Proceedings of International Mathematical Sciences
URL: https://dergipark.org.tr/tr/pub/pims
Volume I Issue 2 (2019), Pages 41-50.

A BLOCK-BASED IMAGE ENCRYPTION SCHEME USING CELLULAR AUTOMATA WITH AUTHENTICATION CAPABILITY

SAEIDEH KABIRIRAD * AND ZIBA ESLAMI**

*DEPARTMENT OF COMPUTER SCIENCE, BIRJAND UNIVERSITY OF TECHNOLOGY, BIRJAND, IRAN

**(CORRESPONDING AUTHOR) DEPARTMENT OF DATA AND COMPUTER SCIENCE, SHAHID BEHESHTI UNIVERSITY, G.C., TEHRAN, IRAN

Abstract. In this paper, we employ a combination of chaotic maps and Cellular automata(CA) to propose a secure block-based image encryption scheme with authentication capability. The authentication mechanism incorporated into the presented scheme can detect slight tampering in the cipher image before full decryption. We use chaotic maps to produce pseudo-random sequences and apply a CA to diffuse the pixel values efficiently. Many of the existing image encryption schemes fall short of providing parallel processing capability and high sensitivity to changes simultaneously. This study tries to provide both capabilities together. Furthermore, our proposed authentication mechanism, while preventing exploitation in brute-search attacks, can be adjusted to any desired level of security. Finally, theoretical analysis and experimental results together with comparisons with related literature indicate that the proposed scheme has a high robustness against common security attacks.

1. Introduction

One of the most widely used systems in the image encryption is chaos systems. Some of the special features of chaos systems include their dependency to initial conditions and control parameters and their ergodic characteristics. Therefore, their application in cryptography provides high complexity, random-like behavior, diffusion and thus better security. Researchers have proposed different chaos based image encryptions in [1, 2, 3]. Overall, to resist common attacks, these schemes have two phases: confusion and diffusion [4]. In the confusion phase plain image is disturbed that the correlation between two adjacent pixels is extremely low and in the diffusion phase the pixel values are altered so that the influence of individual plain image is spread out over the cipher image as much as possible [5]. Furthermore considering the security analysis such as [6], it can be concluded that the encryption key sequence must also be related to the image pixels, otherwise it will be susceptible

2010 *Mathematics Subject Classification.* Primary: 68P25 ; Secondaries: 94A60 .
Key words and phrases. Image Encryption; Cellular Automata; Chaotic Map; Authentication.

to attacks such as known or chosen plain-text attacks. Also, these methods might not be sufficiently sensitive to plain image.

For more computational efficiency, some researchers use tools such as cellular automata (CA) [7, 8, 9] and linear feedback shift registers [10]. CA are discrete dynamical system formed by a finite array of identical objects called cells. Each cell is endowed with a state which changes at every time-step depending on the states of its adjacent cells at previous time-steps. This feature provides confusion and therefore makes CA attractive in cryptography. Furthermore, they have parallel computation capability as well as high execution speed.
In the CA-based method presented in [11], permutation and diffusion stages are implemented in two separate phases. In permutation phase, pixels of the plain image are shuffled using a pseudo-random keystream generated by intertwining logistic chaotic map. In diffusion phase a two-dimensional CA is applied to achieve diffusion on bit-level. This method does not have complete parallel execution capability, because its permutation phase must be executed in order. The algorithms presented in [7] and [12] used second-order CA. In both papers, the proposed algorithms act like a block-cipher but in a fully parallel mode. Since the blocks are encrypted independently, slight changes in one block has no effect on other blocks. Ping et al. [8] used a second-order, non-affine balanced and reversible CA to increase confusion and diffusion of the encryption algorithm. Also, an irreversible CA is used to produce a pseudo-random key sequence. The diffusion is performed by the local interaction among cells, while the confusion is achieved by the nonlinear rules applied to cells. The method has a good execution speed but cannot be executed in parallel. In [1], an image encryption based on non-uniform cellular automata and hyper chaotic functions is proposed. First, six chaotic sequences using three-dimensional Arnold mapping are generated after frequent iterations. Then random sequences are used to permute the image in rows and columns. In confusion step, a key image created using CA and a hyper-chaotic mapping are used to encrypt the pixels. Reviewing existing literature clarifies most of them cannot provide simultaneously parallel processing capability and high sensitivity to changes [2, 3].
Also, some of image encryption schemes [13] check the integrity of cipher images using an authentication mechanism such that any tampering in the cipher images can be detected.

