



**Repositorio Institucional de la Universidad Autónoma de Madrid**

<https://repositorio.uam.es>

Esta es la **versión de autor** del artículo publicado en:  
This is an **author produced version** of a paper published in:

IEEE MultiMedia 19.4 (2012): 28 – 36

**DOI:** <http://dx.doi.org/10.1109/MMUL.2011.54>

**Copyright:** © 2012 IEEE

El acceso a la versión del editor puede requerir la suscripción del recurso  
Access to the published version may require subscription

# Digital Image Scrambling Using 2D Cellular Automata

**Abdel Latif Abu Dalhoum and Basel A. Mahafzah**  
*University of Jordan*

**Aiman Ayyal Awwad**  
*Tafila Technical University, Jordan*

**Ibrahim Aldamari**  
*San'a University, Jordan*

**Alfonso Ortega and Manuel Alfonseca**  
*Universidad Autónoma of Madrid, Spain*

## **ABSTRACT**

***A digital image scrambling method based on a 2D cellular automaton, specifically the well-known Game of Life, produces an effective image encryption technique.***

The rapid development of communication technology due to the global spread of the Internet and the digital information revolution has given rise to a huge increase in the use and transmission of multimedia information (images, audio, and video). As a result, information security during storage and transmission has become a critical issue. For example, images are widely used in industrial processes. These images could contain private information, so they must be protected.

Digital image scrambling, often used for image encryption and data hiding, reorders and changes the position of image pixels to break the relationship between adjacent pixels.<sup>1</sup> These methods include Advanced Encryption Standard,<sup>2</sup> Twice Interval Division,<sup>3</sup> Cat Chaotic Mapping,<sup>4</sup> Magic Cube,<sup>5</sup> and Arnold Transformation.<sup>6</sup> We propose a new scrambling method based on a 2D cellular automaton (CA).

The ability to obtain complex global behavior from simple local rules makes CA an interesting platform for digital image scrambling. The most widely known example, the Game of Life (GL), is a 2D CA that produces large amounts of patterned data. The GL (which was designed by John Conway) can scramble the digital image by providing the complex behavior **that would produce the most useful operations**

In this work, we analyze the GL's complex characteristics using digital image scrambling to decide whether the degree of scrambling is influenced by different GL configurations (such as the number of generations and boundary conditions). We also design various sets of 2D CA rules, variations on the GL rules, with different Lambda parameters around the critical value of Lambda. Our resulting model is simple and robust, and our tests show that the scrambling effects are good.

## **Cellular Automata**

CA are widely used in applications such as art (generated images and music), random number generation, pattern recognition, routing algorithms, and games. The application of CA in the area of digital image processing includes image enhancement, compression, encryption, and watermarking.<sup>8</sup>

CA are dynamic, complex space and time discrete systems originally proposed by Stanislaw Ulam and John von Neumann in the 1940s as formal models for self-reproducing organisms.<sup>7</sup> They consist of a certain number of identical cells, each of which can take a finite number of states. The cells are distributed in space in a rectangular grid in one or more dimensions. At every time step, all the cells update their states synchronously by applying rules (transition function), which take as input the state of the cell under consideration and the states of its neighboring cells. The various CA models differ in the number of dimensions, the number of possible states, the neighborhood relationship, and the state update rules.

In spite of their simple construction, CA can produce complex behavior and generate useful operations. Stephen Wolfram classified 1D CA into four broad categories:

- class 1, ordered behavior;
- class 2, periodic behavior;
- class 3, random or chaotic behavior; and
- class 4, complex behavior.<sup>7</sup>

The first two classes are predictable, whereas random CA is unpredictable. Somewhere in the transition from periodic to chaotic, a complex, interesting behavior can occur.

### Cellular Automata Neighborhood

The radius or range is a shorthand method of specifying the neighborhood for common configurations. In a 1D CA of radius  $r = 1$ , each cell's neighborhood consists of itself, one cell to its immediate left and another to its immediate right.

