Journal of AI and Data Mining
Vol 5, No 1, 2017, 29-37

DOI: 10.22044/jadm.2016.735

A stack-based chaotic algorithm for encryption of colored images

H. Khodadadi[1*] and O. Mirzaei[2]

1. Department of Computer Engineering, Minab Branch, Islamic Azad University, Minab, Iran.
2. Computer Security Lab, Department of Computer Science and Engineering, Universidad Carlos III de Madrid, Madrid, Spain.

Received 04 July 2015; Accepted 18 August 2016
**Corresponding author: habibekhodadadi@gmail.com (H. Khodadadi).*

Abstract
In this paper, a new method is presented for encryption of colored images. This method is based upon using stack data structure and chaos, which make the image encryption algorithm more efficient and robust. In the proposed algorithm, a series of data whose range is between 0 and 3 is generated using the chaotic logistic system. Then the original image is divided into four sub-images, which are subsequently pushed into the stack based on the next number in the series. In the next step, the first element in the stack, which includes one of the four sub-images, is popped, and this image is divided into four other parts. Then, based on the next number in the series, four sub-images are pushed into the stack again. This procedure is repeated until the stack is empty. Therefore, during this process, each pixel unit is encrypted using another series of chaotic numbers (generated by the Chen chaotic system). This method is repeated until all pixels of the plain image are encrypted. Finally, several extensive simulations on well-known USC datasets are conducted to show the efficiency of this encryption algorithm. The tests performed show that the proposed method has a really large key space and possesses a high-entropic distribution. Consequently, it outperforms the other competing algorithms in the case of security.

Keywords: *Chaos, Encryption of Colored Images, Chen Chaotic System, Logistic Chaotic System, Stack.*

1. Introduction

The data security is highly considered day after day, given the growth of internet global network and its influence on all aspects of the human life. The internet is used as a platform for transferring our data. Consequently, the protection and security of this data becomes more and more crucial if it is valuable for us. In electronic exchanges, which are increasing day by day, some trustworthy mechanisms are required. These mechanisms should be designed so adequately and accurately that would make the individuals trust the other individual with whom they are communicating. Moreover, it should make the individuals trust the platform that is used for communication. Preserving information security is the foundation of this trust.

Data encryption is one of the best ways to maintain security. Image encryption is one of the main sorts of encryption areas. So far, several algorithms have been developed for this purpose. Among these methods, the application of chaos for cryptography is highly considered due to its specific features [1]. Chaotic systems are considered as ideal methods for cryptography due to their unique characteristics including sensitivity to initial values, pseudo-randomness, unpredictability, non-periodic, etc. For this reason, the applications of these systems are increasing day by day.

Many encryption algorithms have been proposed based on chaos. In these algorithms, the features of chaotic systems are specially applied in image encryption. In these methods, the pixel values for the image are somehow combined using the numbers generated by the chaotic system. Then the encrypted image is obtained by this combination. Meanwhile, the application of various chaotic functions with different dimensions or the application of a combination of several chaotic functions is very common [2-6]. In [7-9], DNA encoding has been combined with chaotic map for encryption.

In many papers [10-17], the plain-image has first been divided into several sub-images, and then the position of each sub-image is changed pseudo-randomly according to a chaotic map.

A stack-based chaotic algorithm is proposed in this paper for encryption of colored images, different from other research works suggested so far. First of all, a series of data is generated in the range of [0, 3] using the chaotic logistic system. Then the original image is divided into four equal sub-images, and, based on the next number in the produced series, four parts of the image are, respectively, pushed into the stack. In the next step, the first element in the stack, which includes one of the four sub-images, is popped, and this image is divided into four other parts. Then, based on the next number in the series, four sub-images are pushed into the stack again. This procedure is repeated every time until the stack is empty. Thus during this process, every time we reached a pixel unit, we encrypted it with the help of another series of chaotic numbers (generated by the Chen chaotic system). This method was repeated until all pixels of the image were encrypted.

This paper is organized as follows. Section 2 provides a brief description about chaotic systems. Section 3 explains about two fundamental data structures, stack, and queue. It also describes our proposed encryption algorithm in details. Simulation outcomes as well as different security analyses are given in section 4, and finally, section 5 concludes the paper and suggests some future improvements.

2. Chaotic systems

Chaos is a phenomenon that occurs in non-linear definable systems. These systems show a high sensitivity to initial conditions. Moreover, they represent a pseudo-random behavior. Such systems remain stable in the chaotic mode when they meet the Lyapunov exponential equation conditions. The important characteristics of the chaos that make it suitable for encryption purposes are its definable nature along with its pseudo-random behavior. This leads to a condition in which the output of the system seems random from the attackers' viewpoint, while it seems definable from deciphers' viewpoint. As a result, such a system seems very simple and decipherable from deciphers' viewpoint. So many chaotic systems have been introduced so far, which can be classified into two main categories from one viewpoint. The first category includes chaotic systems with specific physical interpretation. These kinds of systems have also been derived using the dynamic equations of real systems such as the Lorenz chaotic system [18]. The Lorenz chaotic system was the first continuous chaotic system studied. The second category includes the chaotic systems that do not have any physical interpretation. They are only unique mathematical models. In fact, this category of chaotic systems is used as an assessment indicator in the chaos control and synchronization issues. For example, the Chen chaotic system [19] is a good representative of the chaotic system category. It was first presented by Chen and Eta in 1999. The equations governing this system are provided in the following equation:

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = (c - a)x - xz + cy \\ \dot{z} = xy - bz \end{cases} \quad (1)$$

In this equation, a, b, and c are parameters. If a = 35, b = 3, and c = [20, 28.4], then the system is in a chaos state. The chaotic behavior of this system is shown in figure 1.