In this paper, we use the advantages of chaotic maps and CA to present a secure block-based image encryption scheme. This means that we use chaotic maps to produce pseudo-random sequences and apply a CA to diffuse the pixel values with efficiency in a parallel setting. This study tries to provide both capabilities together. Furthermore we augment the authentication mechanism while also preventing exploitation in brute-search attacks and can be adjusted to any desired level of security. Finally, theoretical analysis and experimental results indicate that the proposed scheme has a high robustness against common security attacks.

2. Preliminaries on Cellular Automata

Cellular automata (CA) are dynamic systems consisting of an array of similar units called cells. A CA is defined in the form of $CA = \{\mathcal{C}, \mathcal{S}, \mathcal{N}, \mathcal{F}\}$ where $\mathcal{C}$ is the cell space, $\mathcal{S}$ is the set of discrete states of cells where in the simplest state $\mathcal{S} \in \{0,1\}$, $\mathcal{N}$ signifies the neighbors of a cell and $\mathcal{F}$ is the transition function, which includes rules for determining the next state of the cell [14]. Assuming that a_i^T describes the state of cell i in time step T, the local transition function of CA is defined as below:

$$a_i^{T+1} = f(\mathcal{N}_i^T),\ 0 \le i \le N-1 \tag{2.1}$$

In the above equation, $\mathcal{N}_i^T$ describes the state of neighbors of cell i in time step T. This formula expresses that the next state of each cell is determined according to a transition function with inputs the current state of itself and its neighbors. States of all cells at time step T form a configuration. In this paper we use a reversible linear memory CA (LMCA). In a LMCA of order t, the next states of the cells depend on t previous configurations which is defined as follows:

$$a_i^{T+1} = f_1(\mathcal{N}_i^T) + f_2(\mathcal{N}_i^{T-1}) + \cdots + f_t(\mathcal{N}_i^{T-t+1}) \tag{2.2}$$

In the above equation, $f_i,\ i = 1, \ldots, t$ are local transition functions in LMCA. If $f_t(\mathcal{N}_i^{T-t+1}) = a_i^{T-t+1}$, then LMCA is reversible through the said transition function and its inverse is another LMCA with the following local transition function:

$$a_i^{T+1} = \sum_{m=0}^{t-2} f_{t-m-1}(\mathcal{N}_i^{T-m}) + a_i^{T-t+1},\ 0 \le i \le N-1 \tag{2.3}$$

3. The proposed method

To provide parallel processing capability, we first divide the image into multiple blocks. Then each block is encrypted using an m-order LMCA. A permutation algorithm is performed in block level to eliminate the relation between consecutive blocks. This approach increases the execution speed in comparison to pixel or bit permutation. Finally, second pass of the m-order LMCA with new transition rules is performed. To increase the complexity of the method, LMCA rules are determined by a pseudo-random sequence. We generate the pseudo-random sequence using logistic chaotic map defined according to the following formula:

$$x_i = \alpha x_{i-1}\ (1 - x_{i-1}),\ x_i \in [0, 1] \tag{3.1}$$

where α is the logistic map parameter. The output sequence is chaotic when $\alpha \in [3.57, 4]$.

Since each block is encrypted independently, the values of its pixels have no effect on other encrypted blocks. To create such effect, the hashed value of an interleaved image is calculated and is incorporated in the encryption process. This value is defined such that it is used in authentication as well as encryption.

