Global Journal of Computer Science and Technology
Volume 12 Issue 8 Version 1.0 April 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Optimal Rules Identification for a Random Number Generator Using Cellular Learning Automata

By Atefeh Ghalambor Dezfuly, Saeed Setayeshi, Mohammad Mosleh & Mohammad Kheyrandish
Islamic Azad University, Dezful, Iran

Abstract - The cryptography is known as one of most essential ways for protecting information against threats. Among all encryption algorithms, stream ciphering can be indicated as a sample of swift ways for this purpose, in which, a generator is applied to produce a sequence of bits as the key stream. Although this sequence is seems to be random, severely, it contains a pattern that repeats periodically. Linear Feedback Shift Registers and cellular automata have been used as pseudo-random number generator. Some challenges such as error propagation and pattern dependability have motivated the designers to use CA for this purpose. The most important issue in using cellular automata includes determining an optimal set of rules for cells. This paper focuses on selecting optimal rules set for such this generator with using an open cellular learning automata, which is a cellular automata with learning capability and interacts with local and global environments.

Keywords : *Cryptography; Symmetric encryption; stream ciphering; Learning Cellular Automata; local environment; global environment.*

GJCST Classification: *F.1.1*

Strictly as per the compliance and regulations of:

Optimal Rules Identification for a Random Number Generator Using Cellular Learning Automata

Atefeh Ghalambor Dezfuly [α], Saeed Setayeshi [σ], Mohammad Mosleh [ρ] & Mohammad Kheyrandish [ω]

Abstract - The cryptography is known as one of most essential ways for protecting information against threats. Among all encryption algorithms, stream ciphering can be indicated as a sample of swift ways for this purpose, in which, a generator is applied to produce a sequence of bits as the key stream. Although this sequence is seems to be random, severely, it contains a pattern that repeats periodically. Linear Feedback Shift Registers and cellular automata have been used as pseudo-random number generator. Some challenges such as error propagation and pattern dependability have motivated the designers to use CA for this purpose. The most important issue in using cellular automata includes determining an optimal set of rules for cells. This paper focuses on selecting optimal rules set for such this generator with using an open cellular learning automata, which is a cellular automata with learning capability and interacts with local and global environments.

Keywords : Cryptography; Symmetric encryption; stream ciphering; Learning Cellular Automata; local environment; global environment.

I. Introduction

Based on applications, such as data storing, transferring and processing, the information can be threatened in several way. In each threat, dependent to the threat agent, the information may be changed and lose their credibility, or be stolen and lose their confidentiality, only. The cryptography has been considered as an approach to protecting information, in both cases, and in different conditions, can help to preserving data credibility and confidentiality. Generally, this approach includes transforming a plain-text into a ciphered-text. In this way, a determined function and a specific key are used, and the common purpose is establishing a secure communication between a sender A and a receiver B over an insecure communication channel [1]. Regarding key type viewpoint, the encryption algorithms are divided into two broad classes: symmetric and asymmetric algorithms. In the first class, both a communication parties use a common secret key for both encryption and decryption process; whereas in second ones, each of communication parties has its private secret key and also a public key. Asymmetric-key algorithms provide stronger securities compared to symmetric ones; however, because of massive numeral computation for increasing the security of these algorithms, they have lower speeds compared to the first class algorithms [2].

The symmetric algorithms are divided into Block ciphering and Stream ciphering algorithms, themselves. In both ones, using an efficient tool for adding the randomness in the key or ciphered text is considered, as a basis. So, the Cellular Automata (CA) as a complex parallel processing model has been used in both these mentioned algorithms. The CA can be used for increasing encryption and decryption security and speed, via its parallel operation nature and its pseudo-random output [3,4]. However, the main problem in using CA for cryptography includes selecting a rule set for cells that provides security requirements, optimally. In this paper, using a Learning Cellular Automata (LCA), as an extended model of CA, has been considered for selecting an optimal rule set for a key generator based on CA that can be used in stream ciphering.