In 2D CA, two common neighborhoods methods are used: Von Neumann and Moore neighborhoods.<sup>9</sup> The Von Neumann neighborhood of range  $r$  is defined by  $\text{vonNeumannNH}(x_0, y_0, r) = [(x, y): |x - x_0| + |y - y_0| \leq r]$ , and the number of cells in each neighborhood is  $2r(r + 1) + 1$ . A range of one, which is typically used, yields a total of five cells in the Von Neumann neighborhood, with neighbors to the north, south, east, and west.

The Moore neighborhood of range  $r$  is defined by  $\text{MooreNH}(x_0, y_0, r) = [(x, y): |x - x_0| \leq r, |y - y_0| \leq r]$ . The number of cells in each neighborhood is  $(2r + 1)^2$ . Typically  $r = 1$ , which yields a total of nine cells in the Moore neighborhood. John Conway's Game of Life uses the Moore neighborhood.<sup>9</sup>

### Cellular Automata Boundary Conditions

**Boundary conditions are essential in finite grids.** They determine, for instance, which cell is the left neighbor of the leftmost cell. A typical boundary condition is called periodic (or cyclic).<sup>10</sup> That is, 1D rows are turned into circles (their extreme cells become adjacent to each other), and 2D rectangular grids into toroids (connecting the leftmost column to the rightmost column and the top row to the bottom row). Static (or closed) boundary conditions are also common. Here, the extreme cells are assumed to be connected to permanent zero-state cells.<sup>10</sup>

### Cellular Automata Lambda Parameter

Chris Langton proposed a relation between the  $\lambda$  (Lambda) value of a set of rules and the Wolfram classification.<sup>11</sup> He quantified the classification scheme by introducing parameter  $\lambda$  and hypothesized that CA's computational capability is related to their average dynamic behavior, which  $\lambda$  is claimed to predict.

For a binary-state 1D CA, the Lambda parameter is the probability that a given neighborhood (chosen among all possible configurations) leads to a *nonquiescent state* (an active cell) and is equal to the fraction of 1s in the output table bits.

CA produces ordered behavior when Lambda values are close to 0 or 1 and chaotic behavior is somewhere in between. As  $\lambda$  reaches a critical value ( $\lambda_c$ , the edge of chaos), rules tend to exhibit long-lived, complex behavior. Langton's  $\lambda$  parameter has recently been extended to 2D CA.<sup>12</sup>

### Conway's Game of Life

CA have been used to simulate many real-life applications and phenomena.<sup>13,14</sup> It has also been proved that some 1D and 2D CA, such as the GL, are computationally equivalent to the Universal Turing Machine.<sup>7</sup>

The GL consists of an  $[M \times N]$  matrix of cells, where each cell may take only two states: alive or dead, represented by 1 and 0, respectively. Each cell has eight neighbors, according to the Moore neighborhood. At every time step, also called a generation, each cell computes its new state by determining the states of the cells in its neighborhood and applying the transition rules to compute its new state. Every cell uses the same update rules, and all the cells are updated simultaneously. A cell's next state is determined as follows:

- *Birth.* A cell dead at time  $t$  becomes alive at time  $t + 1$  if exactly three of its neighbors are alive at time  $t$ .
- *Death by overcrowding.* A cell alive at time  $t$  dies at time  $t + 1$  if four or more of its neighbors are alive at time  $t$ .
- *Death by exposure.* A cell alive at time  $t$  dies at time  $t + 1$  if one or none of its neighbors are alive at time  $t$ .
- *Survival.* A cell alive at time  $t$  will remain alive at time  $t + 1$  if two or three of its neighbors are alive at time  $t$ .

In this work, at each generation, the GL produces live and dead cells. Live cells will be selected to form the new pixel coordinates for the scrambled image, while dead cells will be neglected. This procedure can easily be generalized and applied to any type of digital media.