Logistic system is a simple chaotic system. It is perhaps the simplest example of a dynamic discrete chaotic system that shows a chaotic behavior. The governing equation of this system is as follows:

$$x_{n+1} = \lambda x_n (1 - x_n), x_0 \in (0,1) \quad (2)$$

In this equation, x_0 is the initial value, and λ is the system parameter in the interval [3.57, 4].

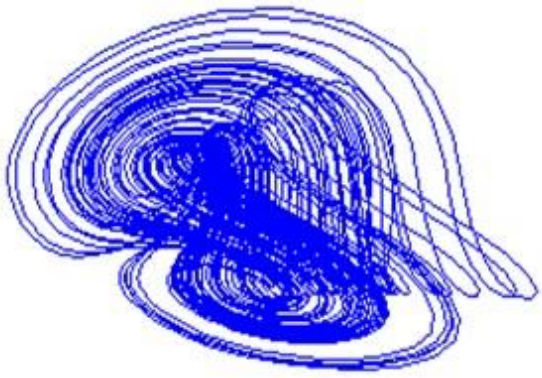

Figure 1. Chaotic behavior of Chen chaotic system.

3. Proposed encryption algorithm

3.1. Stack and queue concepts

Stack and queue are two important and popular data structures in computer science, which are used to solve a variety of data structure problems.

A queue is defined as a list whose elements enter it from one side, named as queue rear, and exit it from the front of the queue. The queue is also known as a FIFO (First in First Out) list because

the first element in a queue is the first element that will be served (exited) from the queue [20]. In other words, the order of entry of the elements into the queue is the same as the order of elements leaving the queue. A sample queue is demonstrated in figure 2, and the adding and removing procedures are also illustrated there.

A stack is a list of elements in which each one of the elements can only be added to the list or removed from it from one side (top of the stack). In other words, the elements are removed from the stack in the inverse order they enter the stack [20]. Two basic actions in the stack include PUSH action, in order to add an element to the stack, and POP action, in order to remove an element from the stack. The stack is also known as a LIFO (Last in First Out) list because the last element that enters the stack is the first element that exits it. This data structure and also its basic procedures are demonstrated in figure 3.

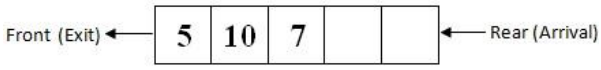

Figure 2. A sample queue as well as removing and adding items.

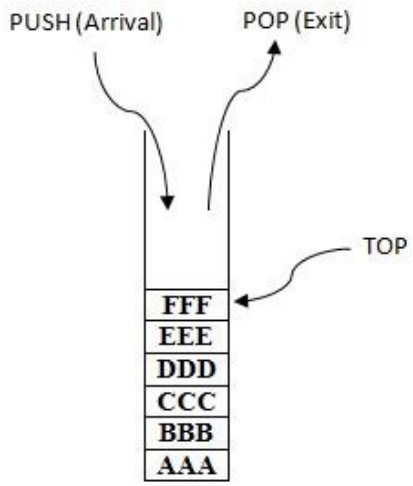

Figure 2. Stack data structure along with PUSH and POP actions.

Referring to figure 3, the six elements AAA, BBB, CCC, DDD, EEE, and FFF are, respectively, pushed in an empty stack. The main procedures of the stack that add and remove the elements can only happen from the top of this data structure. In other words, EEE cannot be removed from the stack before FFF is removed, and DDD cannot be removed from the stack before EEE and FFF are removed, and so forth. As a result, the elements in the list can only be removed or popped in the inverse order they were added or pushed into the stack.

3.2. Stack-based chaotic algorithm for image encryption

The proposed algorithm is a chaotic one, which uses the stack data structure to encrypt the colored images. The algorithm steps are as follow:

Step 1: An empty stack is created with a size exactly the same as the original image. The chaotic logistic system is set to an appropriate initial value. Then a chaotic series of L values is generated after M0 times of execution.

Step 2: X1 series is created from L series using the following equation:

$$x1 = rem(floor((abs(L(i)) * (10^{14}))), 4) \qquad (3)$$

In this equation, rem(L) returns an integer residual value, floor(L) returns the integer part of the number L, and abs(L) returns the absolute value of L. As a result, X1 series includes the numbers 0, 1, 2, and 3, respectively.

Step 3: The image is divided into 4 equal sub-images, according to figure 4, and the K1 counter is set to M0.