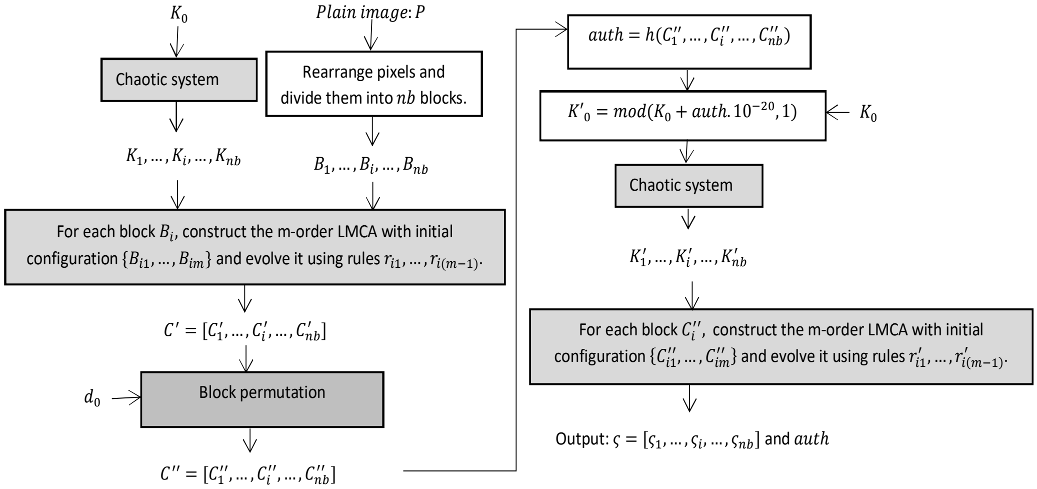
In the following, image encryption and decryption algorithms are described in detail. Also, graphical representation of the encryption algorithm is depicted in Figure 1.

3.1. Encryption algorithm.

We assume that plain image P is an image of size $M \times N$. Suppose that number of blocks is nb and size of each block is $m \times n$. The encryption process includes the following steps:

(1) Consider the pixels of image P in the form of array $P = [P_0, P_{nb}, P_{2nb}, \ldots, P_1, P_{nb+1}, \ldots, P_{nb-1}, \ldots, P_{MN}]$ where the image pixels are numbered from top to bottom and left to right. Then, starting from the beginning of the array, we put each mn pixels into a separate block of size $m \times n$. With this simple reorganization, adjacent pixels will probably not be in a block. Assume that blocks are represented by $B_1, B_2, \ldots, B_{nb}$.

Figure 1. Diagram of the encryption procedure


K_0
Plain image: P
Chaotic system
Rearrange pixels and divide them into nb blocks.
$K_1, \ldots, K_i, \ldots, K_{nb}$
$B_1, \ldots, B_i, \ldots, B_{nb}$
For each block B_i, construct the m-order LMCA with initial configuration $\{B_{i1}, \ldots, B_{im}\}$ and evolve it using rules $r_{i1}, \ldots, r_{i(m-1)}$.
$C' = [C_1', \ldots, C_i', \ldots, C_{nb}']$
d_0
Block permutation
$C'' = [C_1'', \ldots, C_i'', \ldots, C_{nb}'']$
$auth = h(C_1'', \ldots, C_i'', \ldots, C_{nb}'')$
$K'_0 = mod(K_0 + auth.10^{-20}, 1)$
K_0
Chaotic system
$K_1', \ldots, K_i', \ldots, K_{nb}'$
For each block C_i'', construct the m-order LMCA with initial configuration $\{C_{i1}'', \ldots, C_{im}''\}$ and evolve it using rules $r_{i1}', \ldots, r_{i(m-1)}'$.
Output: $\varsigma = [\varsigma_1, \ldots, \varsigma_i, \ldots, \varsigma_{nb}]$ and $auth$

(2) Consider values K_0 and d_0 as the encryption keys. K_0 is the initial value for generating the LMCA rules and d_0 is the initial value for generating the block permutation sequence.