## Scrambling with the Game of Life

Adjacent pixels in an image have a strong correlation, which must be reduced to scramble the image. We propose performing pixel scrambling using several GL generations, which will add the diffusion property to the scrambling technique. The algorithm for the proposed technique can be described as follows:

1. An  $M \times N$  GL automaton is set up with an initial random configuration  $A_0$  and is set to run for  $k$  generations, thus obtaining  $\{A_1, A_2, \dots, A_k\}$  matrices.
2. Let  $Z$  denote the input image (plain image) and  $R$  the output image (scrambled image).  $A_1$  is the first generation produced by the GL. Set row = 1, column = 1.
3. For all  $(i, j)$  such that  $A_1(i, j) = 1$ , take the gray value of pixel  $Z(\text{row}, \text{column})$ , put it in  $R(i, j)$ , and increment (row, column) to point to the next pixel in the input image, in row-first order.
4. For  $q = 2, \dots, k$ , for all  $i, j$  such that  $A_q(i, j) = 1$  and  $A_n(i, j) = 0$  (for  $n = 1, \dots, q - 1$ ), take the gray value of pixel  $Z(\text{row}, \text{column})$ , put it in  $R(i, j)$ , and increment (row, column) to point to the next pixel.
5. Take the gray value of the remaining pixels in  $Z$  and put them in row-first order in those  $R(i, j)$  where, for all  $q = 1, 2, \dots, k, A_q(i, j) = 0$ .

For our experiments, we tested different modifications of the GL automaton using various neighborhoods methods (Von Neumann or Moore) and different boundary conditions (periodic or static). Steps 3 and 4 can also be modified for double scrambling—that is, first the image is scrambled in a horizontal direction and then the result is scrambled in a vertical direction.

Figure 1a displays the first generation of the GL CA; living cells (cells in state 1) are shaded. Figure 1b shows the algorithm's initial step for an image with  $4 \times 4$  pixels. Figure 1c displays the scrambled image after applying step 3 of the algorithm.

*Figure 1. Image scrambling using the first generation. (a) Game of Life first generation ( $A_1$ ). (b) Plain image pixel values. (c) Scrambled image pixel values. Living cells (cells in state 1) are shaded.*

To recover the original image, we execute the inverse of the scrambling algorithm using as keys the initial random configuration  $A_0$  and the number of iterations  $k$ .

## Analysis of the Scrambling Effect

The scrambling effect lets us determine the robustness of the scrambled image against passive attacks. Therefore, we want to show how the GL number of generations and other configurations affect the scrambling degree. Moreover, the scrambled image must have diffusion and confusion properties, which are the basis for designing practical ciphers. These two superior properties can be tested by computing the correlations of adjacent pixels in the scrambled image using the gray difference degree (GDD) function.

To evaluate the effect of image scrambling, we introduce gray difference and gray degree of

scrambling.<sup>15</sup> The gray difference of a pixel with a neighbor pixel is defined as follows:

$$GD(i, j) = \frac{1}{4} \sum_{i', j'} [P(i, j) - P(i', j')]^2$$

where  $(i', j') = \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$  and  $P(i, j)$  denotes the gray value at position  $(i, j)$ .

After computing the gray difference for every pixel in the image, except the pixels at the edges, we can compute the whole image's average neighborhood gray difference by summing and averaging:

$$E(GD(i, j)) = \frac{\sum_{i=2}^{M-1} \sum_{j=2}^{N-1} GD(i, j)}{(M-2) \times (N-2)}$$

The gray value degree is defined by

$$GDD = \frac{E'(GD(i, j)) - E(GD(i, j))}{E'(GD(i, j)) + E(GD(i, j))}$$

where  $E(GD(i, j))$  and  $E'(GD(i, j))$  denote the average neighborhood gray differences of the input and the scrambled images, respectively.

The GDD value will be a number between  $-1$  and  $1$ . Better scrambling corresponds to an absolute value near  $1$ .

### Correlation Analysis of the Number of Generations

To test the performance of the proposed scrambling method for different numbers of generations ( $k$  in the algorithm), we chose  $256 \times 256$ -pixel gray images in JPEG format, where the GL boundary condition was periodic. Figure 2 shows the result when 15 GL generations were used to scramble the images.