Step 4: First, one part of the image is selected based on the first number in the X1 series. Then the other three parts and this selected part are saved in the stack in a clockwise order. In fact, corresponding rows and columns of the pixels are saved. For example, if the first number of the series was 2, then 3, 0, 1, and 2 are, respectively, saved in the stack. Therefore, the top element of the stack is 2. Finally, one unit is added to the K1 counter.

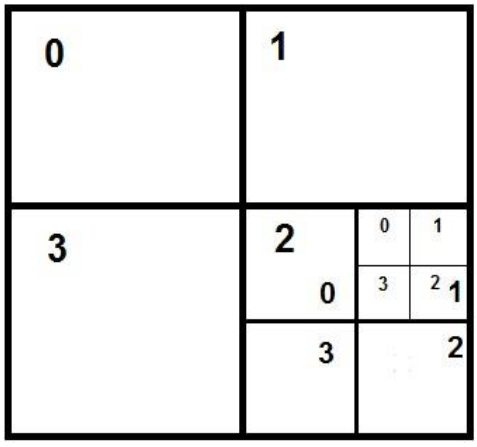

Figure 4. Division of image blocks.

Step 5: The following loop is repeated until the stack is empty:

5.1. The first (top) element is popped from the stack.

5.2. If this element only includes one pixel,

the encryption sub-program is called, this pixel is encrypted, and the control is turned back to the beginning of the loop. If this element does not include one pixel, the popped element is divided into 4 equal sub-images according to figure 4. Then, based on the next number in the X1 series (K1th number in the X1 series), the 4 obtained sub-images are pushed into the stack in a clockwise order. At last, one unit is added to the K1 counter. Therefore, the encryption of image pixels becomes more complicated with the help of stack data structure, and, as a result, it is also difficult to find which pixel is encrypted at what step. In order to encrypt each pixel in the original image, the remaining steps are as follow:

Step 6: First, the Chen chaotic system is set to the initial appropriate values. Then three series named x, y, and z are created.

Step 7: X2, X3, and X4 series are created, respectively, according to (4) from x, y, and z series after N0 times of execution. Then the K2 counter is set to N0.

$$x2 = rem(floor[(abs(x(i)) - floor(abs(x(i)))) * 10^{14}], 256) \quad (4)$$

Therefore, all numbers in the X2, X3, and X4 series are in the range of 0-255.

Step 8: Finally, the following steps are done in order to encrypt each pixel in the image (encryption sub-program):

8.1. The range of pixel values is between 0 and 255 in colored images. The pixel values include the red, green, and blue colors. Therefore, these pixel values and K2th elements of the X2, X3, and X4 series are initially converted to binary forms.

8.2. XOR operation is applied between the K2th element of the X2 series and the red value of pixel, between the K2th element of the X3 series and the green value of the pixel, and finally, between the K2th element of the X4 series and the blue value of the pixel. These three obtained binary numbers are then converted to decimal numbers. These new numbers are considered as new values of the red, green, and blue pixels.

8.3. One unit is added to the K2 counter.

In order to decrypt the image, the above-mentioned steps should be performed on an encrypted image. All the steps are shown in figure 5.

4. Simulation results

A good encryption algorithm should produce cipher-images such that they have noticeable differences with their corresponding plain-images from statistical viewpoints. It should also resist all kinds of known attacks. Thus different statistical experiments were conducted to show the efficiency of our proposed encryption algorithm.

The simulations were all implemented using the MATLAB 7.10 software, and the popular USC data base was used as a benchmark for our experiments. The initial value for the logistic system was set to 0.75798, the λ value was set to 3.85, and the Chen system parameters a, b, and c were set to 35, 3, and 26, respectively. Furthermore, the initial values for the x, y, and z series were set to 1, –1.6, and –0.2, respectively. Finally, M0 was set to 2000, while N0 was set to 3000 (the values of series were not used before the M0 and N0 repeats).

In figures 6, 7, 8, and 9, two sample 256 * 256 images are shown along with the histograms of red, green, and blue channels. Moreover, the encrypted image and its histograms of red, green, and blue channels were presented. As it is clear in this picture, the proposed encryption algorithm has encrypted the image appropriately since the image histogram is completely flat.

4.1. Key space analysis

In the proposed algorithm, the Chen and logistic systems were used in the encryption process. The key values were as follow: the initial values for the logistic and Chen systems that needed a 128-bit space and a 32-bit space for storing λ value of logistic system. Furthermore, a, b, and c values in the Chen system needed a 48-bit space, and finally, the N0 and M0 values needed a 64-bit space. Putting all of these values together, the total number of bits needed for storing all the parameters was 272. Therefore, the cryptosystem provided 2^{272} different combinations, and had a large key space.

The key space of the proposed encryption method was compared with some other competitive algorithms, and the results obtained were presented in table 1. Referring to this table, it is clear that the stack based chaotic algorithm has the largest key space in contrast with the others.