(3) Repeat logistic chaotic map, starting from initial value K_0 for producing nb numbers $[K_1, K_2, \ldots, K_{nb}]$. Each of these numbers will be used for producing the LMCA rules in a block.

(4) Then for each block $B_i, i = 1, 2, \ldots, n$, perform the following:
   (a) Construct the m-order LMCA with initial configuration $C_i^0 = \{B_{i1}, B_{i2}, \ldots, B_{im}\}$, where B_{ij} is the j-th row in i-th block.
   (b) Produce a set of rules $R_i = \{r_{i1}, r_{i2}, \ldots, r_{i(m-1)}\}$ using initial value K_i such that: $[r_{i1}, r_{i2}, \ldots, r_{i(m-1)}] = f(K_i)$, where f is a function that assigns a random bit sequence to the LMCA rules.
   (c) Evolve the LMCA at least m times starting from the initial configuration to obtain the final configuration $C_i^{'} = \{C_{i1}^{'}, C_{i2}^{'}, \ldots, C_{im}^{'}\}$.

(5) In this step, a permutation is conducted on the image blocks using key d_0 to obtain image $C^{''}$.
   (a) First, produce the initial value through $d_0^{'} = mod\ (d_0 + (C_1^{'} \oplus \ldots \oplus C_{nb}^{'})10^{-20}, 1)$, where $\oplus$ is XOR operation and $mod(x, 1)$ is the decimal part of x.
   (b) Generate the sequence $d_1, d_2, \ldots, d_{nb}$ using the logistic chaotic map with initial value d_0.
   (c) Compute the permutation sequence y such that: $y_i = [d_i \times nb] + 1$
   (d) The set $y_1, y_2, \ldots, y_{nb}$ is sorted increasingly and a new set $y_1^{'}, y_2^{'}, \ldots, y_{nb}^{'}$ is obtained.
   (e) Determine the new position of each block according to sequence $y^{'}$; to do so, find the position of values $y_1^{'}, y_2^{'}, \ldots, y_{nb}^{'}$ in $y_1, y_2, \ldots, y_{nb}$ and

the position set $S = \{pos_1, pos_2, \dots, pos_{nb}\}$ is formed, where $y_i^{'}$ is the value of y_{pos_i}.

(6) Calculate hashed value $auth$ using the permutated image $C^{''}$ as: $auth = h(C_1^{''}, C_2^{''}, \dots, C_{nb}^{''})$, where $h : Z_{256}^{*} \rightarrow Z_{256}^{l}$ is a collision resistant hash function. The value of l is optional and depends on the desired authentication strength.

(7) Produce new initial value $K_0^{'}$ for generating the rules of the second pass of the LMCA as:
$K_0^{'} = mod(K_0 + 10^{-20} auth, 1)$, where $mod(x, 1)$ is the decimal part of x.

(8) Produce the new sequence $K_1^{'}, K_2^{'}, \dots, K_{nb}^{'}$ and local rules $R_i = \{r_{i1}^{'}, r_{i2}^{'}, \dots, r_{i(m-1)}^{'}\}$ just like the Steps 3 and 4.

(9) For each block, execute the LMCA at least m times with initial configuration $C_i^{''(0)} = (C_{i1}^{''}, C_{i2}^{''}, \dots, C_{im}^{''})$ and local rules $r_{i1}^{'}, r_{i2}^{'}, \dots, r_{i(m-1)}^{'}$; In the end, the final configuration $\varsigma_i = \{\varsigma_{i1}, \varsigma_{i2}, \dots, \varsigma_{i(m)}\}$ will be obtained.

The final cipher image is obtained as $CI = \{\varsigma_1, \varsigma_2, \dots, \varsigma_{nb}, auth\}$. Here, $auth$ will be used for producing the key sequence as well as in errors detection.