*Figure 2. The analysis of image scrambling effects when  $k = 15$ . (a) Tree image and (b) its scrambled effect. (c) Car image and (d) its scrambled effect. (e) Mountains image and (f) its scrambled effect. (g) Von Neumann image and (h) its scrambled effect.*

For different  $k$  values, we ran the algorithm 10 times, scrambling each image with different initial states  $A_0$  of the GL, and obtained 10 GDD values for each digital image and each  $k$  value. We then computed their average. Table 1 shows that the image-scrambling scheme is influenced by the number of GL generations ( $k$ ). The relation is direct, although the influence gradually decreases when the number of generations is greater than 17. Therefore, the more generations, the better the scrambling effect.

**Table 1. Gray difference degree (GDD) results for various gray images using varying numbers of generations.**

No. of generations ( $k$ )	Tree GDD	Car GDD	Mountains GDD	Von Neumann GDD
1	0.68097	0.70422	0.74168	0.89563
5	0.78549	0.77231	0.81930	0.92789
9	0.79105	0.77906	0.82367	0.92896
13	0.79213	0.78168	0.82399	0.92960
17	0.79304	0.78200	0.82675	0.92992

### Correlation Analysis of Cellular Automata Configurations

To determine which configurations generate the best scrambling effect and decide whether the complex

characteristics of GL are useful for image scrambling, we used our proposed algorithm to scramble the  $256 \times 256$ -pixel Barbara image (see Figure 3), also in JPEG format. We tested all the possible combinations: Moore and Von Neumann neighborhood methods as well as periodic and closed (static) boundary conditions. We set the number of generations for the CA to 15.

Figure 3. The effect of scrambling according to all possible parameter combinations of boundary condition and neighborhood methods for CA, where  $k = 15$ . Barbara (a) image and (b) image histogram. Moore neighborhood and (c) static and (d) periodic boundaries. Von Neumann neighborhood and (e) static and (f) periodic boundaries.

Table 2 shows that the highest GDD value (0.87586) corresponds to the Moore neighborhood and a periodic boundary, while the lowest value corresponds to the Von Neumann neighborhood and a closed boundary. With a periodic boundary, the randomness quality is better and the scrambling degree will be increased. In all cases, the scrambled image does not seem to have any similarity to the original Barbara image.

**Table 2. GDD values with different neighborhood methods and boundary condition.**

Boundary condition	$k$	Von Neumann neighborhood	Moore neighborhood
Periodic	15	0.87387	<b>0.87586</b>
Closed	15	0.86225	0.87494

### Correlation Analysis of Lambda Values

The GL with the Moore neighborhood has  $\lambda = 0.2734$ , which is located in the transition region and gives it its complex behavior. We can use digital image scrambling to study the behavior of different 2D CA, similar to GL, with Lambda parameters more or less similar to the critical value.

For our experiments, we designed eight 2D CA rules with different Lambda values and applied them to image scrambling. We first determined a transition function that provides the chosen Lambda values, and then we apply Equation 1 to calculate the rule number, as Table 3 shows.

The rule number  $C$  associated with a certain transition function  $f$  was derived by Ruisong Ye and Huiliang Li:<sup>15</sup>

$$C = \sum_{s=0}^1 \sum_{n=0}^8 f(s,n) 2^{2n+s} \quad (1)$$

where  $s$  is the center cell's state and  $n$  is the number of live cells (state = 1) in the neighborhood region.