4.2. Key sensitivity test

Several key sensitivity tests were performed in this work. Figures 7, 10, and 11 illustrate the sensitivity of our encryption method to the secret keys L, λ, *a*, *b*, *c*, *x*, y, z, N0, and M0.

Figure 7 is the encrypted image of figure 6 with the actual parameters (cited in section 4). Figure 10 is the encryption result of figure 6 with all the

parameters equal to actual ones, except a = 35.000001, and, finally, figure 11 is the encryption result of the same image with all the parameters equal to the actual ones except λ = 3.85000001.

The histograms of the encrypted images with different initial values are also given in figures 10 and 11.

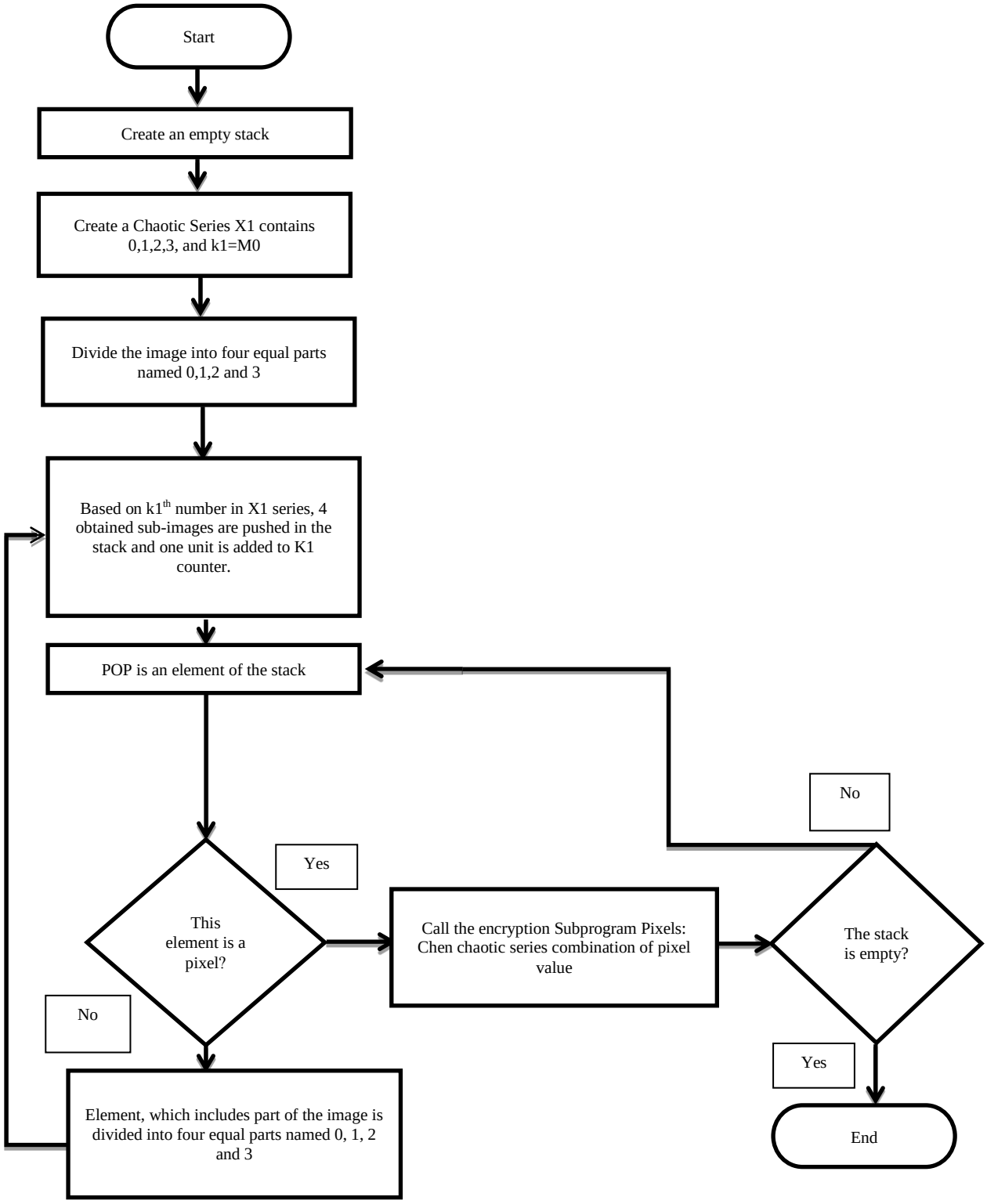

Figure 5 .Proposed encryption algorithms using stack.

Referring to the aforementioned images, it can be concluded that the new chaotic encryption algorithm is sensitive to initial keys such that a small change in their values will generate a completely different decryption result, and, as a result, the plain image cannot be retrieved correctly**.**

4.3. Similarity of adjacent pixels

Usually, in most parts of an image, the neighbor pixel values are very similar to each other. Therefore, a high level of similarity between neighbor pixels is expected in each image. A good encryption algorithm should decrease this level of similarity in order to minimize the possibility of deciphering the pixel values by comparing the neighbor pixel values [10]. For this reason, 3000 pairs (horizontal, vertical, and diagonal) of the neighbor pixels were selected randomly from both the original ciphered images.

