several levels to make security more and provide an encrypted image with a few correlation pixel, high degree of entropy and a histogram has distributed uniform.

REFERENCES

Abraham, L. and N. Daniel, 2013. Secure image encryption algorithms: A review. Intl. J. Sci. Technol. Res., 2: 186-189.

Ahmad, M. and M.S. Alam, 2009. A new algorithm of encryption and decryption of images using chaotic mapping. Int. J. Comput. Sci. Eng., 2: 46-50.

Ail, Y.H. and Z.A. Alobaidy, 2016. Images encryption using chaos and random generation. Eng. Technol. J., 34: 172-179.

Alshibani, D.R., 2015. A new RGB image encryption based on a combination of 2D chaotic maps. Intl. J. Comput. Appl., 118: 29-35.

Aziz, M.M. and D.R. Ahmed, 2015. Simple image scrambling algorithm based on random numbers generation. Intl. J., 5: 434-438.

Banthia, A.K. and N. Tiwari, 2013. Image encryption using pseudo random number generators. Intl. J. Comput. Appl., 67: 1-8.

Dewangan, R.P. and C. Kamargaonkar, 2015. Image encryption using random permutation by different key size. Intl. J. Sci. Eng. Technol. Res., 4: 3531-3535.

Dhannoon, B.N., N.A. Hassan and S. Latef, 2011. Color image encryption using random password seed and linear feed backshift register. J. Al. Nahrain Univ. Sci., 14: 186-192.

Huang, C.K., C.W. Liao, S.L. Hsu and Y.C. Jeng, 2013. Implementation of gray image encryption with pixel shuffling and gray-level encryption by single chaotic system. Telecommun. Syst., 52: 563-571.

Khanzadi, H., M. Eshghi and S. Borujeni, 2014. Image encryption using random bit sequence based on chaotic maps. Arabian J. Sci. Eng. Springer Sci. Bus. Media BV., 39: 1039-1047.

Loukhaoukha, K., J.Y. Chouinard and A. Berdai, 2012. A secure image encryption algorithm based on Rubiks cube principle. J. Electr. Comput. Eng., 2012: 7-7.

Reyad, O., Z. Kotulski and W.M. Abd-Elhafiez, 2016. Image encryption using chaos-driven elliptic curve pseudo-random number generators. J. Appl. Math. Inf. Sci., 10: 1283-1292.

Sivakumar, T. and R. Venkatesan, 2014. Image encryption based on pixel shuffling and random key stream. Intl. J. Comput. Inf. Technol., 3: 1468-1476.