3.2. **Decryption algorithm.** To decrypt cipher image $CI = \{\varsigma_1, \varsigma_2, \dots, \varsigma_{nb}, auth\}$ with the key (K_0, d_0), the steps of the encryption algorithm must be followed in reverse. For checking the validity of the cipher image, we must calculate value: $H = h(C_1^{''}, C_2^{''}, \dots, C_{nb}^{''})$ after evolving the reverse of LMCA in the first pass. If $H = auth$, the image is authenticated.

4. SECURITY ANALYSIS AND EXPERIMENTAL RESULTS

In this section, the security analysis of the proposed scheme and its experimental results are provided. We implemented the proposed algorithm and evaluated this with various images. For example, we run algorithm with plain images of Figure 2(a),(b) of size 512 × 512 and obtained corresponding cipher images of Figure 2(c),(d).

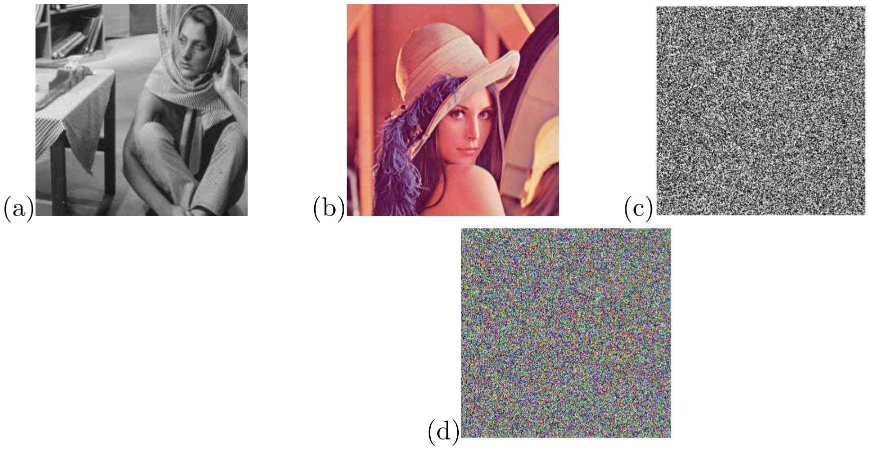


FIGURE 2. (a),(b) Plain images "Barbara" and "Lena"; (c),(d) and corresponding encrypted images.