The levels of similarity between these pairs are shown in figure 12 (only red channel was used for our comparisons). As it can be seen, the level of similarity is high in the plain-image, while it decreases in the ciphered one using the proposed encryption algorithm.

Table 1. Comparison of key space between our proposed method and some other references.

Ref.	[7]	[9]	[17]	Our proposed algorithm
Key Space	192	233	240	272

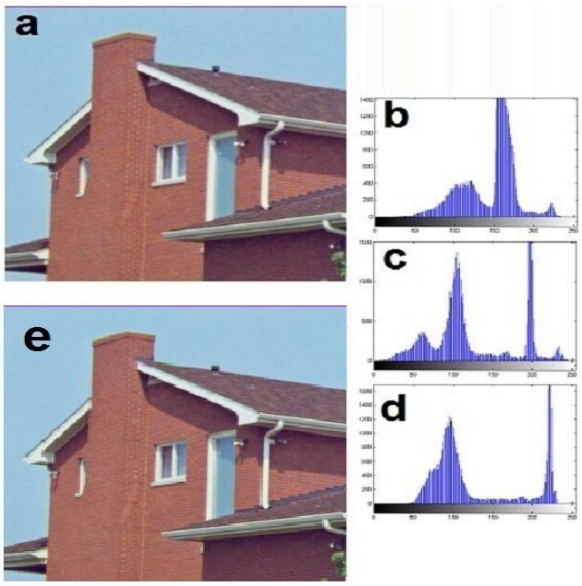

Figure 6. Original image. (b), (c), and (d) are histograms of red, green, and blue channels in original image. (e) Decrypted image.

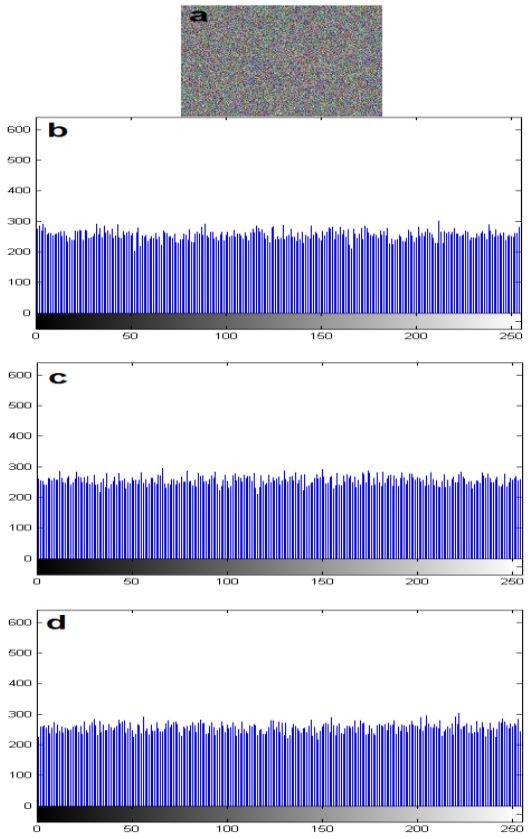

Figure 7. (a) Encrypted image of Fig. 6. (b), (c), and (d) are histograms of red, green, and blue channels.

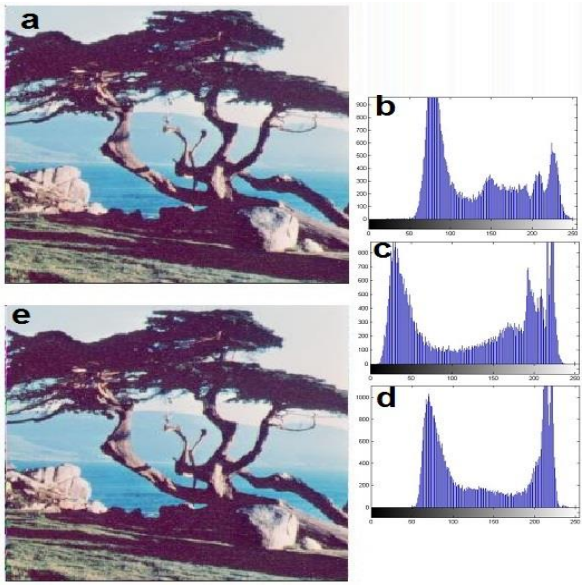

Figure 8. (a) Original image. (b), (c), and (d) are histograms of red, green, and blue channels in original image. (e) Decrypted image.