For the purpose of this paper, it has been organized as follows: section 2 introduces stream ciphering. In section 3, key generating will be focused and after introducing CA as a common key generator, using it for this purpose will be reviewed. Also, the Learning Cellular Automata (LCA) as the basis of proposed model will be introduced. Then, section 4 presents the proposed model for identifying optimal rules that must be used in CA-based random number generator. Section 5 has been designated for reporting experiments results and finally, section 6 includes conclusion.

II. Stream Ciphering

In a block ciphering system, the plain text is divided into several blocks with a specific size, and each block is transformed to a ciphered block, independently. However, the stream ciphering process transforms each plain text bit to a cipher bit, per a time instance [5]. A

Author α : Computer Engineering Department, Dezful Branch, Islamic Azad University, Dezful, Iran. E-mail : at_ghalambor@yahoo.com
Author σ : Medical Radiation Department, Amirkabir University of Technology, Tehran ,Iran. E-mail : Setayesh@aut.ac.ir
Author ρ : Computer Engineering Department, Dezful Branch, Islamic Azad University, Dezful, Iran. E-mail : Mosleh@iaud.ac.ir
Author ω : Computer Engineering Department, Dezful Branch, Islamic Azad University, Dezful, Iran. E-mail : kheyrandish@iaud.ac.ir

stream ciphering system includes a pseudo random bit sequence generator and a function box. The generator produces a bit stream that is considered as a key sequence and is combined with the plain text in a bitwise manner, and generates a cipher bit. This combination is done by a function box which often is an XOR-operator.

The performance of an encryption system depends on randomness degree of key stream, used in encryption. So the key generator plays an important role in this way. Among different key generators, the random number generators based on Linear Feedback Shift Register (LFSR) and Cellular Automata (CA) have been known as prevalent ones.

An LFSR includes a shift register together with a set of XOR operators which combine the feedbacks extracted from the register cells. This model is defined by an n-degree polynomial which specifies the operators and feedbacks arrangements [2]. A sample of these generators has been shown in fig.1.

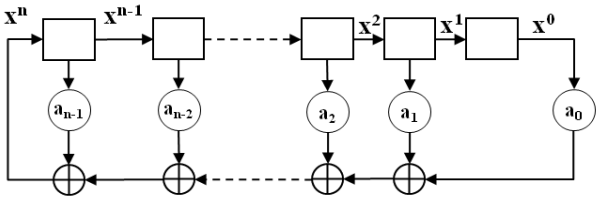


Fig. 1: A typical sample of LFSR

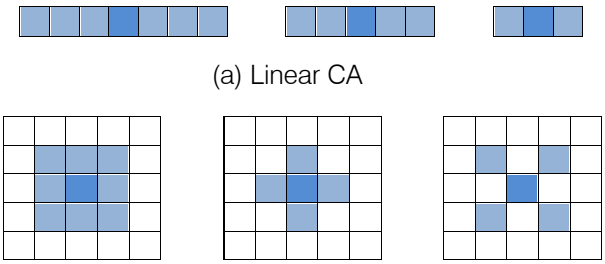
In some systems, a combination of LFSRs has been utilized for increasing the randomness of generated sequence. This requires applying nonlinear combination functions. The researches have indicated that utilizing linear CA provides an operation similar to LFSR [15,16,17,18]. In the other word, CA structures can be used in key producing, as a replacement for LFSR. Some experiments results have shown that CA is proper for producing random features in digital circuits and automatic control systems [17]. A desirable facet of CAs refers to independency between the sequence of generated bits and the states of neighbor cells. In this case, CA can be preferred compared to LFSR for stream ciphering. On the other hand, the main advantage of CA is that several generators designed as nonlinear combinations of LFSRs, when designing with CA, preserve linearity [20].

These mentioned advantages have been considered as the reasons to applying the CA as a Pseudo-Random sequence generator (PRSG). Really, the CA has an outstanding role in this area and is considered as an important tool for generating random sequences [3,6,7,8].