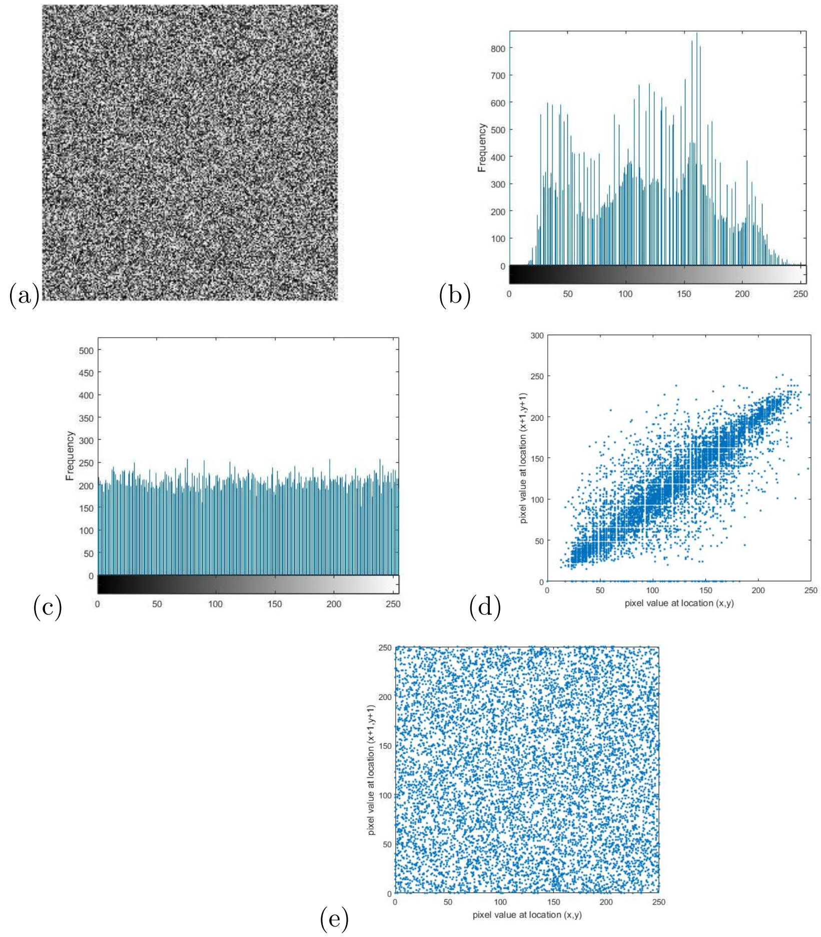


Figure 3. (a) Decrypted image of Barbara by a slightly changed key; (b),(c) Histograms of the plain image "Barbara" and the corresponding cipher image; (d),(e) Plots of the intensity values of adjacent pixels in diagonal direction.

(1) **Chosen plain-image attacks.** In this attack, the attacker has a temporary access to the encryption system and chooses a number of plain-images and obtains their corresponding cipher-images. Then he tries to break the cryptosystem. Without knowing the encryption key, decryption is impossible. Because of Steps 5 and 7 in the encryption algorithm, values of pixels also affect the key sequence; So this sequence will be different for various images. As a result, it is impossible to use a key sequence resulted from a chosen plain-image on a different or even roughly similar image. Therefore this method is resistant against chosen plain-image or known plain-image attacks.

(2) **Key space and sensitivity analysis.** For an encryption scheme to be secure, it must has a sufficiently large key space. In the encryption algorithm, length of key (K_0, d_0) is at least 150 bits, so the key space is at least

2^{150}. As a result, this method is also resistant against brute-force attacks. Experimental results show that this method is sensitive to the key. For example, we have decrypted image of Figure 2(c) with slightly different key value and have obtained image of Figure 3(a). Comparing the recovered image with original one reveals that their pixels are totally different.

(3) **Histogram analysis.** Histogram analysis determines distribution of image pixels. Histogram of the cipher image must be almost uniform and have a significant difference with the histogram of the plain image to prevent the attacker from extracting any meaningful information. Figure 3(b),(c) shows the histograms of the plain and corresponding cipher images.

(4) **Correlation analysis of adjacent pixels.** Pixels of a plain image often have significant dependency, but dependency of cipher image pixels must be negligible. To test the correlation of adjacent pixels in an image, we randomly choose pairs of adjacent pixels and use the following formula to calculate the correlation coefficient in vertical, horizontal and diagonal directions.

$$\sigma_{uv} = \frac{\sum_{i=1}^{N}(u_i - mean(u))(v_i - mean(v))}{\sum_{i=1}^{N}((u_i - mean(u))^2)^{1/2}\sum_{i=1}^{N}((v_i - mean(v))^2)^{1/2}} \tag{4.1}$$

In the above equation, u and v are the arrays of N adjacent pixels. The correlation coefficients of adjacent pixels in some plain and cipher images are summarized in Table 1. Also Figure 3(c),(d) shows plots of the intensity values of 10000 couples of horizontally adjacent pixels in plain image “Barbara” and the corresponding cipher image.

(5) **Differential attack.** In this attack, the attacker makes a slight change in the plain image and compares its cipher image with original one to find a significant relationship. The presented algorithm is robust against this attack, because the LMCA distributes any change in its input on its final output, so a small change in a block of plain image affects the entire block. Furthermore, after the first round of execution of the LMCA, a hashed value of the entire resulting pixels is calculated and is used in the second round. Note that, one of the main features of a hash function is that even a slight change in its inputs makes substantial changes in its output. Thus any small change in a block of the plain image triggers extensive changes in the hashed value, the key sequence and consequently cipher image.

Also, we assess robustness of the proposed scheme against the attack with the use of two measures: NPCR (Number of Pixels Change Rate) and UACI (Unified Averaged Changed Intensity) which are calculated with the following formulas.