**Table 3. 2D CA rules with their Lambda values.**

Lambda value	Transition function	Rule number
0.20703	$f(0, 4) = 1, f(1, 1) = 1, f(1, 2) = 1$ (others are 0)	296
0.23440	$f(0, 2) = 1, f(1, 1) = 1, f(1, 2) = 1, f(1, 3) = 1$ (others are 0)	184
0.25000	$f(0, 2) = 1, f(1, 1) = 1, f(1, 2) = 1, f(1, 3) = 1, f(1, 7) = 1$ (others are 0)	32,952

0.27340	$f(0, 3) = 1, f(1, 2) = 1, f(1, 3) = 1$ (others are 0)	GL
0.30000	$f(0, 4) = 1, f(1, 2) = 1, f(1, 3) = 1$ (others are 0)	416
0.32810	$f(0, 4) = 1, f(1, 2) = 1, f(1, 4) = 1$ (others are 0)	800
0.34375	$f(0, 4) = 1, f(1, 1) = 1, f(1, 2) = 1, f(1, 4) = 1$ (others are 0)	808
0.38671	$f(0, 4), f(1, 1), f(1, 2), f(1, 3), f(1, 6), f(1, 7)$ (others are 0)	41,384
0.41406	$f(0, 4), f(1, 1), f(1, 3), f(1, 4), f(1, 7)$ (others are 0)	33,672

We applied the algorithm to scramble the  $256 \times 256$ -pixel Von Neumann gray image with the different CA in Table 3. The 2D CA configurations used were a closed boundary,  $k = 15$ , and a Moore neighborhood. Table 4 and Figure 4 show the results.

**Table 4. GDD results for Von Neumann gray image with various Lambda values.**

<b>Lambda value</b>	<b>Von Neumann GDD</b>
0.20703	0.90289
0.23440	0.93986
0.25000	0.93836
<b>0.27340</b>	<b>0.94085</b>
0.30000	0.91789
0.32810	0.92185
0.34375	0.92572
0.38671	0.93572
0.41406	0.92949

*Figure 4. Lambda parameter versus gray difference degree (GDD) values.*

We can see from Table 4 and Figure 4 that the complex GL characteristics that occur around  $\lambda = 0.2734$  give scrambling degrees (GDD values) higher than those produced using various 2D CA with different Lambda values, further from the critical one. In fact, the GL and CA with similar Lambda values provide the best scrambling effect. Lambda values less than the critical value tend toward periodic behavior, while those greater than the critical value tend toward chaotic behavior.

## Comparing the Effect of GL Scrambling

Ye and Li proposed a scheme for image scrambling using chaotic CA (CCA) with different rule numbers (224 and 816).<sup>15</sup> To compare results, we used both their algorithms and ours to scramble the same  $256 \times 256$ -pixel gray images: camera man, Barbara, Lena, and three images from the Berkeley Segmentation Dataset and Benchmark ([www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds](http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds)) with a periodic boundary condition, Moore neighborhood, and  $k = 15$ .

Table 5 and Figure 5 show the results of the scrambling. In all cases, the scrambling effect of our

proposed algorithm is slightly better than the one proposed by Ye and Li.<sup>15</sup>

**Table 5. Comparative analysis of the scrambling effect.**

Image	Our proposed scheme	Chaotic CA <sup>15</sup>
Camera man	0.8971	0.8926
Barbara	0.8749	0.8740
Lena	0.9320	0.9311
Image 157,055	0.8821	0.8731
Image 69,015	0.8827	0.8789
Image 239,096	0.9388	0.9134

Figure 5. Image scrambling effects with a periodic boundary, Moore neighborhood, and  $k = 15$ . (a) Camera man image and (b) its scrambled effect. (c) Barbara image and (d) its scrambled effect. (e) Lena image and (f) its scrambled effect. (g) Test image 157,055 and (h) its scrambled effect. (i) Test image 69,015 and (j) its scrambled effect. (k) Test image 239,096 and (l) its scrambled effect.

We also compared our proposed algorithm to the Josephus algorithm (JA)<sup>16</sup> and a chaos mapping algorithm with Logistic mapping algorithm (LA).<sup>17</sup> The results in Table 6 show that our algorithm is clearly better.