III. Key Generation

a) *Cellular Automata (CA)*

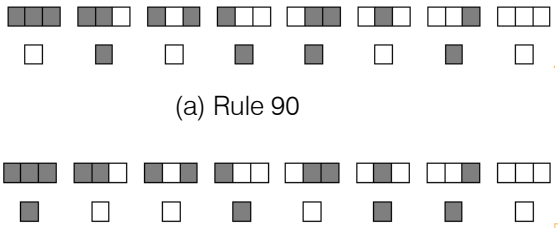
The concept of CA was introduced by Von Neumann and Ulam at 1950s, for the first time, and was considered by Wolfram, more extensively. The simple structure of CA has attracted the researchers in several different areas, such as implementing the computing tools and modeling the natural systems [9]. Each CA includes a set of simple elements called cells that each has a finite set of states, and interacts with its adjacent cells (neighborhood), locally. The next state of each specific cell is determined with a rule that is a function of current states of that cell and its neighbors. Fig. 2 shows some samples of neighborhood patterns [10]. Regarding a k-cell neighborhood, P=2k neighborhood patterns and thus 2P possible rules can be defined. Fig. 3 shows schematic diagrams for some possible rules operations, assuming k=3.



(a) Linear CA

(b) 2-Dim CA

Fig. 2: Some common neighborhood patterns for linear and 2-dim CA



(a) Rule 90

(b) Rule 150

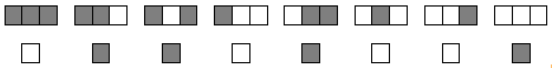
Fig. 3: Two common possible rules by considering neighborhood size k=3

b) *Pseudo-Random Sequence Generator (PRSG)*

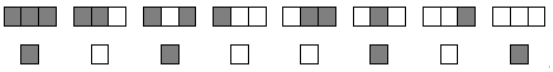
What is important in applying CA as a PRSG is selecting the rule set governing the cells such that the generated sequence provides a desirable level of randomness. Wolfram, for the first time, applied an uniform CA with k=3 neighborhood for encryption, at 1986. He used Rule 30 for all cells. His proposed model operated as a pseudo-random key stream generator for using in stream ciphering [18,21]. Afterward, other researchers showed that when using non-uniform CA, a higher level randomness is provided. For example, Habutsu et al. and Gutowitz and Nandi et al. used non-uniform CA with Rule 90 and Rule 150 for mentioned purpose and indicated that their generated key stream have a better quality than Wolfram's one [22,23].

Tomassini and Perrenoud proposed a linear CA with k=3 and the rule set {90,105,150,165} [24].

Schematic forms for Rule 105 and Rule 165 have been shown in fig.4.



(a) Rule 105



(b) Rule 165

Fig. 4: Schematic diagram for Rule 105 and Rule 165 for k=3

They used an evolutionary method called cell programming for searching optimal rules. This method is an evolutionary computational approach similar to diffusion model in parallel genetic algorithms. The entropy of generated sequence was used as a quality criterion for rules [24].

This paper has focused on using the learning capability of a Learning Cellular Automata for selecting a set of optimal rules for using in a CA-based PRSG.

c) Learning Cellular Automata (LCA)

The Learning Automata (LA) as another model of automata includes a finite automata which interacts with an environment. The automata have a finite set of actions that each can be selected with a specific selection probability. Such this model operates in two phases: Training phase and Testing phase. In the first one, some probabilities are assigned to automata actions, such that all of them sum to one. Then, in each step, the automata select an action randomly and based on probabilities. The environment receives the selected action and responses to it with a desirability or undesirability of selected action, by sending back a response to it. Then, by considering the received response, the automata reward the selected action or penalize it. Rewarding an action implies an increase in action selection probability for next steps and penalizing it includes decreasing this probability. The training phase continues until the probability of a special action approaches to one and thus is determined as optimal action [12,13]. A sample of LA interacting with an environment has been shown in fig. 5 [14].

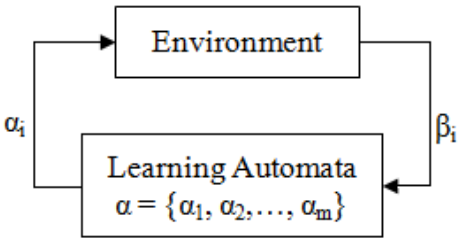


Fig. 5: An LA interacting with an environment

A set of LAs that in addition to interacting with the environment, have local interactions between themselves, form a Learning Cellular Automata (LCA). In such this structure, the desirability of selected action by each learning cell, is determined based on the states or selected actions by that cell and its neighbors. Updating all LAs is performed simultaneously; thus, the LCA has a parallel nature. A sample LCA has been shown in fig. 6, in which, each cell LAi selects an action denoted by Ai and the corresponding environment return a response denoted by Bi.