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

$$UACI = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|p(i,j) - q(i,j)|}{255} \times 100\%$$

In the above equations, p and q are the sets of pixels of the original and the changed images, respectively. If $p(i,j) = q(i,j)$ then $D(i,j) = 0$ otherwise $D(i,j) = 1$.

We have encrypted several images using the proposed method and calculated the values of NPCR and UACI. A summary of the obtained results are presented in Table 1. The value of NPCR is close to 100% and the value of UACI is close to 33.5%, which indicate the high sensitivity of the proposed scheme to the plain image.

(6) **Information entropy analysis.** The information entropy is the most significant measure of unpredictability of information. The information entropy $H(I)$ of an image I can be calculated by the following formula:

$$H(I) = -\sum_{i=1}^{l} P(I_i) log_2 P(I_i) \tag{4.2}$$

where l is the total number of pixels in I and $p(I_i)$ represents the probability of occurrence of value I_i. For a random image emitting 256 values with the same probability, $H(I)$ reaches the maximum value 8. The entropy values for some ciphers are summarized in the last column of Table 1. It can be observed that they are close to a random source.

Table 1. The correlation coefficients, the average NPCR and UACI and entropy of the proposed scheme.

Image	Horizontal		Vertical		Diagonal		Average	Average	Entropy
	Plain	Cipher	Plain	Cipher	Plain	Cipher	NPCR	UACI	
Barbara	0.9132	0.0091	0.9360	-0.0512	0.9720	0.0990	99.606	33.461	7.9992
Lena	0.895	0.0120	0.9041	0.0074	0.8912	-0.0814	99.611	33.518	7.9993
Cameraman	0.9597	0.0355	0.9381	-0.0019	0.9325	0.0189	99.581	33.521	7.9993
Peppers	0.9655	-0.0027	0.9761	0.0712	0.9428	0.0089	99.594	33.361	7.9994

4.1. **Validity of the authentication algorithm.** The presented scheme has an authentication phase in the decryption algorithm. The authentication is performed on image obtained from evolving the inverse of LMCA in the first round. The value of H is calculated the same as stage 6 of the encryption algorithm and is compared with the value of *auth*. If these values are not equal, the cipher image has errors. The value of *auth* is produced using the intermediate values of the encryption process and is calculated by a collision-resistant hash function, so it will not reveal any information about the plain image. Also, considering the use of hash function to produce *auth* and sufficient complexity of the LMCA, the chance of manipulating the cipher image such that the values of H and *auth* would be equal is very negligible. Therefore, any tampering on the cipher image can be detected.

5. COMPARISON

In the following, we compare the proposed method with the methods presented in [1, 7, 8, 11] in terms of cipher image sensitivity, parallel execution capability, block or string nature of algorithm and etc.

Algorithms in [13, 11] use a stream-cipher scheme in the permutation phase and block-cipher scheme in the diffusion phase; while other assessed algorithms are totally block-based. Therefore they cannot provide full parallel processing capability, since it takes a lot of time to execute the sort algorithm in the permutation phase.

Scheme [8] cannot be executed in parallel efficiently.
In terms of sensitivity to the plain image, algorithms [11, 7] cannot be considered sufficiently sensitive, because in these algorithms changing a pixel does not affect the previous encrypted pixels or the entirety of pixels. In the proposed method and methods [1, 8], the encryption key sequence is affected by pixels, so any slight change in the plain image affects all pixels in the cipher image.
The proposed scheme and [13] can check integrity by adding extra bits in the cipher image. In [13] a fixed number of authentication bits (8 bits per block) is used, while in our scheme this parameter is tunable to any arbitrary value.
In all mentioned methods except the method of [8], the confusion and diffusion stages are applied in two separate phases. None of the compared schemes [8, 7, 1, 13, 11] can ensure sensitivity of the entire cipher-image to the plain-image and at the same time provide parallel execution capability. But the proposed scheme could be executed in parallel while also providing high sensitivity to the plain image. A summary of the results of the comparison is presented in table 2.