4.4. Analysis of information entropy

Entropy is one of the most outstanding features that make the images to have a random-like behavior. This parameter was first introduced by Claude E. Shannonin (1949), and can be obtained according to (5).

$$H(S) = \sum_{i=0}^{2^N-1} P(S_i) \log\left(\frac{1}{P(S_i)}\right) \quad (5)$$

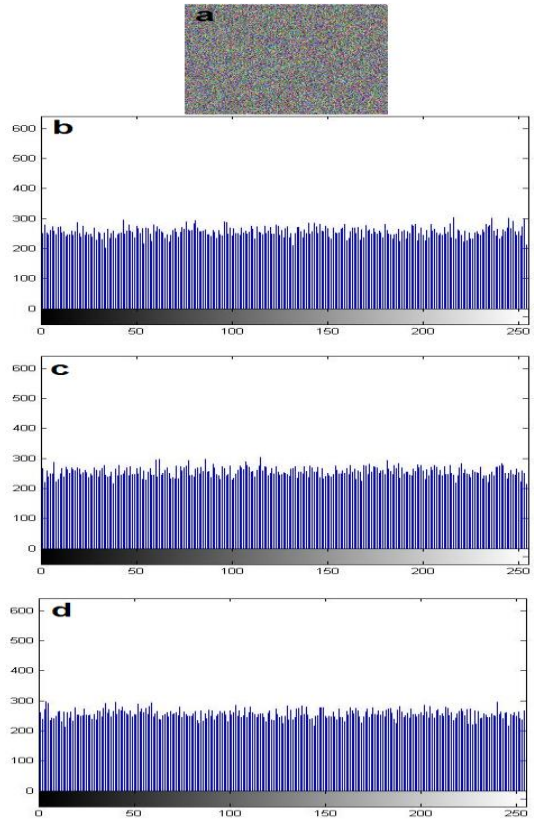

Figure 9. (a) Encrypted image of figure 8. (b), (c), and (d) are histograms of red, green, and blue channels.

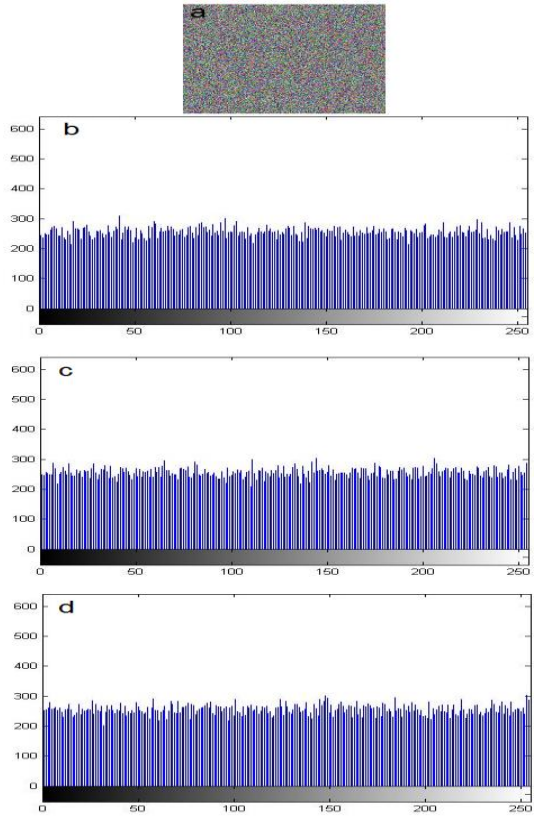

Figure 10. (a) Encrypted image with different initial values (a = 35.000001). (b), (c), and (d) represent histograms of red, green, and blue channels of encrypted image with different initial values.

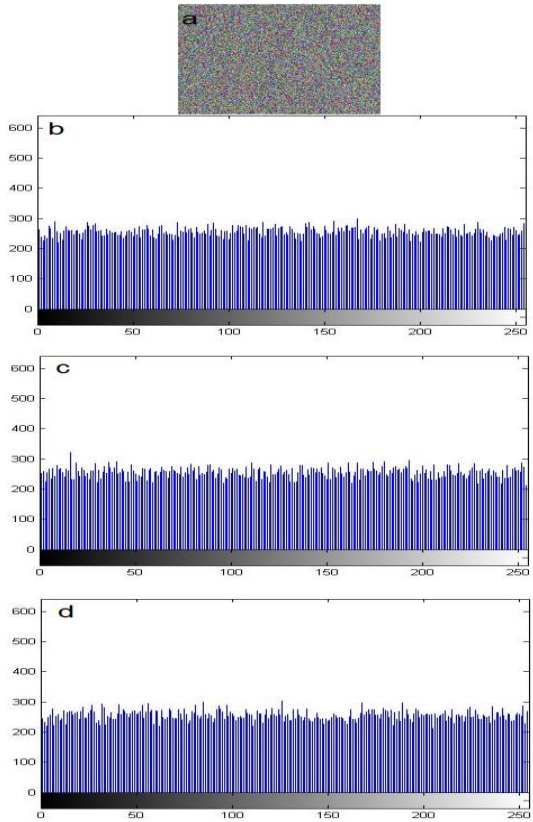

Figure 11. Encrypted image with different initial values (λ = 3.85000001). (b), (c), and (d) represent histograms of red, green, and blue channels of encrypted image with different initial values.