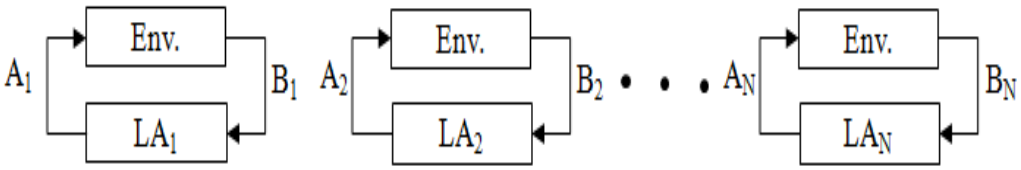


Fig. 6: The block diagram of an LCA

IV. The Proposed Model

The proposed method in this paper focuses on finding a configuration of rules governing on the cells of a CA that operates as a PRSG. In this way, an LCA with two types of environment has been considered.

Each cell of LCA interacts with a local environment and all cells have interactions with a global environment, simultaneously.

The proposed method can be explained in two stages. In the first one, by considering a uniform CA with k-cell neighborhood, selecting M best rules among all possible rules is regarded. In order to do this, two empty vectors named best_rules and best_cnt are defined to hold the best rules indices and their correspond statistics. Then M different configurations are imposed to the CA, one by one. By imposing each configuration ci, as the initial state of the CA, during 2P stages (P=2K, in which, K is the size of neighborhood), all possible rules are assigned to the CA cells, one after another. By assigning each rule Rj to all cells of the CA, N sequences with the length l, are generated during the CA operation. Then, the entropy values are calculated for all sequences, using the method described in [11] and based on Eq. (4).

$$E = -\sum_{i=0}^{n-1} p_i \log_2^{p_i} \quad (4)$$

The max, min and average entropy values for each rule assigned to CA are stored in three vectors, separately. When the entropy values for all rules were calculated, the rules with maximum values in three mentioned vectors are selected. If these three rules are same, and not be found in best_rules vector, the rule is added to the mentioned vector and its correspond counter is set to one in best_cnt. If the rule exists in the best_rules, already, its counter is increased by one.

This process is performed for M times and finally two mentioned vectors will contain best rules and the counters which indicate the number of times that each rule has been identified as the best rule. Among all best_rules members, m ones with top counter values are selected to be used for the next stage. This process has been shown in fig. 7 by a flowchart.