**Table 6. Gray scrambling degree comparison.**

Image	Our proposed scheme	Josephus algorithm <sup>16</sup>	Logistic mapping algorithm <sup>17</sup>
Lena	0.9320	0.9017	0.9010
Camera man	0.8971	0.8871	0.8832
Image 157,055	0.8821	0.8789	0.8780
Image 69,015	0.8827	0.8715	0.8646
Image 239,096	0.9388	0.9065	0.8976

Obviously, the GL rules are totalistic, which means that the number of neighbors in specific states is considered in each step, while their positions are not taken into account.

Finally, we carried out a set of experiments to test the effectiveness and robustness of our proposed scheme under various image processing operations. Figure 6 gives the results, which show that our scheme is robust with respect to common image processing operations, such as the Salt and Pepper, Gaussian noise pollution, and lossy JPEG compression attacks.

Figure 6. GDD of scrambled image under various attacks. (a) With no attack, the GDD was 0.8971. (b) After the salt and pepper noise attack (density = 0.02), the GDD was 0.6804. (c) After the Gaussian noise attack (variance=0.01), the GDD was 0.6120. Lastly, we performed two lossy JPEG compression attacks: (d) With quality factor = 80, the GDD was 0.8821. (e) With quality factor=50, the GDD was 0.8337.



## Conclusion

**//AUTHOR: MM's style discourages summarization, so I've removed such text from your conclusion. Can you provide any future applications of your work or other directions?//**

Our experimental results reveal several characteristics of the proposed algorithm. First, we can provide high security by scrambling pixel locations using double scrambling—first scrambling the image in horizontal order and then scrambling the result in vertical order. Second, this method enables excellent confidentiality. That is, scrambling broke the correlation between adjacent pixels, making it difficult for attackers to crack. An attacker cannot break the encrypted image even if the algorithm is open. Our approach also proved highly efficient. When we use complex GL behavior, the rate of encryption and decryption obviously improved because of the high degree of the diffusion process—that is, the correlativity of the adjacent pixels decreases. The restoration process can also be easily carried out following the inverse steps with the precise keys. Lastly, the proposed algorithm is suitable for any size digital image and can be used to encrypt binary, grayscale, and color images.

In the future we intend to test this procedure with multimedia files of other types, such as audio files. We expect that CA of the GL type will also be very useful to scramble files in those formats.

## Acknowledgment

*This work has been partially sponsored by the Spanish MICINN project TIN2011-28260-C03-02.*

## References

1. F. Maleki et al., "An Image Encryption System by Cellular Automata with Memory," *Proc. 3rd Int'l Conf. Availability, Reliability, and Security*, IEEE CS Press, 2008, pp. 1266–1271.
2. N. Jiping et al., "A Digital Image Scrambling Method Based on AES and Error-Correcting Code," *Proc. Int'l Conf. Computer Science and Software Eng.*, vol. 3, IEEE CS Press, 2008, pp. 677–680.
3. L. Xiangdong et al., "A New Chaotic Image Scrambling Algorithm Based on Dynamic Twice Interval-Division," *Proc. Int'l Conf. Computer Science and Software Eng.*, vol. 3, IEEE CS Press, 2008, pp. 818–821.
4. L. Zhu et al., "A Novel Algorithm for Scrambling Digital Image Based on Cat Chaotic Mapping," *Proc. Int'l Conf. Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 06)*, IEEE CS Press, 2006, pp. 601–604.
5. L. Zhang et al., *Computational Intelligence and Security*, Springer, 2005.
6. Z. Shang, H. Ren, and J. Zhang, "A Block Location Scrambling Algorithm of Digital Image Based on Arnold Transformation," *Proc. 9th Int'l Conference for Young Computer Scientists*, IEEE CS Press, 2008, pp. 2942–2947.
7. S. Wolfram, *A New Kind of Science*, Wolfram Media, 2002.
8. R.-J. Chen and J.-L. Lai, "Image Security System Using Recursive Cellular Automata Substitution," *Pattern Recognition*, vol. 40, no. 5, 2007, pp. 1621–1631.
9. S. Wolfram, *Theory and Applications of Cellular Automata*, World Scientific, 1986.
10. S. Torbey, "Towards a Framework for Intuitive Programming of Cellular Automata," master's thesis, School of Computing, Queen's Univ., 2007.
11. C. Langton, "Computation at the Edge of Chaos: Phase Transitions and Emergent Computation," *Physica D*, vol. 42, no. 1–3, 1990, pp. 12–37.
12. M. Mitchell, P.T. Hraber, and J.P. Crutchfield, "Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations," *Complex Systems*, vol. 7, 1993, pp. 89–130.
13. H. Wang, G. Nie, and K. Fu, "Cellular Automata Simulation of the Growth of Bone Tissue," *Proc. 4th Int'l Conf. Natural Computation*, vol. 7, IEEE CS Press, 2008, pp. 421–424.