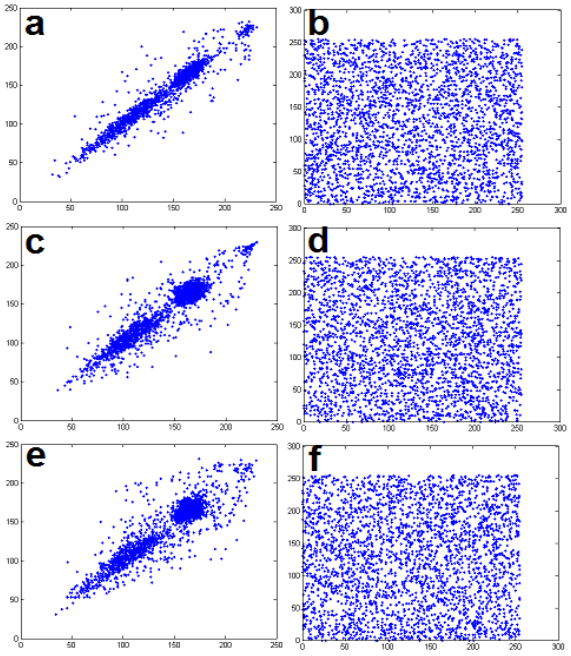

Figure 12. Levels of similarity between two neighbor pixels in red channel of both original and encrypted images from top to bottom in horizontal, vertical, and diagonal.

In this equation, N is the number of grey scale levels in an image (ex: N = 256 for 8-bit image pixels), and P(si) is the occurrence probability of the grey scale "I" in the image. The entropy value

is 8 for images that are produced totally randomly. The entropy value for the proposed encryption algorithm were measured for a sample image, and the result obtained was shown in table 2.

Table 2. Entropy of original image and its corresponding encrypted one.

Entropy of plain-image	Entropy of cipher-image
7.0686	7.9989

From table 2, it can be well-understood that the information entropy of the cyphered image is very close to 8. This means that the encrypted images are close to a random source, and that the proposed algorithm is secure against the entropy attack.

Moreover, the information entropy of the proposed encryption method was compared with some other algorithms for the LENA standard image, and the results obtained were all gathered in table 3. As it can be seen, the entropy of our method is higher than the other four competing algorithms. High-entropic distribution implies that an adversary, given the cipher image, is unable or hardly able to compute any predicate on the cipher image with greater probability than an adversary that does not possess it, and, as a result, it is more secure.

Table 3. Comparison of information entropy between our proposed method and some other references.

Ref.	[7]	[16]	[9]	Our proposed algorithm
Information Entropy	7.9890	7.9977	7.9967	7.9991

5. Conclusions and future works

In this paper, a stack-based chaotic algorithm was introduced for the encryption of colored images. In the proposed method, the series generated from the Chen chaotic system were combined with image pixels after initial pre-processing. Then the encrypted image was generated by this combination. The order of pixels encryption was determined by another series of chaotic numbers generated from the logistic system and also the entry/exit mode of either one pixel or part of the image to/from the stack. The experimental results show that the proposed algorithm has a high security, and also a large key space.

Moreover, the following suggestions can be considered as the future extensions of our research work:

1. Some other data structures such as queues and trees can be used, and some other ways can be applied for their composition (e.g. using two or more stacks) to enhance the security of encryption algorithm.
2. Other chaotic systems or a combination of different chaotic systems can be incorporated to our encryption method.

References

[1] Amigo, J. M., Kocarev, L. & Szczepanski, J. (2007). Theory and practice of chaotic cryptography. Physics Letters A, vol. 366, no. 3, pp. 211-216.

[2] Tong, X., Liu, Y., Zhang, M. & Wang, W. (2012). A Novel Image Encryption Scheme Based on Dynamical Multiple Chaos and Baker Map. 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science, Guilin, China, 2012.

[3] Maksuanpan, S., Veerawadtanapong, T. & San-Um, W. (2014). Robust Digital Image Cryptosystem Based on Nonlinear Dynamics of Compound Sine and Cosine Chaotic Maps for Private Data Protection. ICACT Transactions on Advanced Communications Technology (TACT), vol. 3, no. 2, pp. 418-425.

[4] Gao, H. J., Zhang, Y. S., Liang, S. Y. & Li, D.Q. (2006). A new chaotic algorithm for image encryption. Chaos, Solitons & Fractals, vol. 29, no. 2, pp. 393–399.

[5] Zuo, Y. Z. F., Zhai, Z. & Xiaobin, C. (2008). A New Image Encryption Algorithm Based on Multiple Chaos System. International Symposium on Electronic Commerce and Security, Guangzhou, China, 2008.

[6] Paul, A., Das, N., Prusty, A. K. & Das, C. (2013). RGB Image Encryption by Using Discrete Log with and Lorenz's Chaotic Function. IEEE, 4th International Conference on Computer and Communication Technology (ICCCT), Allahabad, India, 2013.

[7] Liu, L., Zhang, Q. & Wei, X. (2012). A RGB image encryption algorithm based on DNA encoding and chaos map. Computers and Electrical Engineering, Elsevier, vol. 38, pp. 1240-1248.