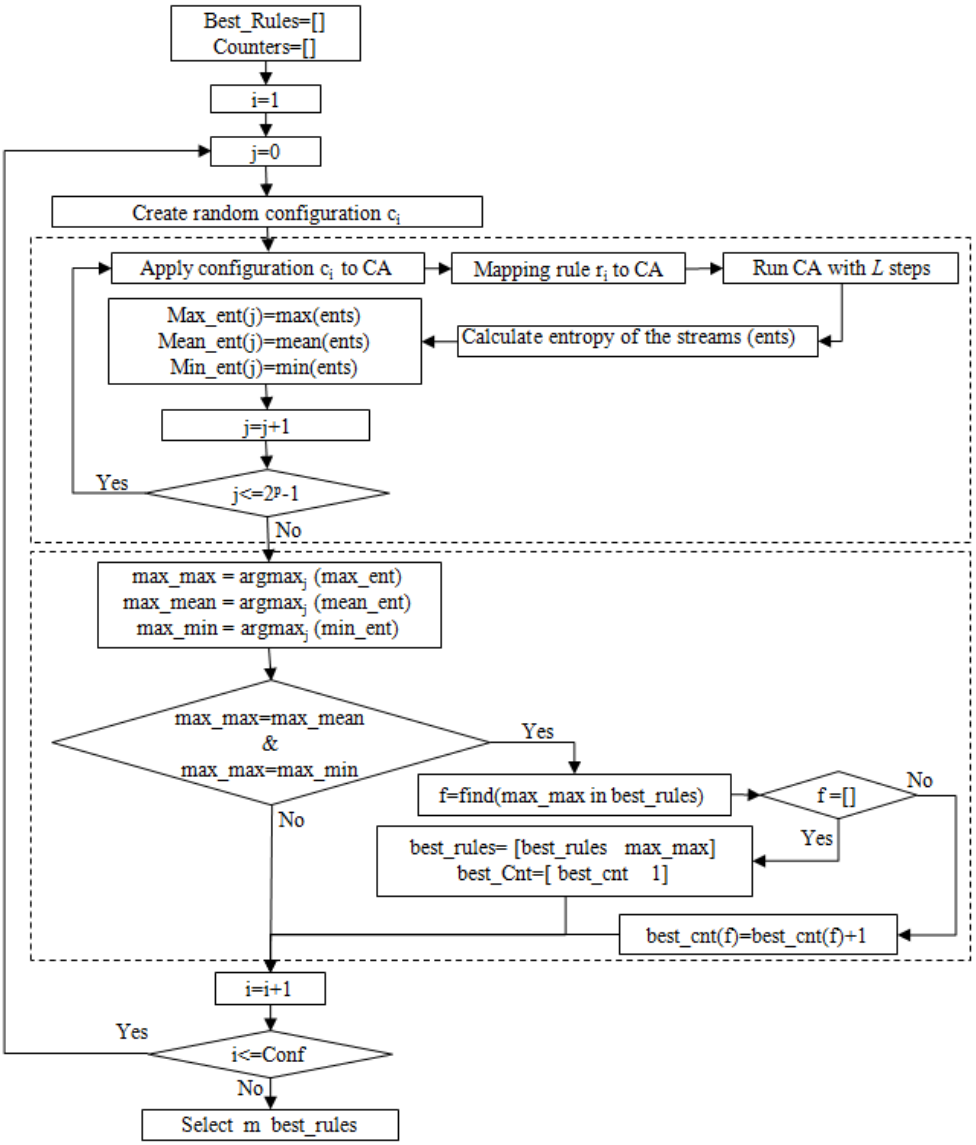


Fig. 7: The flowchart for selecting m best rules

In the second stage, the main part of proposed model is considered. This is an LCA that each of its cells can select an action among m different actions. Each action corresponds to a rule in best rules set which has been obtained in previous stage. Each cell interacts with a local environment which receives sequences from the current cell, and its neighbor cells, and returns a response to the cell. Also, a global environment interacts with all cells and returns a response based on the sequences generated by all LCA cells.

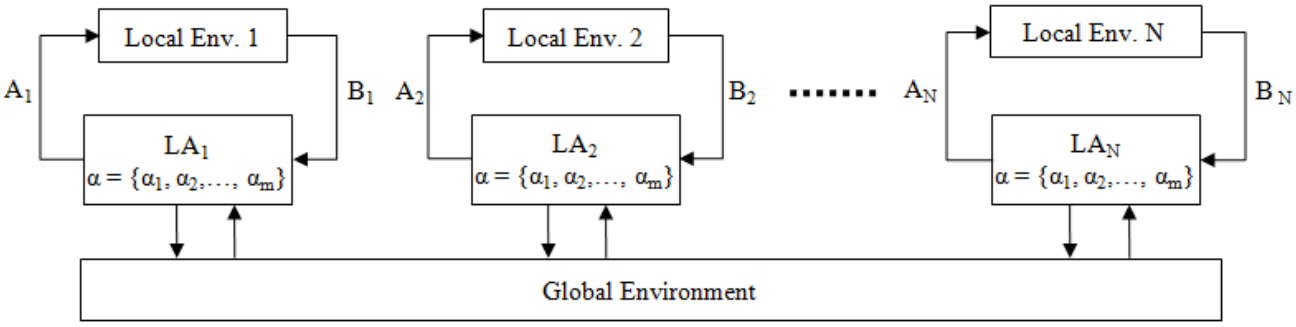


Fig. 8: The schematic diagram of proposed model

This stage includes an initial phase in which a probability value is assigned to each action, initially. Then, in the training phase, all cells of LCA select actions, simultaneously. The selected action by each cell is considered as its rule, and all cells will generate l-length sequences with using that rules and through l operation step. The generated sequences by each cell and its neighbor cells (its r-1 left and right cells) are passed to the local environment and the entropy values will be calculated for these sequences. Then, by considering an r-cell neighborhood for evaluating the quality of generated sequences, each local environment compares the calculated entropy values and if this entropy is maximum value among all r entropy values, returns a response as zero or returns one, otherwise.