14. F. Kyoomarsi et al., "Using Chemical Cellular Automata in Simulation of Chemical Materials," *Proc. 5th ACIS Int'l Conf. Software Eng. Research, Management, and Applications (SERA 2007)*, IEEE CS Press, 2007, pp. 769–773.
15. R. Ye and H. Li, "A Novel Image Scrambling and Watermarking Scheme Based on Cellular Automata," *Proc. Int'l Symp. Electronic Commerce and Security*, IEEE CS Press, 2008, pp. 938–941.
16. X. Desheng and X. Yueshan, "Digital Image Scrambling Based on Josephus Traversing," *Computer Eng. and Applications*, vol. 10, 2005.
17. G. Ye, X. Huang, and C. Zhu, "Image Encryption Algorithm of Double Scrambling Based on ASCII Code of Matrix Element," *Proc. Int'l Conf. Computational Intelligence and Security*, IEEE CS Press, 2007, pp. 843–847.

**Abdel latif Abu Dalhoum** is an associate professor of computer science at the University of Jordan. His research interests include evolutionary algorithms, DNA computing, data mining, fractals, cellular automata, and P systems. Abu Dalhoum has a PhD in computer engineering from the Universidad Autónoma of Madrid, Spain. Contact him at [a.latif@ju.edu.jo](mailto:a.latif@ju.edu.jo).

**Basel A. Mahafzah** is an associate professor of computer science at the University of Jordan. His research interests include parallel and distributed computing, interconnection networks, performance evaluation, artificial intelligence, data mining, software testing, and cellular automata. Mahafzah has a PhD in computer engineering from the University of Alabama. Contact him at [b.mahafzah@ju.edu.jo](mailto:b.mahafzah@ju.edu.jo).

**Aiman Ayyal Awwad** is a lecturer in computer science at the Tafila Technical University, Jordan. His research interests include image processing, cellular automata, network security, data hiding, and watermarking. Awwad has an MSc in computer science from the University of Jordan. Contact him at [aimanw\\_2000@yahoo.com](mailto:aimanw_2000@yahoo.com).

**Ibraheem AL-Dhamari** is a lecturer in computer science at San'a University, Yemen. His research interests include cellular automata, image processing, fMRI, and machine learning. AL-Dhamari has an MSc. in computer science from the University of Jordan. Contact him at [ibr\\_ex@yahoo.com](mailto:ibr_ex@yahoo.com).

**Alfonso Ortega** is a professor at the Escuela Politécnica Superior in the Universidad Autónoma of Madrid, Spain. His research interests include computer languages, complex systems, graphics, and theoretical computer science. Ortega has a PhD in computer science from the Universidad Autónoma of Madrid. Contact him at [alfonso.ortega@uam.es](mailto:alfonso.ortega@uam.es).

**Manuel Alfonseca** is a member of the Department of Computer Science at the Escuela Politécnica Superior in the Universidad Autónoma of Madrid, Spain. His research work includes digital continuous simulation, complex systems, fractals and parallel development grammars, and artificial intelligence. Alfonseca has a PhD in electronics engineering from the Universidad Politécnica of Madrid. Contact him at [manuel.alfonseca@uam.es](mailto:manuel.alfonseca@uam.es).