Table 2. Comparisons between the proposed scheme and similar schemes

	Our scheme	[13]	[7]	[1]	[8]	[11]
Block-based	✓	✓	✓	✗	✓	✓
Parallel computing	✓	✗	✓	✗	✗	✗
High sensitivity	✓	✗	✗	✓	✓	✗
Authenticated	✓	✓	✗	✗	✗	✗
Simultaneous confusion and diffusion	✗	✗	✓	✗	✓	✗

6. CONCLUSION

This paper proposes an image encryption scheme based on chaotic maps and linear memory cellular automata. In this method, the encryption and decryption of image is block-based. To determine the integrity of the cipher image, an authentication mechanism is provided which can detect manipulation in the image. A prominent point of the proposed method is the combination of parallel execution capability with high sensitivity of cipher image to slight changes in the plain image. Security analysis and experimental results demonstrate that the presented scheme fulfills all the desired properties of a secure cryptosystem including large key space, statistical attacks, differential attack and chosen-plaintext attack.

References

[1] A.Y. Niyat, M. H. Moattar, and M. N. Torshiz, Color image encryption based on hybrid hyper-chaotic system and cellular automata, Optics and Lasers in Engineering **90**, 225-237 (2017).

[2] A. Souyah, and K. M. Faraoun, Fast and efficient randomized encryption scheme for digital images based on quadtree decomposition and reversible memory cellular automata, Nonlinear Dynamics **84**,715-732 (2016).

[3] L. Xu, Z. Li, J. Li, and W. Hua, A novel bit-level image encryption algorithm based on chaotic maps, Optics and Lasers in Engineering **78**, 17-25 (2016).

[4] C. E. Shannon, Communication theory of secrecy systems, Bell system technical journal **28**, 656-715 (1949).

[5] G. Alvarez, and S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, International journal of bifurcation and chaos **16(22)**, 2129-2151 (2006).

[6] H. Liu, and Y. Liu, Cryptanalyzing an image encryption scheme based on hybrid chaotic system and cyclic elliptic curve, Optics & Laser Technology **56**, 15-19 (2014).

[7] K. M. Faraoun, A parallel block-based encryption schema for digital images using reversible cellular automata, Engineering Science and Technology, an International Journal **17**, 85-94 (2014).

[8] P. Ping, F. Xu, and Z. J. Wang, Image encryption based on non-affine and balanced cellular automata, Signal Processing **105**, 419-429 (2014).

[9] B. Jeyaram and R. Raghavan, New cellular automata-based image cryptosystem and a novel non-parametric pixel randomness test, Security and Communication Networks **9**, 3365-3377 (2016).

[10] D. Dey, D. Giri, B. Jana, T. Maitra and R.N. Mohapatra, Linear-feedback shift register-based multi-ant cellular automation and chaotic map-based image encryption, Security and Privacy **1(6)**, 1-11 (2018).

[11] X. Wang and D. Luan, A novel image encryption algorithm using chaos and reversible cellular automata, Communications in Nonlinear Science and Numerical Simulation **18(11)**, 3075-3085 (2013).

[12] K. M. Faraoun, Fast encryption of RGB color digital images using a tweakable cellular automaton based schema, Optics & Laser Technology, **64**, 145-155 (2014).

[13] A.S. Rajput and M.A. Sharma, Novel Image Encryption and Authentication Scheme Using Chaotic Maps, In Advances in Intelligent Informatics, Springer, Cham, 277-286 (2015).

[14] S. Wolfram, Cellular automata and complexity, collected papers. CRC Press (2018).

S. Kabirirad,
Birjand University of Technology, Birjand, Iran, Phone: +985632391144
Email address: `s_kabirirad@sbu.ac.ir`

Z. eslami,
Shahid Beheshti University, Tehran, Iran, Phone: +982129903007
Email address: `z_eslami@sbu.ac.ir`