Each learning cell rewards its selected action, if receive a zero as the local environment response; otherwise, it penalizes the selected action. After imposing rewards or penalties by all cells, all generated sequences are passed to the global environment.

This environment calculates all entropy values and the max, min and mean values for them. These values are compared with their previous values and if an improvement is detected, a response as 0 will be passed back to each LA.

In this case, each cell will impose a reward to its selected action. The training phase will be continue as described, until the actions probabilities in all cells reach a steady state. A flowchart describing each LA operation interacting with environments has been shown in fig. 9.

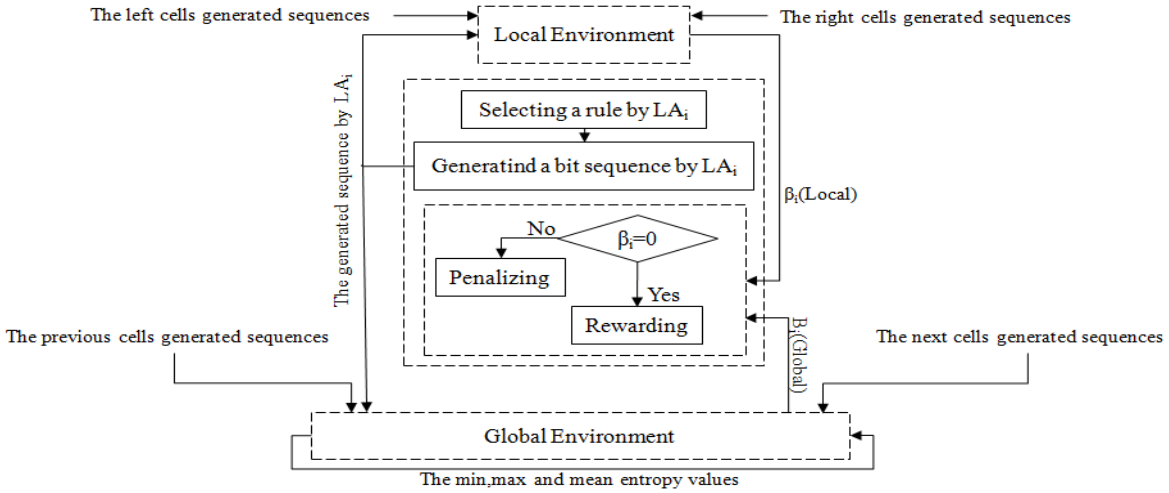


Fig. 9: The schematic diagram of each LCA cell interacting with local and global environments

V. Experiments Results

For evaluating the proposed model, a CA with the size N=50 and a neighborhood size, k=3, was considered and by applying 100 different initial configurations, m=10 top rules among 256 possible rules were selected (based on l=100 bits length sequences). A bar diagram for best rules found during these 100 configuration imposing has been shown in fig. 10.

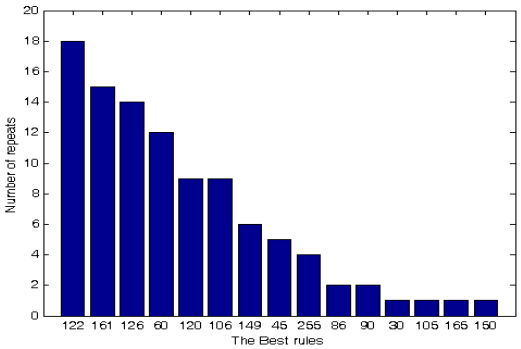


Fig. 10: The bar diagram for best rules repetitions

The 10 selected rules were assigned as the actions of each cells in a LCA with N=32 cells and initial probability values as 0.1 were considered for them. The initial configuration of LCA states was determined by a N=32 bits vector. Also, the neighborhood size considered for next state calculation in each cell was selected as k=3. Afterward, multiple experiments were done.