[8] Som, S., Kotal, A., Chatterjee, A., Dey, S. & Palit, S. (2013). A Colour Image Encryption Based On DNA Coding and Chaotic Sequences. IEEE, 1st International Conference on Emerging Trends and Applications in Computer Science (ICETACS), Shillong, India, 2013.

[9] Wei, X., Guo, L., Zhang, Q., Zhang, J. & Lian, S. (2012). A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system. The Journal of Systems and Software, vol. 85, no. 2, pp. 290-299.

[10] Mirzaei, O., Yaghoobi, M. & Irani, H. (2011). A new image encryption method: parallel sub-image encryption with hyper chaos. Nonlinear Dynamics, vol. 67, no. 1, pp. 557-566.

[11] Jiang, H. & Fu, C. (2008). An Image Encryption Scheme Based on Lorenz Chaos System. IEEE, Fourth International Conference on Natural Computation (ICNC), Jinan, China, 2008.

[12] Dongming, C., Zhiliang, Z. & Guangming, Y. (2008). An improved Image Encryption Algorithm

Based on Chaos. IEEE The 9th International Conference for Young Computer Scientists, Hunan, China, 2008.

[13] Wang, Y., Wong, K-W., Liao, X. & Chen, G. (2011). A new chaos-based fast image encryption algorithm. Applied Soft Computing, vol. 11, no. 1, pp. 514–522.

[14] Ye, G. (2010). Image Scrambling encryption algorithm of pixel bit based on Chaos map. pattern recognition letters, vol. 31, pp. 347-354.

[15] Zhu, Z. L., Zhang, W., Wong, K. W. & Yu, H. (2011). A chaos-based symmetric image encryption scheme using a bit-level permutation. Information Sciences, Vol. 181, pp. 1171–1186.

[16] Murillo-Escobar, M. A., Cruz-Hernández, C., Abundiz-Pérez, F., López-Gutiérrez, R. M. & Acosta Del Campo, O.R. (2015). A RGB image encryption algorithm based on total plain image characteristics and chaos. Signal Processing, vol. 109, pp.119-131.

[17] Mazloom, S. & Eftekhari-Moghadam, A. M. (2009). Color image encryption based on Coupled Nonlinear Chaotic Map. Chaos, Solitons & Fractals, vol. 42, no. 3, pp. 1745–1754.

[18] Lorenz, E. N. (1963). Deterministic nonperiodic flow. Journal, J. Atmos. Sci, 1963, vol. 20, no. 2, pp. 130-141.

[19] Chen, G. & Ueta, T. (1999). Yet another chaotic attractor. Journal, Int. J. Bifur. Chaos, vol. 9, no. 7, pp. 1465-1466.

[20] Seymour, L. (1986). Schaum's Outline of Theory and Problems of Data Structures. McGRAW-HILL BOOK Company.

نشریه هوش مصنوعی و داده کاوی

روشی جدید به منظور رمزنگاری تصاویر رنگی بر مبنای آشوب و به کمک پشته

حبیب خدادادی[1,*] و امید میرزایی[2]

[1] گروه کامپیوتر، واحد میناب، دانشگاه آزاد اسلامی، میناب، ایران .

[2] آزمایشگاه امنیت کامپیوتر، دپارتمان علوم و مهندسی کامپیوتر، دانشگاه کارلوس سوم، مادرید، اسپانیا.

ارسال ۲۰۱۵/۰۷/۰۴؛ پذیرش ۲۰۱۶/۰۸/۱۸

چکیده:

در این مقاله، روش جدیدی به منظور رمزنگاری تصاویر رنگی ارائه شده است. اساس این روش بر مبنای استفاده از ساختمان داده پشته و آشوب است که رمزنگاری تصویر را کاراتر و مقاوم‌تر نموده است. در این روش ابتدا یک سری از اعداد بین ۰ تا ۳ را به کمک سیستم آشوبناک لاجستیک تولید کرده و سپس هر بار تصویر به چهار قسمت تقسیم کرده و بر حسب عدد بعدی سری، به ترتیب چهار قسمت تصویر را در یک پشته Push می‌کنیم. در مرحله بعد عنصر اول پشته (که شامل یکی از ۴ قسمت تصویر است) POP شده و آن قسمت به ۴ قسمت دیگر تقسیم شده و دوباره و بر حسب عدد بعدی سری به ترتیب ۴ زیر تصویر را در پشته Push می‌کنیم. این عمل را هر بار تکرار می‌کنیم تا پشته خالی شود. در طی این فرآیند هرگاه به یک پیکسل واحد رسیدیم آن را به کمک سری دیگری از اعداد آشوبگون (تولید شده توسط سیستم آشوبناک چن) رمزنگاری می‌کنیم. این روش تا رمزنگاری تمام پیکسل‌های تصویر ادامه داده می‌شود. آزمایشاتی که بر روی تصاویر استاندارد انجام شده است نشان‌دهنده کارا و مقاوم بودن این روش رمزنگاری می‌باشد.

کلمات کلیدی: آشوب، پشته، رمزنگاری تصاویر رنگی، سیستم کیاتیک چن، سیستم کیاتیک لاجستیک.