In the first one, the effect of learning neighborhood on the entropy values has been considered. For this purpose, after entering the training phase, each cell selects an action randomly and based on exist probabilities and then, all cells start generating pseudo-random sequences. The local environments, by considering r as the neighborhood size in evaluating each cell performance, calculates entropy values and produces a response as zero or one for each cell. Then the global environment, regarding all entropy values returns a response. This experiment was performed for r=3 and r=5 and its results have been reported in table(1).

Table 1: The effect of Learning Neighborhood on entropy values

Neighborhood	Entropy		
	Mean	Min	Max
r=3	3.8487	3.6998	3.9205
r=5	3.9899	3.9889	3.9902

In the second experiment, the effect of learning neighborhood on the number of required train steps has been regarded.

In order to this, the above experiment has been repeated for 10 different initial configurations and for r=3 and r=5. The results have been reported in Table (2).

Table 2: The effect of Learning Neighborhood on training steps number

Neighborhood	1	2	3	4	5	6	7	8	9	10
r=3	137	168	133	126	145	87	124	130	153	137
r=5	132	15	177	108	136	199	138	178	172	137

The FIPS-140-2 experiments include statistics tests defined by National Institute of Standard and Technology (NIST) in USA for evaluating encryption processes and random number generators. These include 4 parts: Frequency (Monobit) Test, Poker Test, Runs Test and Long Runs Test. So, in other experiment, mentioned tests have been applied for evaluating the key streams generated by Non-uniform LCA model presented in this paper. The results for 20000 bits key stream and for neighborhood sizes r=3 and r=5 have been reported in Table (3).

Table 3: The results for FIPS-140-2 Tests on generated key streams by proposed model

Test	r=3	r=5	Permitted Values
	Result	Result	
Monobit	Ok	Ok	9725<X<10275
Poker	Ok	Ok	2.16 < X<46.17
Runs	Ok	Ok	Refer to Table (2)
Long runs	Ok	Ok	a run length ≥26

Table 4: Standard values for Runs Test

Required Interval	Length of Run
2,343 – 2,657	1
1,135 – 1,365	2
542 – 708	3
251 – 373	4
111 – 201	5
111 – 201	6

Table 5: The results of proposed model compared with previous models

Model	Proposed Model	Szaban Model	Tomassini Model	Wolfram model
Max Entropy	3.9902	3.9903	3.9902	3.9905
Min Entropy	3.9889	3.9888	3.9885	3.9882
Mean Entropy	3.9899	3.9894	3.9894	3.9894
Monobit Test	Ok	Ok	Ok	Ok
Poker Test	Ok	Ok	Ok	Ok
Runs Test	Ok	Ok	Ok	Ok
Long Runs Test	Ok	Ok	Ok	Ok

VI. Conclusion

The proposed model in this paper, has considered a LCA for determining optimal rules used by a pseudo-random key stream generator based on CA. On this way, parallel operation by LCA cells has a significant effect on optimal rules determination speed. Also, the number of training steps, considerably, affects the convergence ratio in cells. For evaluating the key stream generated by proposed model, a set of standard tests (FIPS-140-2) have been applied that evaluate its randomness property. A 20000-bit stream produced by proposed model has passed all tests defined in mentioned standard. Also, the obtained results for proposed model, in compare with previous models have been summarized in table (5).

References Références Referencias

1. Schneier, B., (1996), Applied cryptography, Second Edition, John Wiley and sons.
2. Lai, C. & Saskatchewan, R., (Aug 2000), High-Speed Cellular Automata based block cipher and fault tolerant public key cryptosystems, Thesis, computer science ,Regina ,Canada.
3. Seredynski, F. & Bouvry, P. & Zomaya, A. & (2004), Cellular automata computations and secret key cryptography, Published by Elsevier B.V.
4. Golomb, S., (1982), Shift- Register Sequences (revised edition), Aegean Press.
5. F´uster-Sabater, A. & Caballero-Gil, P., (Sept 2006), On the Use of Cellular Automata in Symmetric Cryptography, Volume 93, Numbers 1-3, pp. 215-236, Springer.
6. Bardell, P. H., (1990), Analysis of Cellular Automata used as Pseudo-Random Pattern Generators, In International Test Conference, pages 762-768.
7. Kari, J., (2005), Theory of cellular automata: A survey, Theoretical Computer Science, Vol. 334, No. 3, pp. 3 - 33.
8. Fúster-Sabater, A. & Caballero-Gil P., (2009), Synthesis of Cryptographic Interleaved Sequences by Means of Linear Cellular Automata, Applied Mathematics Letters, Vol. 22, No. 10, pp. 1518-1524.
9. Neumann, J. V., (1966), The Theory of Self Reproducing Automata, A. W. Burks (ed), Univ. of Illinois Press, Urbana and London.
10. Ganguly, N. & Sikdar, B. & Deutsch, A. & Canright, G., & Chaudhuri, P. ,(2003), A Survey on Cellular Automata, Centre for High Performance Computing, Dresden University of Technology, Dresden, Germany.
11. Szaban, M. & Seredynski, F. & Bouvry ,P., (2006), Evolving Collective Behavior of Cellular Automata for Cryptography, IEEE MELECON May 16-19, Benalmadena (Malaga), Spain.
12. Beigy, H.& Meybodi, M. R., (2004), A Mathematical Framework for Cellular Learning Automata, Advanced in Complex Systems,to Appear,vol 7,pp.295-319.
13. Narendra, K. S. & Thathachar, M. A. L., (1989), Learning Automata: An Introduction, Prentice Hall.
14. Kumpati S.& Narendra K. S. & Thathachar M. A. L., (July 1974), Learning Automata A Survey, IEEE Transactions on Systems, Man, And Cybernetics ,Vol. Smc-4, No. 4.
15. Bao, F., (2003), Crytanalysis of a New Cellular Automata Cryptosystem, 8th Australasian Conference on Information Security and Privacy-ACISP, Lecture Notes in Computer Science, Springer Verlag 2727 .416- 427.
16. Blackburn, S.& Merphy, S.& Paterson, K., (1997), Theory and Applications of Cellular Automata in Cryptography, IEEE Transactions on Computers 46, 637- 638.
17. Nandi, S.& Kar, B.K.& Chaudhuri, P.P., (1994), Theory and Applications of Cellular Automata in Cryptography, IEEE Transactions on Computers 43,1346- 1357.
18. Wolfram, S., (1994), Cryptography with Cellular Automata, Advances in Cryptology-CRYPTO'85. Lecture Notes in Computer Science, Springer Verlag 218 , 22- 39.
19. Cho, S.& Un-Sook C.& Yoon- Hee, H., (2004), Computing Phase Shifts of Maximum-Length 90/150 Cellular Automata Sequences, Proc. of ACRI 2004. Lecture Notes on Computer Science, Springer Verlag, 3305, 31- 39.
20. Serra, M. & Slater, T., Muzio J. ,Miller, D.M., (1990), The Analysis of One dimensional Linear Cellular Automata and Their Aliasing Properties, IEEE Transactions on Computer- Aided Design of Integrated Circuits and Systems 9 (7),767- 778.
21. Wolfram, S., (2002), A new kind of science, Champaign, IL: Wolfram Media,pp.29-30,52,59,317 and p.871.
22. Diaz Len, R.& Hernandez Encinas, A.& Hernandez Encinas, L.& Hoya White, S. & Martin Del Rey, A. & Rodriguez, G.& Visus Ruiz, I., (2001), Wolfram Cellular Automata And Their Cryptographic Use As Pseudorandom Bits Generators, Memoria Samuel Solorzano Barruso Foundation (Spain) and Ministerio de Ciencia y Tecnologia (Spain) under grant TIC2001-0586.
23. Habutsu, T. et al., (1991), A Secret Key Cryptography By Iterating Chaotic Map, Proc. of Eurocrypt_91, pp. 127-140.
24. Tomassini, M.& Perrenoud, M. , (2000), Stream Cipher With One And Two Dimensional Cellular Automata, in M. Schoenauer et al. Eds.) Parallel Problem Solving from Nature - PPSN VI, LNCS 1917, Springer, pp. 722-731.

This page is intentionally left blank